Intelligent State-of-Charge and State-of-Health Estimation Framework for Li-ion Batteries in Electrified Vehicles using Deep Learning Techniques

INTELLIGENT STATE-OF-CHARGE AND STATE-OF-HEALTH ESTIMATION FRAMEWORK FOR LI-ION BATTERIES IN ELECTRIFIED VEHICLES USING DEEP LEARNING TECHNIQUES

BY

EPHREM CHEMALI, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2018)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Intelligent State-of-Charge and State-of-Health Estimation Framework for Li-ion Batteries in Electrified Vehicles using Deep Learning Techniques
AUTHOR:	Ephrem Chemali M.Sc., (Physics) York University, Toronto, Canada
SUPERVISOR:	Dr. Ali Emadi
NUMBER OF PAGES:	xxv, 199

This work is dedicated to my parents, Julia and Elias Chemali. No words can describe my appreciation for your profound love and sacrifice.

Abstract

The accurate and reliable estimation of the State-of-Charge (SOC) and State-of-Health (SOH) of Li-ion batteries is paramount to the safe and reliable operation of any electrified vehicle. Not only is accuracy and reliability necessary, but these estimation techniques must also be practical and intelligent since their use in real world applications can include noisy input signals, varying ambient conditions and incomplete or partial sequences of measured battery data. To that end, a novel framework, utilizing deep learning techniques, is considered whereby battery modelling and state estimation are performed in a single unified step.

For SOC estimation, two different deep learning techniques are used with experimental data. These include a Recurrent Neural Network with Long Short-Term Memory (LSTM-RNN) and a Deep Feedforward Neural Network (DNN); each one possessing its own set of advantages. The LSTM-RNN achieves a Mean Absolute Error (MAE) of 0.57% over a fixed ambient temperature and a MAE of 1.61% over a dataset with ambient temperatures increasing from 10°C to 25°C. The DNN algorithm, on the other hand, achieves a MAE of 1.10% over a 25°C dataset while, at -20°C, a MAE of 2.17% is obtained.

A Convolutional Neural Network (CNN), which has the advantage of shared weights, is used with randomized battery usage data to map raw battery measurements directly to an estimated SOH value. Using this strategy, average errors of below 1% are obtained when using fixed reference charge profiles. To further increase the practicality of this algorithm, the CNN is trained and validated over partial reference charge curves. SOH is estimated with a partial reference profile with the SOC ranging from 60% to 95% and achieves a MAE of 0.81%. A smaller SOC range is then used where the partial charge profile spans a SOC of 85% to 95% and a MAE of 1.60% is obtained.

Finally, a fused convolutional recurrent neural network (CNN-RNN) is used to perform combined SOC and SOH estimation over constant charge profiles. This is performed by feeding the estimated SOH from the CNN into a LSTM-RNN, which, in turn, estimates SOC with a MAE of less than 0.5% over the lifetime of the battery.

Acknowledgements

I would like to express my deepest gratitude to my academic supervisor Dr. Ali Emadi. A great deal of growth and persistence was learned over the years and much of it was driven by his guidance and support. I would like to also extend my profound appreciation to Dr. Matthias Preindl for co-supervising my PhD studies. What seemed like massive research problems always turned into minuscule hurdles after a phone call with Dr. Preindl.

I would like to thank Dr. Phil Kollmeyer for his support on anything related to data and experimental setups. His deep knowledge on experimental battery test benches and battery data always turned our meetings into a fun and enlightening conversation. I would like to also thank Dr. Ryan Ahmed for offering his guidance on all things industry-related.

This research was supported, in part, by the Canada Excellence Research Chairs (CERC) Program. I gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. Financial support was also provided by the Ontario Governments Graduate Scholarship Program. I also would like to thank the McMaster Engineering EcoCAR 3 Team and the Advanced Vehicle Technology Competition for providing a remarkable and unforgettable learning experience.

The long meandering road of a doctorate is never possible without the support of close family and friends. I would like to express my deepest gratitude for my wife, Christelle, for not only offering her tenacious and everlasting love and support but doing so while she was across the Atlantic sea. I would like to thank my parents, Julia and Elias for their incessant love and support. The last few year would not have been bearable without your cooking and weekly phone calls. My deepest appreciation goes to my sisters Daisy, Lara and brother-in-laws for their support as well as my friends Mark and George for the best mix of late night philosophical discussions and Youtube videos.

I would also like to thank my colleagues and friends Maximilian Heinrich, Fabian Kelch and Deqiang Wang for their support inside and outside the lab.

Nomenclature

η Non-linearity or activation function used at output of network layer to produce activations

λ Averaging parameter used to compose inputs for DNN when performing SOC estimation

$\boldsymbol{\Psi}$ Array of inputs pertaining to the reference charge profile

$\mathcal{L}$ Loss function or loss energy of network

$\mathit{\Psi}$ Vector of inputs given to a LST-RNN at predefined sampling frequency

ψ Vector of inputs given to a DNN at predefined sampling frequency

$\widetilde{N}$ Batch length used for LSTM-RNN training, also called depth in time of the network

$\mathit{\Xi}$ Set of reference charge profiles spanning lifetime of battery cell in one ageing study

ξ Reference charge profile used for SOH estimation

b Network layer bias

h	Hidden layer in network
SOC^*	Observable State-of-Charge computed through Coulomb counting, also referred to as ground-truth values of SOC
SOH^*	Observable State-of-Health computed through Coulomb counting, also referred to as ground-truth values of SOH
W	Array of network weights
w	Individual network weight
AI	Artificial Intelligence
AUKF	Adaptive Unscented Kalman Filter
CC	Constant Current
CNN	Convolutional Neural Network
CNN-RNN	Fusion of Convolutional Neural Network and Recurrent Neural Network with LSTM cell
CV	Constant Voltage
DL	Deep Learning
DNN	Deep Feedforward Neural Network
DOD	Depth of Discharge
EKF	Extended Kalman Filter
EOL	End of Life

ESR	Equivalent Series Resistance
ESS	Energy Storage System, also referred to as Rechargeable Energy Storage System
EV	Electric Vehicle
GA	Genetic Algorithm
GPU	Graphical Processing Units
HEV	Hybrid Electric Vehicle
HWFET	Highway Fuel Economy Cycle
ICA	Incremental Capacity Analysis
ICE	Internal Combustion Engine
LA92	Unified Cycle Driving Schedule
LFP	Li-ion battery with LiFePO chemistry
Li-Air	Lithium Air battery
Li-S	Lithium Sulfur battery
LSSVM	Least Square Support Vector Machine
LSTM	Long Short-Term Memory cell
LSTM-RNN	Recurrent Neural Network with Long Short-Term Memory
MAE	Mean Absolute Error

MAX	Maximum Absolute Error
ML	Machine Learning
NCA	Li-ion battery with $LiNiCoAlO_2$ chemistry
NiMH	Nickel metal hydride batteries
NMC	Li-ion battery with LiNiMnCo chemistry
NN	Neural Network
OCV	Open Circuit Voltage
OEM	Original Equipment Manufacturers
PDF	Probability Density Function
PHEV	Plug-In Hybrid Electric Vehicle
R-RC	Equivalent circuit model with internal resistance element and one RC pair
R-RC-RC	Equivalent circuit model with internal resistance element and two RC pairs
RBF NN	Radial Basis Function Neural Network
RMS	Root Mean Squared Error
RUL	Remaining Useful Life
SOC	State-of-Charge, also used to denote the estimated SOC obtained by a network
SOH	State-of-Health, also used to denote the estimated SOH obtained by a network
STDDEV	Standard Deviation of the Error

UAV Unmanned Aerial Vehicles

UDDS Urban Dynamometer Driving Schedule

US06 US06 Supplemental Federal Test Procedure

USABC US Advanced Battery Consortium LLC

XEV General term which includes Hybrid Electric Vehicles (HEV), Plug-in Hybrid Electric Vehicles (PHEV) and Electric Vehicles (EV)

ZEBRA A type of Sodium Nickel battery, also referred to as molten salt battery

Contents

List of Figures

Chapter 1

Introduction

1.1 Major Themes

Recently, Li-ion batteries have found themselves at the intersection of many paradigm shifts and societal movements. These include climate change, stricter emissions regulations, energy security, a growing population, most of which is concentrated in major metropolitan areas, and an increasingly energy conscious market. These factors have enticed a growth in the market share of electrified vehicles and have encouraged newer technologies using Li-ion batteries, like manned and unmanned aerial vehicles, to emerge.

1.1.1 Climate Change and Hazardous Emissions

According to the World Health Organization (WHO), air pollution is responsible for 7 million premature deaths per year worldwide. This represents one eighth of the total mortality rate globally and is estimated to have an annual economic cost of US$1.4 trillion in the European region alone as of 2015. Emissions from the road transport, the residential energy and the power generation sectors are responsible for about half of the annual premature deaths due to air pollution [1]. As a result, some governments are taking action and proposing new regulations which would render new petrol and gas powered vehicles illegal by 2030 or, as early as 2025 for Norway. One notable method to mitigate these chronic issues calls for the mass adoption of not only electrified vehicles but, also, grid storage for load-leveling. This argument is further strengthened when considering the much higher efficiencies offered by electrified vehicles [2; 3] and the massive cost savings offered by peak shaving through energy storage systems [4; 5]. Automotive Original Equipment Manufacturers (OEM) are responding to the tighter regulations and to a growing energy conscious generation with greener propulsion systems. Nowadays, Li-ion batteries have a growing dominance in the electrochemical energy storage market as a result of their high specific energy, good cycle life and low self-discharge rates [3]. Li-ion batteries are not only used in electrified vehicles and grid-storage but also most portable electronic devices, and in newer technologies like Unmanned Aerial Vehicles (UAV) and passenger drones aimed for medium to short range distances otherwise referred to as air taxis.

1.1.2 Energy Security

In recent decades, there has been a rising concern over the security of energy supply or more popularly referred to as energy security. The topic of energy security became important during World War II, when the survival of a nation within the war hinged on the supply of fuels and petroleum to their armies. Eventually, with the widespread use of petroleum products, not only were military organizations dependent on oil but whole nations became dependent on oil for transportation as motor vehicles became ubiquitous, for power generation, heating, manufacturing, food products, etc. Thus, nowadays, there are several connotations to energy security albeit, here, the protection of the economy from energy shortages is discussed. These shortages can also be the catalyst for surging energy prices due to energy scarcity which can negatively affect a nation's economy [6; 7]. Early on, most developed nations did not produce sufficient oil nationally to appease the growing societal dependence on it. The economy of those countries rich in fossil fuels became polarized towards revenues from oil exports. This polarization, as would be seen on countless occasions, did not promote political stability and economic growth. To mitigate these issues, it becomes important to have a significant energy mix which includes more electrified vehicles and renewable energy sources integrated into the grid. Countless nations have and are still implementing incentivized programs to encourage usage of renewable energies, electric vehicles and biofuels [8; 9].

1.1.3 Electrified Vehicles and their Declining Cost

Electrified vehicles which include Hybrid Electric Vehicles (HEV), Plug-in Hybrid Electric Vehicles (PHEV) and Electric Vehicles (EV) which, in general, are referred to as XEV, have the added benefit of an electrified powertrain, offering much greater efficiencies than its counterpart; the Internal Combustion Engine (ICE). In addition to being more efficient, electrified powertrains are able to recapture breaking energy that is otherwise wasted through mechanical breaks. The primary components in an electric propulsion system are the motor/generator, the battery pack or Energy Storage System (ESS), and the power electronics drive system. The latter ensures the energy conversion and transfer between the electric machine and the ESS [2; 10]. Battery technology is one of the major barriers to the mass market adoption of XEV. As such, battery technologies are being extensively researched nowadays and the industry's landscape is rapidly changing.

The higher cost of oil in recent history has prompted many consumers to reconsider their opinions on XEVs since electricity is cheaper. As mentioned previously, an electrified vehicle's battery can be one of the costliest components in the vehicle. Until recently, this has not allowed manufacturers to price EVs in the realm of what is considered affordable by the mass market, while retaining a similar driving range as conventional vehicles. Nevertheless, recently, with the rapidly declining cost of batteries amongst other factors, major OEMs as well as new entrants like Tesla Inc.

have been racing towards an affordable offering having sufficient range and mass-market appeal.

1.1.4 Barriers and A Changing Market Perspective

Although XEVs offer significant benefits over their ICE counterparts, some barriers to mass market penetration still exist. These include battery cost, range anxiety, reliability and safety. Battery costs is one of the main reasons why the cost of EVs has been prohibitive to mass-market penetration. However, as stated before, this is currently changing and more affordable EV offerings are beginning to enter the market. This inflection point can be partly attributed to the reduction in battery costs which is falling faster than some expected. In 2015, one study [11] showed that contrary to some estimates of the cost of batteries (about $500-$1000/kWh), the cost for the industry as a whole and for EV manufacturers is $410/kWh and $300/kWh, respectively. The study further states that with continued investment in large-scale battery production facilities and in government-run incentives programs, the cost of $200/kWh at a volume of 100,000 battery packs annually, can be achieved in the short-term. This increasingly reduces the gap between the actual cost for batteries and the $100/kWh cost which the US Advanced Battery Consortium (USABC) would like to see by the year 2020 [12].

Although some early adopters of EVs might have been influenced by their environmental impacts or the lack thereof, currently consumers are much more concerned about the reliability and performance, e.g. driving range and cost savings [13]. One study [14] shows that consumers were willing to pay $35 to $75, for every additional mile of range and a premium of $6000 - $16000 over conventional petrol-powered vehicles, for an EV. The rapidly falling cost of batteries and increased offerings of affordable EVs are also big factors in the changing perspectives of the public vis-a-vis EVs.

1.2 State-of-Charge and State-of-Health

Given the amount of energy they can hold, safe and reliable operation of Li-ion batteries needs to be paramount. To this end, an Energy Management System (EMS) is used to maintain the safe and reliable operation of the battery. This includes balancing of the cells, protection to ensure operation within safe limits of the battery, and estimation of the State-of-Charge (SOC) and State-of-Health (SOH) of the battery.

A reliable state-of-charge estimation is required to ensure an accurate gauge of a vehicle's remaining driving range as well as proper balancing of the battery pack [3; 15; 16]. The SOC is analogous to the fuel gauge of petrol-powered vehicles. SOC is defined as the remaining charge within the battery and is defined as the ratio of the residual capacity of the battery to its nominal capacity [3]. Figure 1.1 shows a diagram of the state-of-charge of a battery cell as it is discharged as well as a typical plot of the cell voltage as a function of SOC. Due to unpredictable driving habits and the repeated acceleration and deceleration of a vehicle, the battery can be exposed to highly dynamic load demands. As a result of these dynamic load demands, SOC estimation is a tedious task. SOC is not an observable quantity, therefore its accurate estimation becomes essential for reliable and safe operation of the vehicle[17; 15].

The relationship between the battery's observable signals to the estimated SOC is a highly non-linear one, varying with temperature and discharge/charge currents

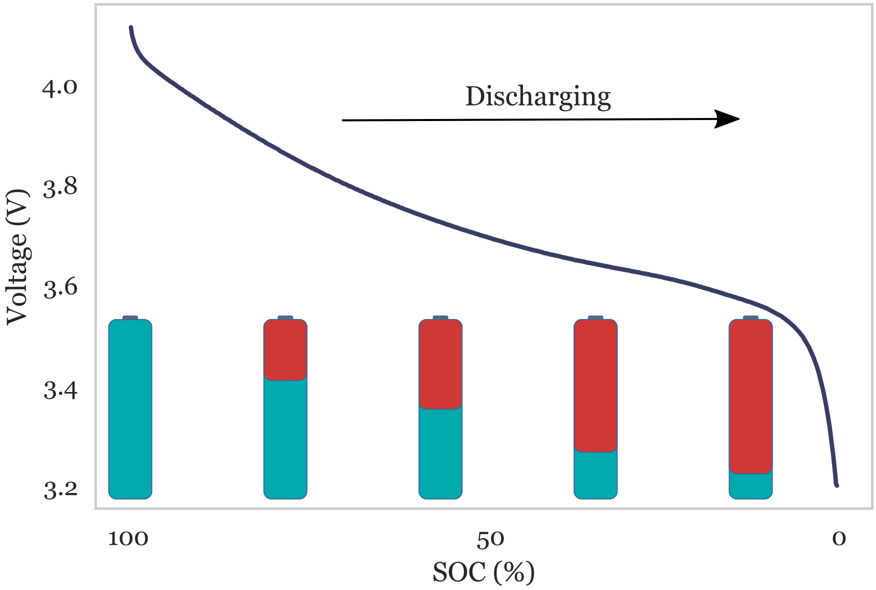


Figure 1.1: Diagram describing the State-of-Charge (SOC), starting from a fully charged cell where SOC=100% to an empty cell where SOC=0%.

[18; 19]. Traditionally, the two main estimation methods have used *open circuit voltage* based techniques and *coulomb counting* [17; 20]. These methods are known to have their limitations and have been generally displaced by more sophisticated methods. Typically, these use an (adaptive) cell model as well as voltage, and current measurements to issue an estimate. They include Luenberger observer [17; 21], adaptive observer [17; 22], sliding mode observer [17; 23; 24], and Kalman filters [25; 26; 27]. These techniques often require tedious model identification to adequately represent the non-linear behavior of a battery and they tend to contain many partial differential equations, which can be computationally expensive. In addition, they often require large numbers of parameters to perform SOC estimation at varying ambient conditions.

Like most things, Li-ion batteries age with time; a process underpinned by degradation of electrode materials, loss of lithium in active carbon, lithium metal plating and chemical decomposition, to name of a few. If aging is not correctly and accurately monitored, premature failure can occur which, typically for Li-ion batteries, are catastrophic in nature. Some recent reported incidents with Li-ion battery powered devices are examples of such premature failures. Therefore, accurate aging estimation methodologies are dire to the reliable and safe operation of these batteries.

Battery aging is typically determined through a State of Health (SOH) estimation which has a value ranging between 0 and 100%. It is a figure of merit that does not correspond to a physical entity and is given in per unit. A SOH of 100% represents the battery health at the start of a battery's lifetime where capacity is at it's highest and 0% would define the SOH observed at the end of the battery's lifetime, as shown in Figure 1.2. In some industries like the automotive industry, the standard for end of

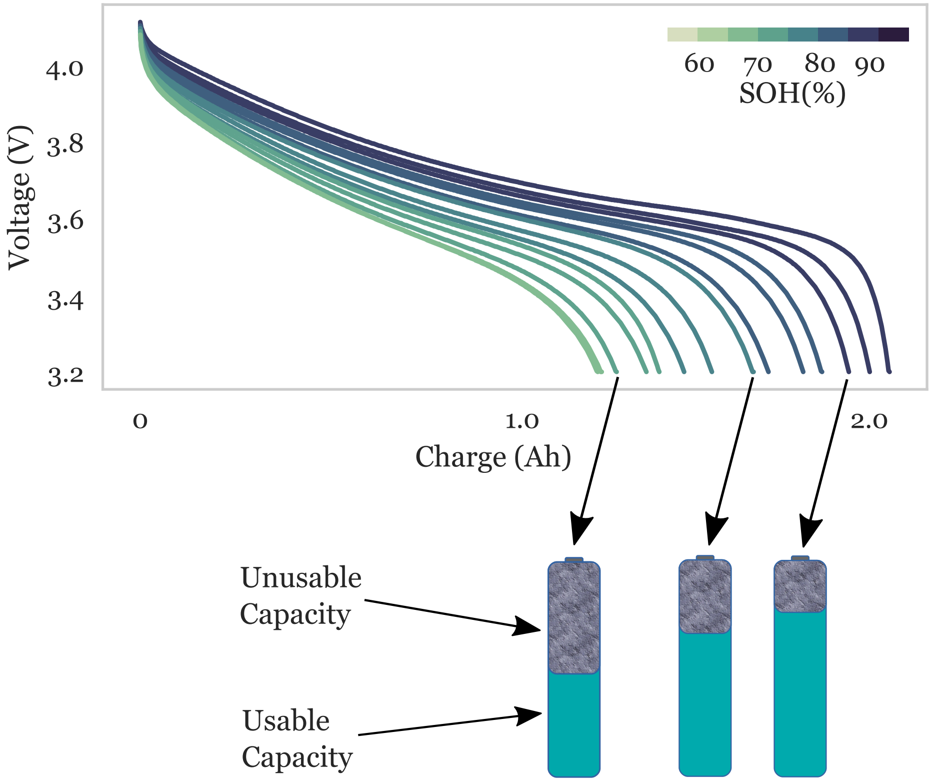


Figure 1.2: Diagram showing battery aging and how this is defined by the State-of-Health (SOH). (Top) Plots of battery cell voltage as a function of available capacity in Ah. (Bottom) When the battery ages and SOH declines, the usable capacity in a battery cell also diminishes.

life of a Li-ion battery is when an SOH of 80% is reached. SOH is not an observable quantity and is a highly non-linear entity, dependent, in large part, on the volatility of loading profiles, ambient temperature, Depth of Discharge (DOD) and self-discharge, to name a few [28]. Therefore, an accurate estimate of SOH is as tedious to estimate as it is critical to the safe operation of the battery. If a pack reaches the lowest SOH, which can be zero or 80% in the case of the automotive industry, the pack needs to be replaced to ensure a minimum capability, e.g. the range of an EV, and safety. However, a pack with the same characteristics can be perfectly suitable for a less demanding application, e.g. as an energy storage element in a smart home, and batteries can be reused in what is known as second-life operation [29].

SOH can be determined with respect to capacity or internal resistance. Both versions of the value are calculated while taking one or more battery parameters into account [30; 17; 31; 32]. Most methods estimating capacity-based SOH will utilize capacity as the main parameter, however some will also consider voltage, self-discharge and/or temperature to more accurately capture the energy capability.

1.3 Deep Learning Models in the Context of SOC and SOH

Since, the SOC and SOH of the battery are not directly observable quantities, methods are consistently created to achieve further accuracy and dependability of the estimation technique. Not only is accuracy and reliability necessary, but these estimation techniques should also be practical and intelligent since their use in real world applications can include noisy input signals, varying ambient conditions and incomplete or partial sequences of measured battery data which, typically are not encountered in laboratory settings. Reliable and accurate SOC and SOH estimation hinges on reliable and high fidelity battery models as well as estimation strategies with good performance. In this research work, a novel framework, utilizing deep learning techniques, is considered whereby battery modeling and estimation are performed in a single unified step. Furthermore, it is shown that the parametrization process becomes automated with the use of these learning algorithms removing the need to manually handcraft and parametrize battery models.

Using data-driven and neural computation-based techniques offers a few advantages over more traditional approaches. They have flexible expressibility due to their non-parametric nature; where the amount of network weights and layers can be naturally adjusted to meet the complexity of the modelled system. Deeper networks are also shown to be more efficient; being able to retain the same expressibility as the shallower networks but with a smaller network size or, in other words, with fewer weights. Since, most works which look at SOC and SOH estimation, typically rely on measurements obtained in isolated and controlled environments, this work puts these deep learning estimation techniques through a complete arsenal of validation test cases, most of which are aimed to mimic real world scenarios. These include additional noise, offsets and gains to battery measurements which, interestingly, not only robustifies the algorithms but also improves their accuracy through a process called jittering. Universality is further increased by training and testing the algorithms through various ambient temperatures and, in the case of SOH estimation, different ranges of reference profiles.

Furthermore, looking ahead to the future, using data-driven and neural computation-based algorithms for SOC and SOH estimation as well as other battery state estimation can be very beneficial for increasingly autonomous vehicles. In a world where vehicles are increasingly automated, and streams of data are being collected on every subsystem of the vehicle including the battery, data-driven approaches, like deep learning, used to perform SOC and SOH estimation become very important when optimizing routes based on the vehicle's energy consumption in various terrains and ambient conditions.

1.4 Thesis Structure

Implementing deep learning techniques for SOC and SOH estimation on Li-ion batteries is attempted for what is seemingly the first time in this research work. Some attempts have been made in the past using traditional machine learning models but have had little success. Where the networks failed to deliver sufficient accuracy, filters and inference systems were used to compensate and achieve better results. Therefore, given its unlikeness in the field, many topics discussed are rooted in machine learning and can be unfamiliar to those involved with battery research. As a result, this thesis is compiled with the intention of also acting as a road-map for first-time users of the deep learning methods applied for SOC and SOH estimation. It is important to note that there are no claims made in this thesis on the creation of the deep learning techniques used in this research work, however, their application on State-of-Charge and State-of-Health estimation for batteries is novel and useful for anyone involved in battery research.

In Chapter 2, a short introduction is given for rechargeable batteries and a review is conducted on modern battery cell technologies as well as their applications. Further details are provided on the progress of advanced lithium-based battery chemistries. A rigorous derivation of the theory behind all the techniques utilized in this research work is given in Chapter 3 and a detailed account of how this theory is particularly beneficial to SOC and SOH estimation is conducted in Chapter 4. For SOC estimation, this thesis, proposes two different deep learning algorithms; a Recurrent Neural Network with Long Short-Term Memory (LSTM-RNN) and a Deep Feedforward Neural Network (DNN). The latter both have good estimation accuracy nevertheless, there are trade-offs between both approaches in terms of requirements of input data and ease of implementation. The results from both of these techniques are presented in Chapter 5 and Chapter 6, respectively. The results from the Convolutional Neural Network (CNN) used to perform SOH estimation are presented and discussed in Chapter 7. Two sets of results are presented; one for fixed reference charge profiles and the other for partial reference curves. To extend this latter study further, Chapter 7 includes combined SOH and SOC estimation results using a dual CNN and LSTM-RNN fusion network. Finally, a concluding statement is given in Chapter 8 and a brief discussion is provided on the future works which can be performed in this relatively new research area.

Chapter 2

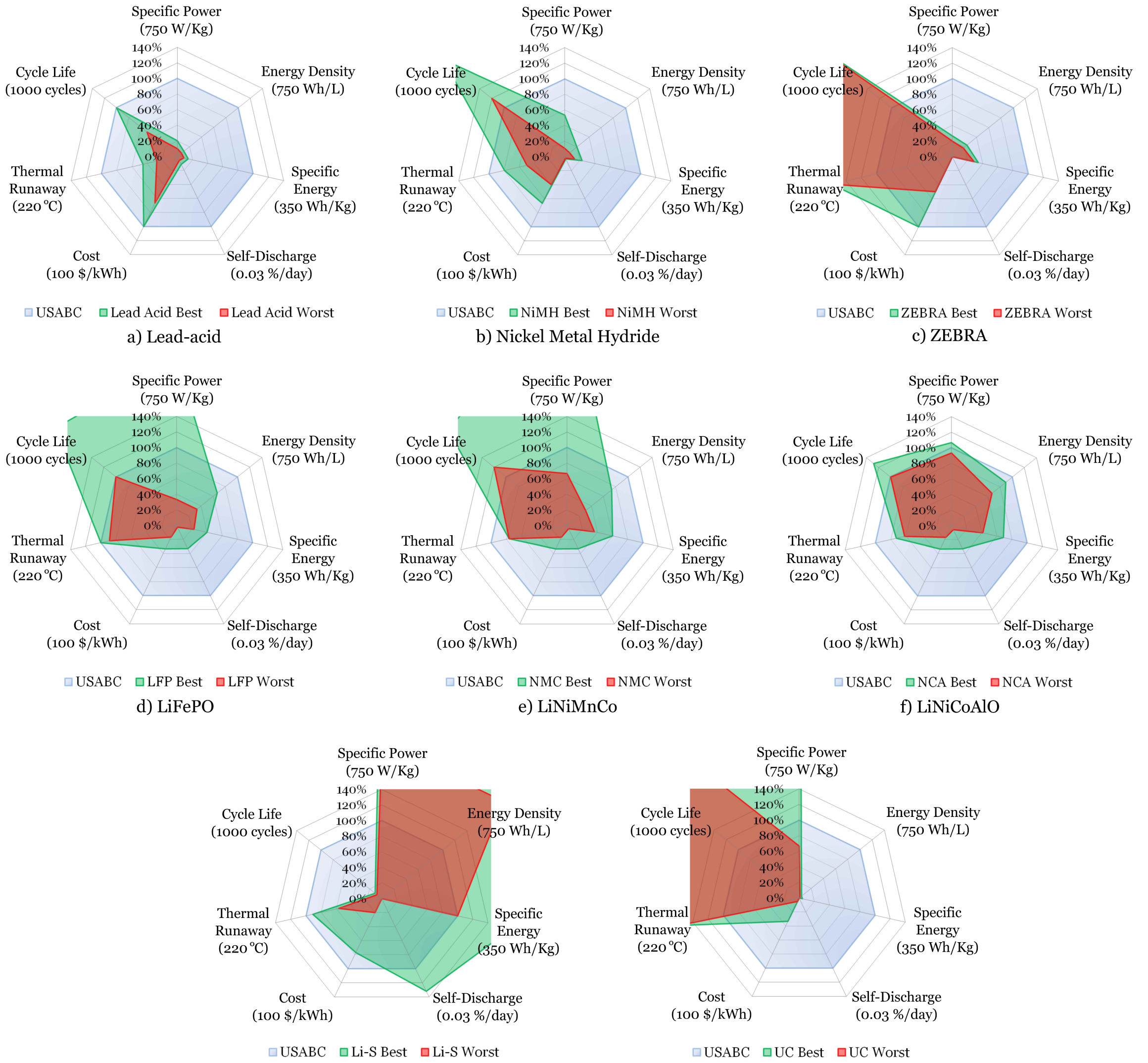
Electrochemical and Electrostatic Energy Storage and Management Systems in Electrified Vehicles

2.1 A Brief History of the Battery

The term battery was first coined by Benjamin Franklin in 1760, when he created an array of glass jars which can store electrical charge. The term battery was inspired by a military term which describes multiple weapons operating together. It wasn't until later, in 1799, that Allesandro Volta was able to store charge in what is called the Volta pile which used two dissimilar metals separated by a brine-soaked cloth. Although the Volta pile was a major discovery which interested even the likes of Napoleon Bonapart, it was not rechargeable. In 1859, a French physicist by the name of Gaston Planté, invented the lead-acid battery which was the first rechargeable electric battery. Although electric cars were created earlier using non-rechargeable batteries, it wasn't till Planté's invention and Faure's improvements to the invention that companies started building electric cars at greater numbers in the mid to late 1800'.

2.2 Modern Cell Technologies and their Applications

Nowadays, most research efforts in battery technology is geared towards increasing specific energy and reducing cost while maintaining a high cycle life. The U.S. Advanced Battery Consortium (USABC) has set specific goals to be reached by 2020 for pure EV battery packs. These include a specific energy, at cell level, of 350 Wh/kg, a cost of $100/kWh at production volumes of 100,000 cells and a cycle life of 1000 cycles.



a) Lead-acid

b) Nickel Metal Hydride

c) ZEBRA

d) LiFePO

e) LiNiMnCo

f) LiNiCoAlO

g) Lithium-Sulfur

h) Ultracapacitor

Figure 2.1: Spider plots of prevalent battery technologies

Table 2.1: Existing battery and ultracapacitor specifications

Specifications	Augmented USABC	Lead Acid	Nickel Metal Hydride	ZEBRA	Ultracapacitor
Specific Power (W/Kg)	700	75 - 150 [33]	80 - 400 [33]	150 - 200 [34; 35]	500 - 100,000 [36; 37]
Energy Density (Wh/L)	750	50 - 80 [36]	60 - 150 [36]	135 - 180 [34; 36]	10 - 30 [36]
Specific Energy (Wh/Kg)	350	30 - 50 [36; 37]	45 - 80 [38]	100 - 120 [34; 36]	2.5 - 15 [36]
Self-Discharge (%/day)	0.03	0.29 - 0.57 [33]	1 - 1.43 [39; 33]	15 [36]	20 - 40 [36]
Cost ($/kWh)	100	100-150 [40]	150 - 250 [2; 38]	100 - 200 [36]	300 - 2000 [36]
Cycle Life (cycles)	1000	500 - 1000 [36]	1200 - 2000 [41]	>2500 [36; 37]	>100,000 [36; 37]
Thermal Runaway (^{o}C)	220 [a]	60 - 100 [42; 43]	110 - 175 [37]	- [b]	- [c] [44]

[a] Added to USABC goals

[b] ZEBRA batteries operate at temperatures larger than 270 °C

[c] Cell vented before thermal runaway onset temperature was reached

Table 2.2: Li-ion battery specifications

Specifications	Augmented USABC	$LiFePO_4$	LiNiMnCo	$LiNiCoAlO_2$	Li-S
Specific Power (W/Kg)	700	250 - 1600 [33]	500 - 2400 [33]	700 - 800 [33; 45]	1500 - 3000 [46]
Energy Density (Wh/L)	750	250 - 500 [36; 47]	230 - 550 [36; 47]	500 - 670 [36; 48]	1500 - 2600 [36; 48]
Specific Energy (Wh/Kg)	350	80 - 140 [45; 33]	126 - 210 [45; 47]	145 - 240 [49; 48]	350 - 600 [46; 47]
Self-Discharge (%/day)	0.03	0.1 - 1.29 [36; 33]	0.1 - 0.71 [36; 33]	0.1 - 0.57 [36; 33]	0.025 - 32 [50]
Cost ($/kWh)	100	300 - 600 [11]	300 - 600 [11]	300 - 600 [11]	>130 [a] [48]
Cycle Life (cycles)	1000	1000 - 2000 [45]	1200 - 1950 [45; 51]	1000 - 1280 [45; 49]	80 - 110 [45; 51]
Thermal Runaway (^{o}C)	220 [b]	195 [52]	168 [52]	136 - 160 [53]	125 - 200 [54]

[a] Estimation based on cited work

[b] Added to USABC goals

In this section, modern cell technologies are considered and compared. Each cell chemistry is evaluated on 7 different metrics and compared to the USABC 2020 goals, shown in Figure 2.1. Safety as a performance metric is not particularly outlined in the USABC goals however this is a critical metric in light of the recent incidents associated with batteries. Therefore, the USABC goals have been augmented to include the onset temperature of thermal runaway for each cell technology as an indicator of safety [55; 53; 52; 42; 56; 57; 58; 59]. A score of 0% to 100% for each metric is given to the cell technology depending on how well it meets the augmented USABC goals. The metrics used are energy density (Wh/L), specific energy (Wh/Kg), specific power (W/Kg), self-discharge (% of initial cell capacity per day), cost ($/kWh), cycle life (cycles) and safety. A summary of each cell chemistry's performance and specifications is given in Table 2.1 and in Table 2.2. The data outlined in these two tables which are also illustrated in Figure 2.1, were obtained after an extensive investigation of existing battery/UC technology literature. Particular attention was mostly paid to literature which obtained experimental results for each one of the seven metrics considered in this section. For each metric of each battery/UC technology, there typically exists more than one experimental value reported in literature. The maximum and minimum values for each one of these metrics for each cell technology are outlined in Table 2.1 and in Table 2.2. These minimum and maximum values are used to generate the green and red plots, shown in Figure 2.1, to portray a technology's best and worst performance, respectively.

2.2.1 Nickel Metal Hydride Batteries

Nickel metal hydride (NiMH) batteries, whose performance data is shown in Figure 2.1b) and in Table 2.1, have been used in HEVs for 14 years. Prominent cell manufacturers of NiMH include PEVE and Sanyo Electric (Panasonic). The technology has been well developed and has shown durability in vehicles such as the NiMH RAV4 EV which has been in operation for over 10 years [60]. Electrochemically, the negative electrode is hydrogen that is absorbed/released in an intermetallic compound. The positive electrode is composed of nickel metal and the electrolyte is typically an aqueous solution containing potassium hydroxide [61]. Commercially available cells are manufactured in both cylindrical and prismatic hard case variants and span a voltage of 1.2-1.35 V [61]. Coulombic efficiency is about 10% less then that of lead-acid batteries [61], however NiMH power/energy capabilities are far greater; typically factors of two to three times the power/energy density of lead-acid. NiMH technology's shortcomings lie in their significantly higher self-discharge rate [61] which does not allow them to be ideal candidates for energy-oriented applications such as PHEVs and EVs. Recent advances in NiMH include the usage of bi-polar cell designs to achieve higher power capabilities [62].

2.2.2 ZEBRA Batteries

The origins of the Sodium Nickel battery, also called the ZEBRA battery, are traced back to South Africa, where the first patent was awarded for this technology in 1978 [63]. In 1984, the first electric vehicle to be powered by a ZEBRA battery was driven in Derby, UK. The technology has been in development since then and has matured. Commercially available ZEBRA batteries, whose performance data is shown in Figure 2.1c), are based on sodium nickel chloride (Na-Ni-Cl) electrochemistry where sodium is the negative electrode and nickel chloride is the positive electrode. ZEBRA batteries operate at a temperature of 270°C to 350°C [64] since sodium is found in its molten state at these temperatures. As a result these batteries are more commonly referred to as ”molten salt” batteries. The cells are manufactured in an upright rectangular box format. Special pack design and thermal considerations are needed to maintain the high operating temperature. As a result the energy required to maintain operating temperatures result in a self-discharge. Some advantages of this technology include a relatively high specific energy of 90-120 Wh/kg [34; 36] shown in Table 2.1, their insensitivity to ambient temperature and fault tolerance. The latter is due to the cells having a low resistance short-circuit state when cell damage occurs, allowing continuous operation during cell failures [34; 63]. This makes them operationally safe and good candidates in harsher climates. In addition, they have a near 100% Coulombic efficiency [65; 63] and a cycle life measured to be over 2500 cycles [63; 37]. These batteries have better energy density and specific energy than NiMH and their power capabilities and pack cost are less. However, ZEBRA batteries loose charge to maintain the elevated internal temperature of the battery, which maintains the molten state of the salt. This effect can account for a loss of about 15% [36] of its capacity per day. Applications which have employed ZEBRA batteries include European EVs such as Think EV, Iveco Electric Daily and Modec EV vans.

2.2.3 Li-ion Batteries

Battery technology has come a long way in the last few decades. Li-ion battery technology powers most mobile applications; from smartphones and laptops to, nowadays, energy storage systems for electric drive vehicles and smart-grid applications. In the case of XEVs, Li-ion batteries are energy dense and therefore allow the vehicle to have a longer electric-only driving range however they typically do not have the specific power to provide or accept large power spikes resulting from the dynamic power profile of a vehicle. During a regenerative braking scenario, Li-ion batteries are limited in terms of the power levels which they can handle therefore a portion of the regenerated energy can be dissipated through the mechanical brakes. Under highly dynamic power profiles, the battery of a XEV can be greatly overstressed which negatively affects the longevity of its lifespan. Highly dynamic load profiles imposed on the battery pack induces degradation at the cell level which leads to increased internal resistance. Capacity fade is also a consequence of this phenomenon which most often results in premature cell End of Life (EOL) or even premature failure [66].

There are various materials used in the construction of Li-ion cell electrodes. The cathode materials are usually oxide variants of lithium metal amalgams which usually contain either manganese (LMO), cobalt (LCO), nickel (LNO), iron-phosphate (LFP), or mixtures thereof such as LiNiMnCo (NMC) and $LiNiCoAlO_2$ (NCA); containing an aluminum blend [67]. Anode materials are typically graphite, although hard carbon, silicon-carbon compounds, lithium titanate (LTO), tin or cobalt alloys and silicon-carbon blends have also been used in consumer electronics. In all varieties, lithium ions transport back and forth between the electrodes and transfer electrons in an intercalation based reaction instead of a traditional molecule-to-molecule chemical reaction. The benefits of Li-ion technology is higher cycle life [45; 49], high Coulombic efficiency (up to 98%) [67], and low self-discharge [33; 36]. A wide variety of electrolyte materials are also possible from solid-based to liquid-based, usually of the organic non-aqueous form. The usage of lightweight materials and high voltage potential of lithium-ion electron transfer give rise to high power/energy densities and some of the highest electrochemical cell nominal voltages, e.g. 3.2-3.8V [62; 68]. The LiFePO (LFP) variant, whose spider plot is shown in Figure 2.1d), is considered to be among the safest Li-ion cell chemistries due to it's higher thermal runaway temperature [67]. Meanwhile, NMC and NCA cells are dominating the EV market nowadays given their stronger power/energy performance when compared to other technologies, as can be observed in Figure 2.1e)f) and Table 2.2. As a result, companies such as Panasonic, Tesla, LG Chem and Samsung SDI are heavily investing in these two cell chemistries.

Nowadays, Li-ion chemistries are being extensively researched and developed to significantly increase energy and power capabilities as well as operating voltage. For example, improvements in NMC cathodes have shown an increase in operating voltage from 4.13 V to 4.3 V [69]. In addition, operating voltages of up to 4.7-4.8 V have been observed in lithium vanadium phosphate (LVP) cathodes which have been integrated into the Subaru 64e prototype [70]. Although, some members of the research community are focusing on incremental improvements in conventional Li-ion chemistries, many others are betting on a leap in technology embodied by the next generation Li-based battery cells that have the capability to significantly outperform conventional Li-ion cells.

2.2.4 Advances in Li-based Battery Technologies

Conventional Li-ion battery technology has a theoretical specific energy of 387 Wh/kg [71; 72]. Commercial cells manufactured nowadays are approaching a specific energy of 240 Wh/kg, [68] thus current Li-ion battery technology can be reaching its limits.

If mass marketed EVs are to ever reach a driving range of over 500 km per charge, the dawn of a new age in commercial battery technology must come about.

Li-Air (Li_2O_2) and Li-S chemistries have been gaining a great deal of interest from the research community primarily due to their very high theoretical specific energy of 3582 Wh/kg and 2567 Wh/kg, respectively [72; 73]. In addition, Li-Air and Li-S cathodes are composed from abundantly available materials such as O_2 and elemental sulfur, respectively, which would render these cells cheaper to manufacturer. Both of these battery technologies possess the same metallic Lithium anode where Li is oxidized when a load is observed in the external circuit. Lithium ions then travel across the electrolyte to reduce oxygen or elemental sulfur in the cathode of Li-Air and Li-S cells, respectively. Much of the increased theoretical energy density of these batteries is a consequence of its pure metallic Lithium anode which can hold more charge than lithiated graphite anodes per unit mass found in traditional Li-ion batteries [72; 74]. Furthermore, the cathode in readily available Li-ion batteries such as the cathode in the $LiCoO_2$ cells store less lithium than Li_2O_2 or Li_2S cathodes [72; 74].

In Li-Air cells, Li ions react with O_2^{-2} which is reduced from O_2 in surrounding air. Initially, unwanted discharge products form due to Li ions reacting with other molecules within air such as CO_2. As a result, Li-Air cells are wrapped with various types of membranes to increase their permeability to O_2. A key and important factor inhibiting commercialization of these cells is electrolyte degradation giving rise to poor lithium cycling efficiencies and capacity fading [75; 76; 72; 74]. Currently, research on Li-Air cells shows as much as 50% capacity fade after only 20 discharge cycles [72].

Similarly, Li-S cells suffer from poor sulfur cycling efficiencies which lead to fast capacity fading [46; 72]. In addition, sulfur is a good insulator thus leading to poor electrode kinetics and limited discharge rates [72; 74; 77]. Currently, Li-S prototype cells experience as much as 50% capacity fade after 50 discharge cycles [72]. This and other performance specifications of Li-S are shown in Figure 2.1g) and in Table 2.2. Sion Power is one of a few privately held companies which is developing this technology. They currently claim having developed Li-S cells with a specific energy of 350 Wh/kg and predict that this value will increase to 600 Wh/kg in the future [78].

Some advances have been made recently to accelerate the advent of Li-Air and Li-S batteries. Lately, these studies have focused primarily on replacing traditional Carbon-S cathodes with Graphene-S cathodes. Graphene has a high electrical conductivity and has a large surface area which can lead to improved cycle life [79; 71; 80; 81; 82; 83]. Using Graphene, a capacity fade of under 30% after 100 cycles is being reported by some studies [71]. Recently, silicon is also being considered as an alternative to graphite-based anode materials to address the issue of capacity fading in Li-based battery technology [84; 85; 86; 87]. One study reported a capacity fade of only 3% after 1000 cycles by constructing the electrode from pomegranate-shaped silicon-carbon structures [87].

2.3 Energy Management Systems for Electrified Vehicles

An energy management system (EMS) is a system for *protection*, *estimation*, and *control* of a battery, ultracapacitor, or hybrid ESS to ensure safe high-performance operation and a long lifetime [88; 89]. ESS consist of multiple cells that are connected in series and/or in parallel. Cells are connected in series to obtain a rated voltage of the pack. Parallel connections increase the rated capacity of a pack such that the required amount of charge, i.e. energy can be stored. Also, adding parallel cells increases the rated current, i.e. power, of a pack.

Large packs for XEV consist of hundreds or thousands of cells. To simplify the arrangement, packs are divided into modules [90]. Modules deal with the EMS functionality on a cell level and provide abstraction for a supervisory control. Communications are typically implemented using a fieldbus system, e.g. CAN for automotive.

2.3.1 Li-ion Battery Protection

ESS store significant amounts of energy and require *passive* and *active* safety precautions to prevent an uncontrolled release. Although electromagnetically driven contactors are used as a form of last line of protection at the pack level, passive and active safety mechanisms within the pack are used to ensure the continued safe operation of the battery. Passive safety mechanisms are used to prevent worst case scenarios in case of electrical faults, high temperature, or high internal pressure [91]. Each cell is typically protected against overcurrents using positive temperature coefficient resettable fuses and short-circuits using bi-metallic circuit breakers [90]. Most battery cells also feature overpressure release valves. Modules are designed considering the expected physical abuse (vibration, impact, etc.), and thermal isolation of the cells e.g. to prevent the progression of a thermal runaway [91].

Energy management systems are used to actively protect ESS. Battery and ultracapacitor cells are designed to operate in a specified voltage and current range. Energy storage cells operate typically in the range $v_{\min} = 2.5\text{V}$ to $v_{\max} = 4.3\text{V}$ for

Li-Ion batteries [68]; up to $v_{\max} = 2.7$V for electrostatic double-layer capacitors [92]; and $v_{\min} = 2.2$V to $v_{\max} = 4$V for hybrid ultracapacitors [93]. Cells are designed to handle a rated current but the minimum and maximum current depends on temperature. Electrostatic cells need to be derated when they find themselves above a certain temperature. The temperature dependence of electrochemical cells is nonlinear e.g. Li-Ion cells cannot be charged below a certain temperature.

Safe operation challenges arise when connecting cells in series (and parallel) to form a pack. Cells do not charge and discharge equally due to variations in intrinsic cell specifications like internal impedance and self-discharge rate. These variations are caused by manufacturing tolerances, cell degradation and temperature variations across a pack [94]. The EMS has to monitor each cell individually. State-of-charge and state-of-health estimation is also necessary to monitor the amount of charge in the battery and its degradation accurately. Each cell has to be operated strictly within its operation limits to avoid gradually damaging some of the cells. Operation of a battery pack without protecting each cell corresponds to a nearly exponential reduction of the battery life as the string length increases [95; 96]. In contrast to passive protections, fine-grained safety mechanisms can be implemented in the EMS, e.g. taking temperature-dependent current limits into account or communicate with chargers and loads to adjust the operation.

2.3.2 Battery Modeling

The work in this thesis targets a new way to perform battery modeling and state estimation in a single unified step. Therefore, it becomes important to understand what underlying non-linear systems the deep learning algorithms are trying to model

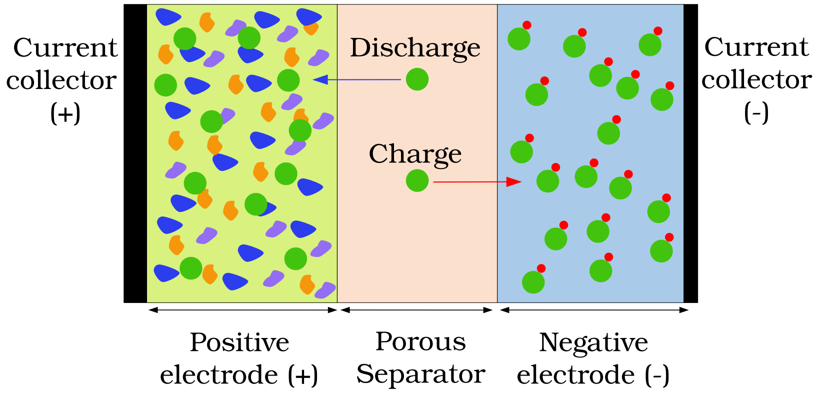


Figure 2.2: Overview of Li-ion battery cell operation. Porous separator divides the positive electrode from the negative one and is permeable to Li-ions but impermeable to electrons.

or express in order to achieve SOC and SOH estimation.

A battery cell, shown in Figure 2.2, is composed of an anode and a cathode immersed in an electrolyte and separated by a porous membrane called a separator. An oxidation reaction of the following type occurs at the anode

$$A^{x+} - ne \rightarrow A^{(x+n)+} \tag{2.1}$$

Each oxidation reaction occurring between the electrolyte and the anode transfers ne electrons from the electrolyte phase to the anode phase giving the anode its negative charge. In order to have charge flowing in the external circuit, connecting the anode to the cathode, electro-neutrality must be established. Electro-neutrality requires a second reducing reaction to occur at the cathode, where electrons are transferred in the opposite direction; from the cathode phase to the electrolyte phase. The reducing reaction has the following form

$$C^{(x+m)+} + me {\rightarrow} C^{x+} \tag{2.2}$$

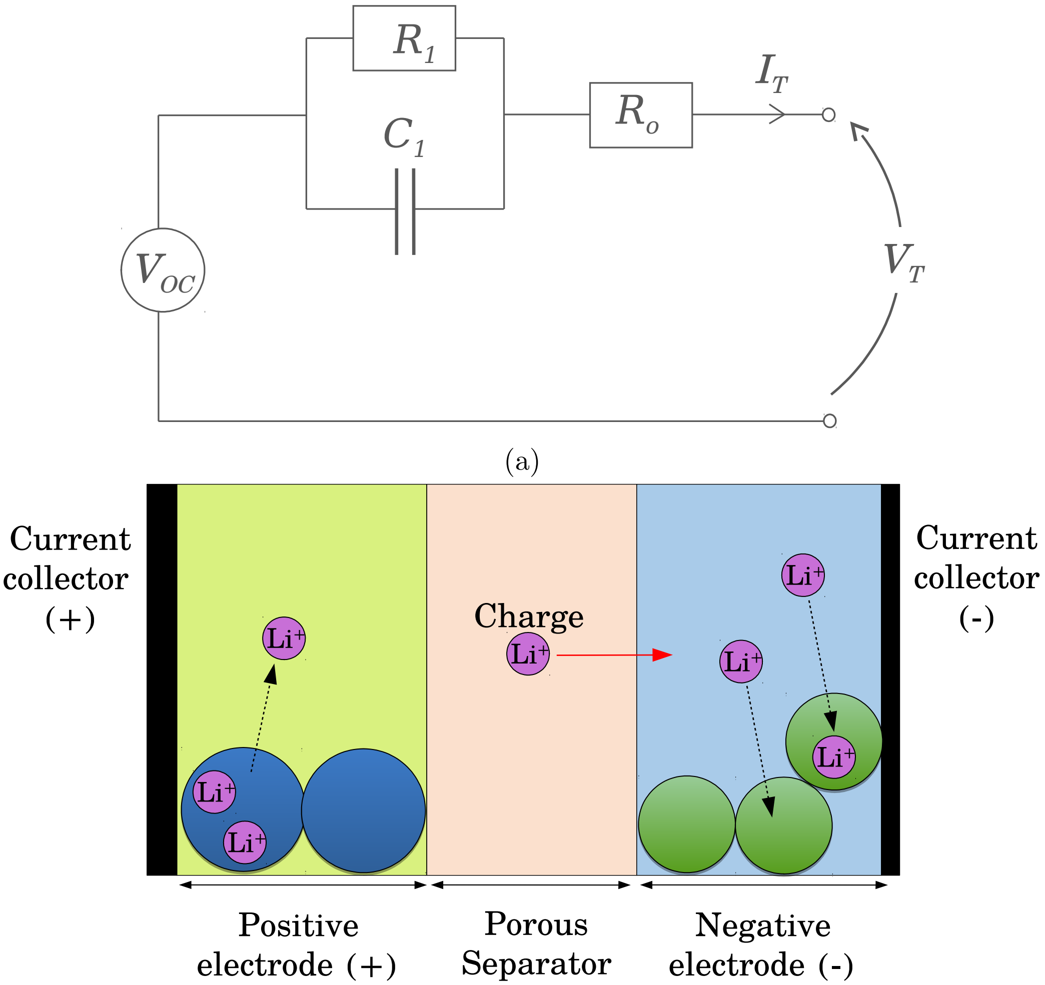
When the external circuit switch is closed, negative charge accumulates on the anode while positive charge is collected on the cathode. If a standard voltmeter with infinite resistance is used to measure the voltage across the two electrodes, the Open Circuit Voltage (OCV) would be measured. The OCV is a measure of the energy available to drive charge in the external circuit [97]. In the electrolyte phase, charge is transported form one electrode to another since the ions can freely move within it.

Accurate SOC estimation requires cell models, e.g. lumped parameter models and electrochemical models [98; 99]. Each model relies on experimental data to render accurate representations [2; 100; 101; 102].

Lumped Parameter Models

A simple type of model which has low computation overhead is the lumped parameter model [103; 30]. Linear and nonlinear parameters can be added to these models for higher accuracy or to model specific behavior [104; 105]. These typically do not contain parameters which are representative of any physical phenomena but can nevertheless compute the battery terminal voltage for certain charge/discharge rates. This type of model can be a combination of other simpler models like the Shepherd model, Unnewehr model and the Nerst model, which is given by

$$\begin{aligned} y_k &= K_0 - Ri_k - \frac{K_1}{z_k} - K_2 z_k + K_3 ln(z_k) + K_4 ln(1 - z_k) \\ z_{k+1} &= z_k - \left(\frac{\eta_i \delta t}{C}\right) i_k \end{aligned} \tag{2.3}$$



(a)

(b)

Figure 2.3: (a) Equivalent circuit battery model, where R_0 is the Equivalent Series Resistance (ESR) and the capacitor resistor pair C_1 and R_1 expresses some of the transient response. (b) Electro-chemical intercalation model of a Li-ion battery cell. These are often referred to as full order or reduced order particle models.

where y_k is the battery terminal voltage, R_i is the internal resistance, K_1 through K_4 are parameters used to fit the model, z_k is the SOC, η is the coulombic efficiency and C is the capacity of the cell [106]. Lumped parameter models have ease of implementation however they can fall short of sufficient representational accuracy when faced with variable ambient conditions and noisy real world measurements.

Equivalent Circuit Models

Equivalent circuit models or Thevenin models are loosely based on the underlying electrochemical process of batteries. Figure 2.3(a) shows an example of an equivalent circuit model. These typically are composed of a voltage source representing OCV which is connected to a series resistor and one or many RC pairs. The resistance represents the equivalent series resistance and the RC pairs express the various predominant time-constant inherent in the underlying diffusive processes. Although, some argue that these diffusive phenomena have a continuous spectrum of time-constants ranging from seconds to hours [107; 108] and that one or a few RC pairs are insufficient when trying to capture the full spectrum. Although, this might be only necessary for applications requiring high fidelity models. Solving the circuit equations of Figure 2.3(a), the terminal voltage of the battery cell is represented as follows;

$$V_T(t) - V_{OC}(t) = I_T(t)\left[R_o + R_1\left(1 - e^{-t/\tau}\right)\right] \tag{2.4}$$

and, similarly a second order equivalent circuit battery model is given by;

$$V_T(t) - V_{OC}(t) = I_T(t)\left[R_o + R_1\left(1 - e^{-t/\tau}\right) + R_2\left(1 - e^{-t/\tau_2}\right)\right] \tag{2.5}$$

These models are widely used for battery modeling as a result of their simplicity,

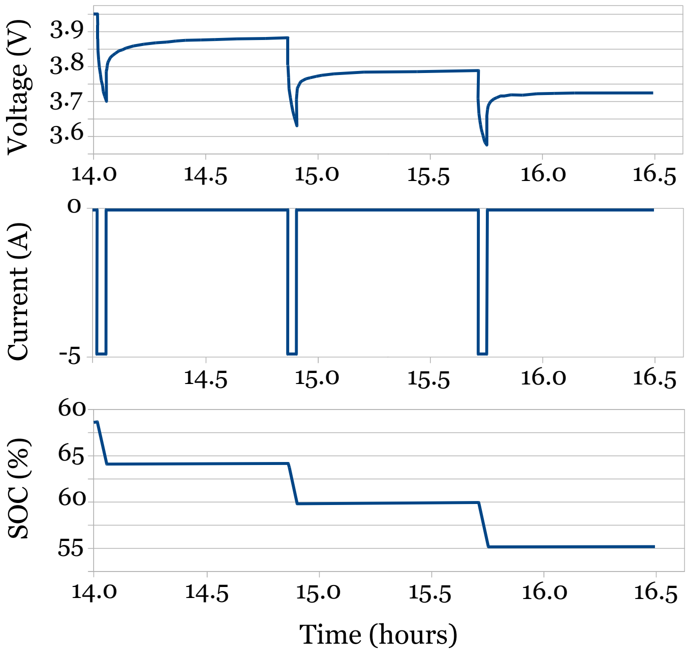


Figure 2.4: OCV as a function of SOC for LG Chem cell

ease of implementation and model transparency. To parameterize these circuit models, pulse tests, like the ones shown in Figure 2.4, are typically performed at every 5-10% SOC. A relaxation time of a few hours is needed between tests in order to allow the battery to reach equilibrium. Therefore, when performed for various ambient temperatures, this test can consume many hours or days. In addition, to allow the model to map the OCV to the state-of-charge, an OCV-SOC relationship, like the one shown in Figure 2.5, needs to be determined. To ensure that the relationship is properly determined, this latter test is required to the be performed at C-rates of C/20 or below. This requires an additional 20-50 hours depending on how low of a C-rate is used.

For tasks such as SOC and SOH estimation in varying ambient conditions, these models will require numerous look-up tables in order to continuously update their parameters. However, as mentioned above, for applications requiring further representational power, there are better options like the electrochemical models, discussed in the next section.

Electrochemical Models

Electrochemical models, which try to express the diffusive processes in batteries, offer higher fidelity than lumped parameter or equivalent circuit based models however they tend to be much more computationally expensive. The latter ensues as a result of the extensive amount of parameter fitting involved in electrochemical models as opposed to lumped parameter and equivalent circuit models.

Diffusion describes the movement of chemical species as a result of concentration differences. This usually continues to occur until the concentration gradient goes to zero. On an atomic level, diffusion defines the random walks and collisions undergone by ions with a kinetic energy k_BT, where k_B is Boltzmann's constant (1.3806×10^{-23} J K^{-1}). As a result, regions of high concentrations will intermingle with regions of lower concentrations and after enough time has elapsed a uniform distribution of concentration can be measured within the electrolyte. A current resulting from diffusion in the electrolyte is created by a concentration gradient. Fick's Law tells us that the constant of proportionality between the current and the concentration gradient is the diffusivity coefficient in the electrolyte, D_e [109; 110]. In a Li-ion cell, this is given by;

$$\frac{\partial c_e(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(D_e \frac{\partial c_e(x,t)}{\partial x}\right) + \frac{1}{F\epsilon_e}\frac{\partial t^0 i_e(x,t)}{\partial x} \tag{2.6}$$

where ϵ_e is the volume of the electrolyte and t^0 is the transference constant. In the solid phase, diffusion is the predominant phenomenon, thus, Li-ion mass transfer is described by the following function written in spherical coordinates;

$$\frac{\partial c_s(x,r,t)}{\partial t} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(D_s r^2 \frac{\partial c_s(x,r,t)}{\partial r}\right) \tag{2.7}$$

Although diffusivity is a dominant phenomenon in a Li-ion battery, there are other phenomenons which are often ignored in electrochemical battery modeling. These can include migration which occurs as a result of a potential gradient and convection which is a consequence of the general bulk electrolyte flow. In addition, as mentioned previously, electrochemical models require a slew of parameters and, often, many of these parameters like the diffusion coefficient are proprietary in nature which can be a major hindrance to anyone who does not have access to proprietary information.

2.3.3 State-of-Charge Estimation

Accurate and reliable battery models are essential for SOC and SOH estimation. A modern EMS estimates the SOC of each cell or each module of cells in the pack [17; 15]. The SOC is mapped onto the open circuit voltage (OCV) [111] and, in general, is a nonlinear function, varying with age and temperature [30; 18; 98]. OCV based SOC estimation measures the terminal voltage after a cell has rested for minutes or hours (dependent on chemistry) to attain electrochemical equilibrium and reconstructs SOC with lookup tables. An example of OCV-SOC profiles taken from data found in the NASA prognostics repository [112], is shown in Figure 2.5 for an LG Chem cell discharged at C/50. The main limitations of this approach are that it cannot be used during operation; the OCV-SOC characteristic is subject to changes when a cell ages; and some battery chemistries (e.g. Li-ion) have flat voltage profiles for intermediate charge levels that yields large uncertainties in SOC estimation. The conceptually opposing approach is coulomb counting [30; 17; 113; 114; 115; 20; 116]. This method integrates the current flowing into or out of each battery cell. For chemistries with low self-discharge rates such as Li-ion, this method can accurately determine small changes in the stored amount of charge. The main limitation of this approach is that the capacity and charge, i.e. discharge efficiency needs to be known with sufficient precision. In addition, the method is susceptible to drifts of the estimates over time due to small measurement or coefficient offsets. Hence the SOC needs to be corrected which is typically done through the OCV information [30; 17; 115; 20; 117; 118; 119]. More sophisticated methods have been developed to overcome the shortcomings of the above mentioned techniques. These methods are based on (adaptive) cell models and use voltage and current measurements to correct the estimate. They can be classified

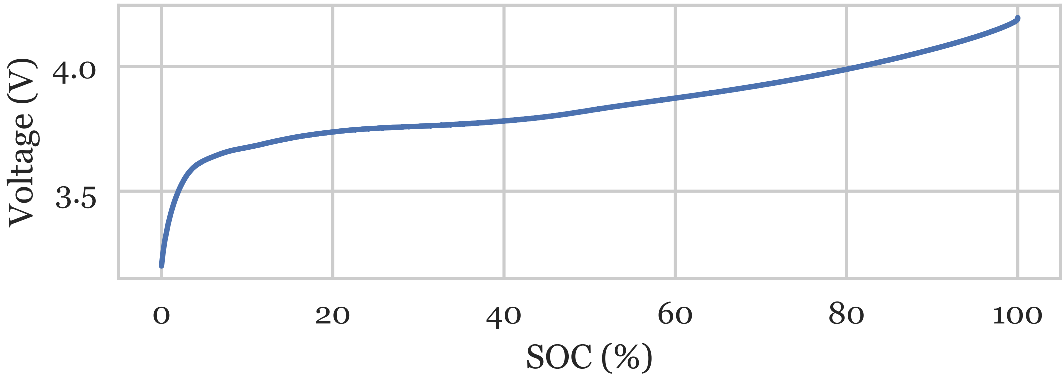


Figure 2.5: OCV as a function of SOC for LG Chem cell

with respect to their estimation principle: Luenberger observer [17; 21; 85; 120; 121], adaptive observer [17; 22; 122], sliding mode observer [17; 23; 123; 124; 125; 119; 126; 24], and Kalman Filters [2; 101; 102; 127]. Typically, in observer methods, the parameters of an equivalent circuit model like resistances and open circuit voltage are fit to observed battery current and voltage data. An estimate is issued by mapping these parameters to SOC. In Kalman filter-based algorithms, it is typically required to linearize around an operating point which can significantly increase computational load. The measured current, voltage and the previously estimated SOC are provided to the algorithm and the filter issues an estimate of SOC at the next time step. These techniques are often tied to some battery model, like a lumped parameter model or an equivalent circuit model which require arduous model identification to adequately represent the non-linear behavior of a battery. In addition, they often require large numbers of parameters or different versions of the model to perform SOC estimation at varying ambient conditions.

Strategies involving classic machine learning algorithms have also been used in the past. The benefit of these types of techniques is that they can be trained with real world data and self-learn SOC estimation without the need for hand-engineered models. However, where neural networks have been solely used, the results are typically not accurate enough, and therefore require the additional use of Kalman filters or other inference mechanisms to achieve sufficient estimation accuracy. Although some works have used Kalman filters in conjunction with combined battery models or equivalent circuit battery models [128], many other works have also used them in conjunction with NN battery models.

The work performed in [129] uses a Radial Basis Function Neural Network (RBF NN), having as inputs the battery terminal voltage at the previous time step, current and SOC at the current time step. The trained 2-layer RBF NN having 30 neurons in the hidden layer outputs the terminal voltage. The activation function used in this NN are Green functions given by;

$$\phi_i(\mathbf{r}_k) = exp\left(-\frac{\|\mathbf{r}_k - \mathbf{t}_i\|^2}{\sigma_i^2}\right) \tag{2.8}$$

where $\mathbf{r}_k = [V(k-1), SOC(k), i(k)]^T$ is the vector of inputs, k is the time step, $\mathbf{t}_i$ and σ_i are the center and standard deviation of the Gaussian function. The output from the network is a regressive function of the following form;

$$F(\mathbf{r}_k) = \sum_{i=1}^{M} w_i \phi_i(\mathbf{r}_k) + b_0 \tag{2.9}$$

where $i = 1, ..., M$, M is the number of neurons in the hidden layer, w_i are the weights connected the hidden layer activations to the output layer, and w_0 are the biases. All inputs and outputs from the RBF NN are normalized. The output of the RBF NN is used as a state variable at the next time step of the state-space model. An Extended Kalman Filter (EKF) approximates the non-linear model and estimates its states. Appropriate covariance matrices for the measurement noise and the process noise are selected from the battery data. After training, the RBF NN infers the outputted terminal voltage within a 4% RMS error. Using the RBF NN in combination with the EKF, the RMS error is further reduced down to 2%.

In [130], an Extreme Learning Machine is used at a constant ambient temperature of 25°C. An SOC estimation error of under 1.5% is claimed however this is only achieved in conjunction with a Kalman filter. Furthermore, the extreme learning machine is trained on constant discharge pulses hence their performance in transient load demand, experienced in real world scenarios, is unknown.

In [131], a Least Square Support Vector Machine (LSSVM) uses RBF kernels to estimate battery voltage. The network structure is very similar to the network structure discussed in [129], where the same RBF kernel and output function described in Equation (2.8) and Equation (2.9), are used, respectively. The input to the network is battery current. A moving window method is used such that only the measurement data found within the window is fed into the LSSVM during training. This limits the training dataset to a manageable size. The output of the LSSVM is used in the state-space model constructed for an Adaptive Unscented Kalman Filter (AUKF). AUKF is used since this is not dependent on the Jacobian matrix and can adaptively adjust the covariance matrices. The method is first tested on a Li-ion battery model constructed in Simulink and achieves an error of less than 0.5%. The method is then applied over recorded experimental battery data. The experimental validation data included both charging and dynamic profiles to test the method on slow and fast dynamics. The LSSVM with AUKF achieves a mean absolute error of below 2%.

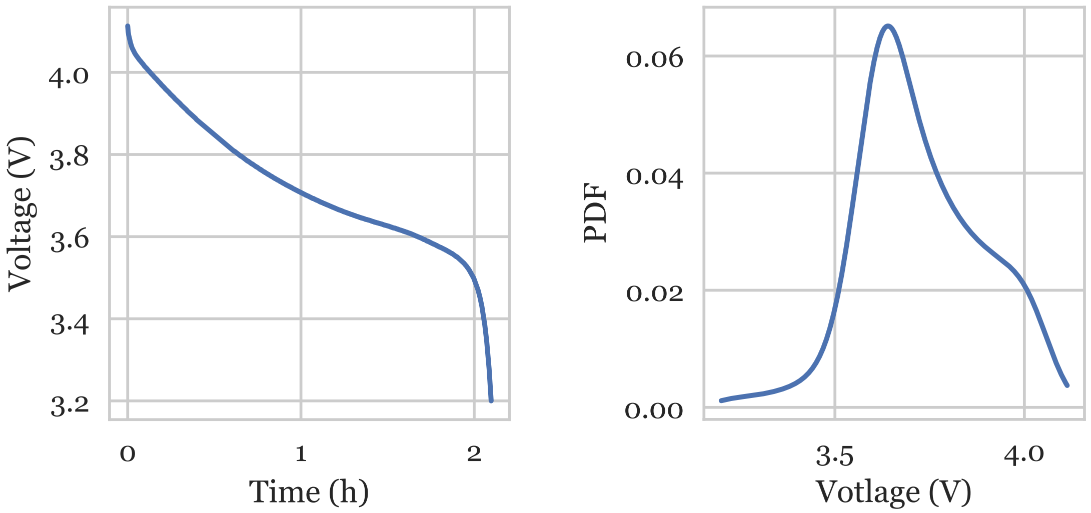
In [132], a 2 layer feedforward neural network is used as a battery model. A variant of gradient descent called the Levenberg-Marquardt method is used to train the algorithm. The input data is divided into charging, idling and discharging segments. When the battery data is classified as being in one of these three categories, it is passed to a set of neurons that are different than the neurons used for the other categories of input data. The output layer aggregates the activations from all three sets of neurons in the hidden layer and passes a single output value. The output still contained some noise so the authors use a filter composed of a moving average calculation followed by a saturation function. The reference SOC used to generate the estimation error was computed using current integration. Various inputs were used in this approach; these included voltage, current, first and second derivatives of the voltage and the current. The load classifying neural network is trained on 12 US06 drive cycles and achieves an average estimation error of 3.8%, or 2.6%, when additional filtering is performed.

The work performed in [133] uses a dual neural network fusion model. The first of these neural networks is a linear one which takes the voltage and current at the previous time step and current at the present time step and infers the voltage at the current time step. This NN is a structured one where the network weights correspond to the parameters of an equivalent circuit model. These network weights are then used to compute the OCV. The OCV is then fed into a second neural network containing one hidden layer and outputs the estimated SOC. Using the parameters of a first order equivalent circuit model (R-RC) the fusion neural network model achieves a mean absolute error and a maximum error of 1.03% and 4.82%, respectively. When a second order equivalent circuit model (R-RC-RC) is used, the mean absolute error and the maximum error become 0.75% and 3.4%, respectively. Although estimation accuracy is good, the method is untested at ambient temperatures other than room temperature, therefore the expressibility of the method in varying ambient conditions is unknown.

2.3.4 State-of-Health Estimation

SOH estimation can be done through parameter identification of equivalent circuit models as is performed in internal resistance measurement methods [134; 135]. In [134], a specific identification signal which occurs regularly is used to parameterize an R-RC-RC equivalent circuit model. Parameter identification is performed with the linear least-squares method. The identification signal is obtained when the ICE is turned on where current and voltage are recorded. This signal occurs regularly and has a predefined time-span which does not change over time. In addition, the work performs this resistance-based SOH estimation for numerous ambient temperatures ranging between -20°C to 50°C. A degradation index is obtained from the internal resistance value however this aging index was put through a low-pass filter to reduce noise. Although this method strategically uses a regularly occurring reference signal to obtain a smooth degradation index, these signals can only be obtained from hybrid electric vehicles and are nonexistent in the case of electric vehicles.

Recently, there are increasingly more methods utilizing probabilistic approaches and incremental capacity analysis over reference charge or discharge profiles to perform SOH estimation. These approaches are appealing given that they do not use

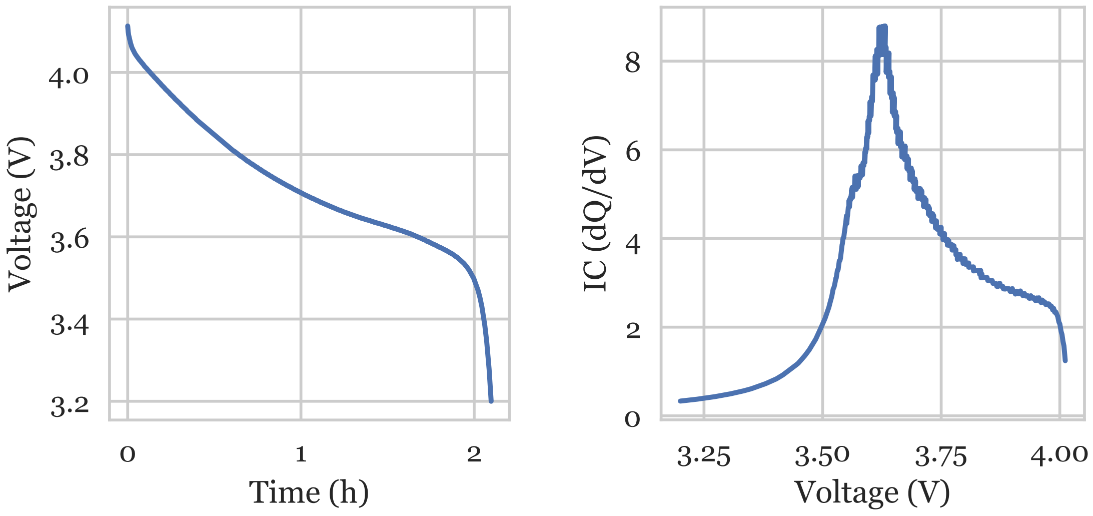


(a) (b)

Figure 2.6: (a) Terminal voltage as function of time of LG Chem cell discharging at 1C. (b) Probability Density Function (PDF) as a function of the terminal voltage used for SOH estimation.

battery models and they do not rely on historic degradation data or ambient conditions. Since the voltage curves, taken at constant C-rates, of most Li-ion battery chemistries contain a plateau region, these approaches will typically transform the voltage data into another representation which is more sensitive to these plateaus.

In [136], the Probability Density Function (PDF) of the terminal voltage of a constant current discharge profile is used to estimate the battery SOH. An example of a PDF curve is shown in Figure 2.6 which is plotted using data from an LG Chem cell [112]. This is performed by categorizing the voltage data into bins which span specific voltage increments and normalizing this discrete dataset to get the PDF. In the study, the PDF is then smoothed so that its peak is well represented. The strategy relies on the peak values of the PDF profile to map the PDF to an estimated SOH value. Therefore, one drawback of this strategy is that a specific range of voltage



(a) (b)

Figure 2.7: (a) Terminal voltage as function of time of LG Chem cell discharging at 1C. (b) Incremental Capacity (IC) as a function of the terminal voltage.

values is required in order to have a well defined PDF peak. The study achieves an error of below 1.6% on the reference charge curves however its performance over partial reference profiles, often encountered in real world scenarios, is unknown.

Since discharge profiles observed between charging events occur in an uncontrolled fashion, some works have relied on charge profiles to estimate SOH. In [137], Incremental Capacity Analysis (ICA) profiles are identified from 0.5 C charge curves. ICA curves, which describe the derivate of the capacity with respect to the voltage, dQ/dV, are better at detecting small incremental changes in the charge curves. For example Figure 2.7, is an incremental capacity curve for an LG Chem cell at the beginning of its life. To identify this ICA profile from noisy measurements and from partial reference profiles, a support vector machine showed best results. This method achieves a

1% Mean Absolute Error (MAE) however, battery packs in electrified vehicles for example, can be charged at higher charging rates than 0.5 C; sometimes reaching more than 1 C rates. Therefore, its performance at these higher charge rates is unknown. Because of noisy measurements and partial charge curves encountered in real world environments, this method required the use of a support vector machine in order to maintain consistent and reliable identification of ICA profiles throughout the battery's lifetime. In [138], the constant current component of the charging voltage curve is used with an equivalent circuit model to extract the aging parameters. Parameter identification is achieved using a nonlinear lest squares technique. This SOH estimation method does not require the calculation of OCV and SOC which removes some computational burden. This work achieves 2-3% errors although its performance on partial charge curves and on charge rates higher than 0.5 C is unknown.

Works using observers [139] and Kalman filters [140] have also showed great promise. However, although the latter methods show good estimation performance, they often have a drawback which hinders their practical application; these can include the use of distinct loading profiles at regular intervals or characteristic loading profiles performed at low C-rates to identify signatures associated with SOH. Recently, the use of machine learning models and other data-driven approaches have shown good performance while attempting to solve some of these drawbacks. As a result, these types of algorithms have been garnering greater interest [141; 142; 143].

In [144], the authors capitalize on the correlation between the diffusion capacitance of a second order equivalent circuit model and state-of-health. A Genetic Algorithm (GA) is used to estimate the voltage drop across this diffusion capacitance as well as to estimate the open circuit voltage. The latter are then used to parameterize the battery model. The maximum error observed while using this estimation strategy is 4.35% however, the parameter identification is performed online with the GA which can be computationally intensive.

In [145], a structured neural network is compared to an extended Kalman filter to perform parameter estimation. The architecture of the structured neural network tries to mimic the acausal structure of an equivalent circuit model while a feedforward neural network has inputs fully-connected to all the weights in the first hidden layer. Therefore, after training, the weights in the structured neural network will correspond to parameters of the an equivalent circuit model. Training uses gradient descent to converge to a solution. The inputs used are measured battery current, temperature as well as SOC and the output from the network is the battery voltage. The network parameters can be combined to form the internal resistance of the battery cell which can be used to estimate SOH during the batteries lifetime. The neural network is composed of few network weights given its structured architecture and therefore can have low computation time. The second more popular approach used in this study is the Extended Kalman Filter. This method uses the voltages across the RC pairs in an equivalent circuit model as well as the internal resistance as the state variables. The system is linearized using the Jocobian Matrix and a covariance matrix is identified. The two approaches are compared over drive cycle data taken from hybrid electric vehicle. The authors state a mean error between measured and estimated voltage of less than 12mV for the structured neural network although no error metrics are offered for the internal resistance estimation. They also claim that the performance between the structured neural network and the EKF method is almost identical.

Chapter 3

Machine Learning and Modern Deep Learning Algorithms for State Estimation of Li-ion Batteries

3.1 Learning Algorithms

Since ancient times mankind has been enamored with endowing inanimate objects with intelligence. The earliest signs of automatons were sacred statues in ancient Egypt and Greece where it was believed that the craftsmen of those statues endowed them with real minds. The birth of Artificial Intelligence (AI) can be traced back to the 1950s, when multiple promising theories and works intersected. Norbert Wiener defined cybernetics in 1948, which considered the scientific study of control and communication of any system using some form of technology. Information theory was proposed by Claude Shanon in his famous work titled ”A Mathematical Theory of Communication”. In addition, Alan Turing's popular theory of computation described that any form of computation can be represented digitally. In 1956, the Dartmouth Conference, organized by Marvin Minsky, John McCarthy and others, is by most accounts, where the birth of AI took place. Many attendees of this conference would later work on some of the foundational aspects of AI. Inspired by the human brain, Frank Rosenblatt conceived the perceptron which was akin to a neural network. Great optimism ensued this period and with it came increased amounts of funding from organizations like DARPA. However, scientists soon realized that hardware limitations would not allow them to reach some of the milestones which they had set for their research projects and, as a result, AI funding dried out. This would later be known as the 'AI winter'. Unfortunately, this was one of many 'AI winters' as, over the years, interest in the field waxed and wanned.

In the 1990s and early 2000s, some of the oldest goals set for AI at its infancy had started to be achieved. In 1997, Deep Blue became the first computer to beat the world chess champion, Gary Kasparov. With the advent of big data and continuously advancing hardware technology, great strides have been made in the field of AI and interest in this field is rising once again. Machine Learning (ML) is the field of study which considers machines learning from data. Popular algorithms like logistic regression and genetic algorithms are included in the field of ML. Deep learning is one approach to AI and a sub-field of ML, which describes the process of allowing machines to learn from experience and perceiving the real world in a hierarchal way. Figure 3.1 is a Venn diagram showing the organizational structure of deep learning and machine learning within the field of artificial intelligence. Deep learning algorithms learn simpler subsystems and build on this knowledge to learn and express more complex systems. Therefore, this compositional structure of learning features from other features resulting in a hierarchy of learned layers is the reason why the field is entitled 'deep learning'.

Many classical methods of AI also learned from feature sets. However, traditionally, many of these techniques rely on hand-crafted features to facilitate learning. For many tasks, especially more complex learning tasks, it is tedious to hand-engineer all possible features so that the computer can have a good generalized representation of the system. Therefore, representation learning allows the computer to learn its own representation, removing model rigidity imposed by human intervention. This feature of machine learning and deep learning not only allows the computer to have a generalized representation of the system but also allows it to adapt to varying changes in its environment, which would otherwise be very difficult to accomplish with hand-crafted representations [146].

For a particular task, the data or set of examples from which a computer learns, is as important as the learning algorithms themselves. Sufficient data will allow

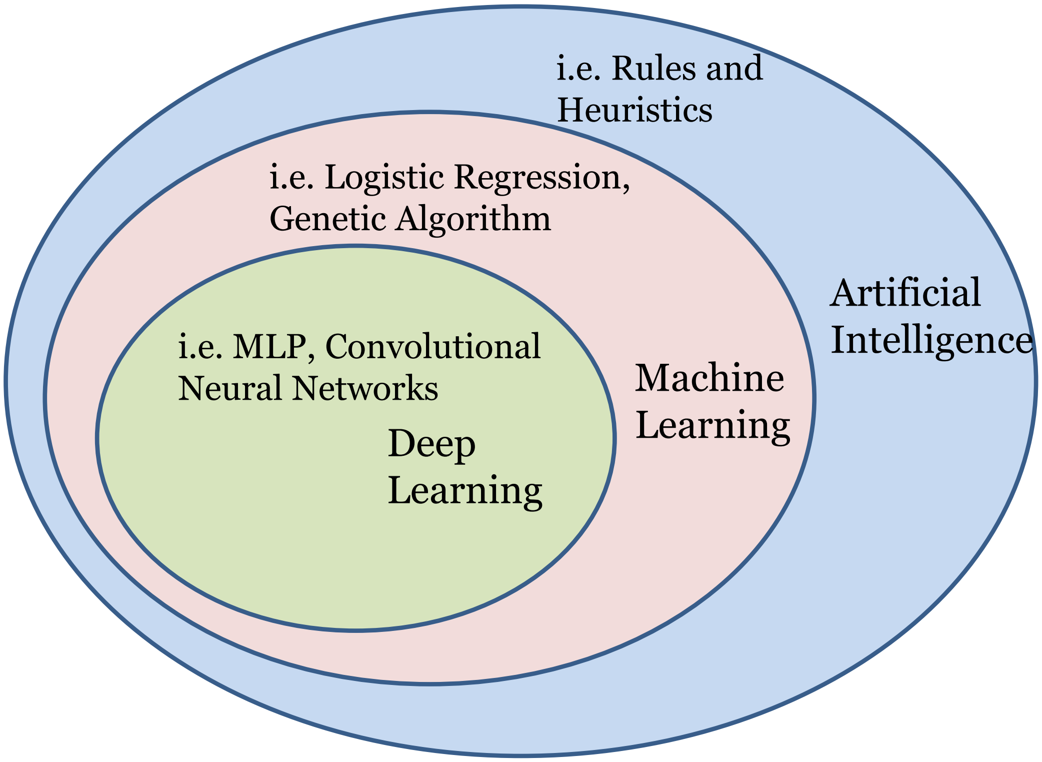


Figure 3.1: Venn diagram of the field of AI, machine learning as a subfield of AI and subsequently, deep learning being a subfield of ML. Deep learning and some techniques in ML benefit from representation learning which allows the computer to learn from experience in contrast to some classical forms of machine learning like logistic regression which rely on hand-crafted features.

the learning algorithm to ensure a general representation although this is difficult to assess aside from looking at its performance over validation data. Recently, an abundance of readily available real world data has allowed for better training and better performance from these algorithms. In addition, semiconductors and information technology have been increasing in performance at a pace which some would even argue is moving faster than Moore's Law. Therefore, recently, training learning algorithms can be accomplished at a faster pace. In addition, Graphical Processing Units (GPU), which operate at lower clock speeds than CPUs but have many thousands of processing cores, have allowed for faster training times when parallelizing the learning system's operations.

Learning algorithms can perform many different types of tasks. Two of the more popular tasks are classification and regression. Classification is a task which requires the computer to classify an input feature vector, $\mathbf{x}$, to a class y, where y is an integer. Regression is a task whereby the computer is asked to predict or to infer some numerical value from some input; this is given by $y = f(\mathbf{x})$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The deep learning algorithms which are used in this research work, perform regression rather than classification. Also, hyperparameters typically represent aspects of the network which are not directly trainable and which determine the structure of the network. These include the learning rate of the models, the number of layers and the number of neurons in each layer [146].

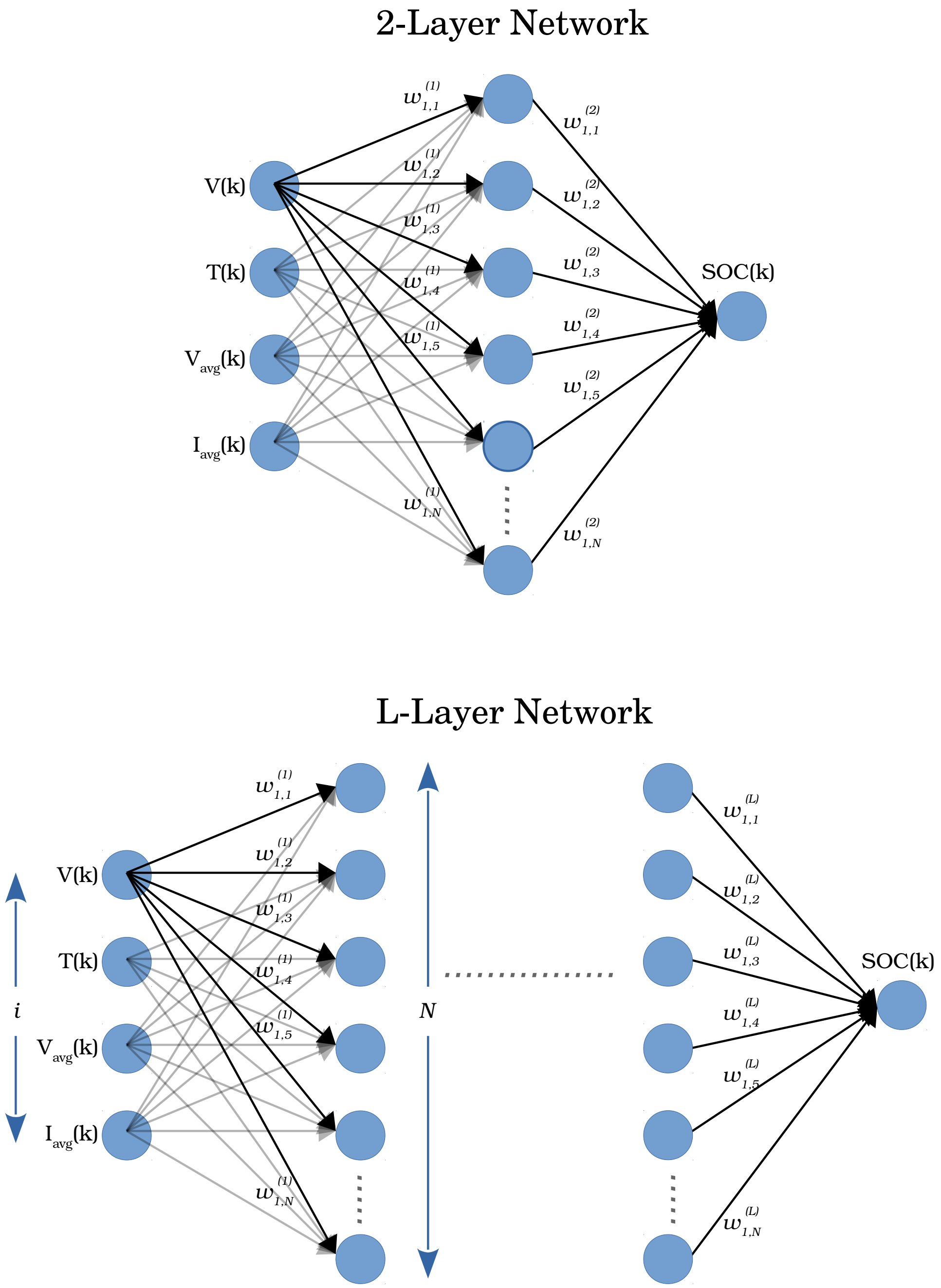
3.2 Modern Deep Learning Networks

There are many examples where deep learning architectures have made significant improvements over conventional algorithms. In 2012, AlexNet, a deep convolutional neural network won the ImageNet competition where teams are tasked with classifying over 1 million high resolution images in 1000 different categories. AlexNet achieved a top-5 error rate of 15.3% compared to a more traditional model taking second place with a top-5 error of 26.2% [147]. Recently, Microsoft Research's deep learning algorithm, called a deep residual network, won the 2015 ImageNet challenge with an error rate of 3.57% which even surpasses human level accuracy valued at 5.1% [148].

3.2.1 Deep Neural Networks for SOC Estimation

Traditional machine learning techniques contain no more than one or two layers of non-linear and linear transformations [149]. With the advent of faster computational power and an abundance of available real world data, deeper architectures were investigated which, in many cases, allowed researchers to make striking improvements in many applications [147; 150; 151; 152; 153].

Feedforward neural networks, whose 2-layer and multi-layer DNN architectures are shown in Figure 3.2, can, in principle, model any non-linear system by mapping the observables to a desired output. Once trained offline, FNN and DNN offer fast computational speeds online since they are composed of a series of matrix multiplications, as opposed to other strategies which can contain computationally intensive calculations like partial differentials equations. When FNN and DNN are applied for SOC estimation, a typical dataset that is used to train the networks is defined by $D = \{(\psi(1), SOC(1)^*), (\psi(2), SOC(2)^*), ..., (\psi(\tau), SOC(\tau)^*)\}$, where $SOC(t)^*$ and $\psi(t)$ are the ground-truth state-of-charge value and the vector of inputs at time step t, respectively. The current measurement used to determine the ideal or ground-truth $SOC(t)^*$ is described in more detail in Chapter 6. The vector of inputs is defined as



2-Layer Network

L-Layer Network

Figure 3.2: Architecture of 2-layer neural network (top) and architecture of Deep Neural Network (DNN) (bottom). The input data is given by $\psi(t) = [V(t), T(t), I_{avg}(t), V_{avg}(t)]$ where $V(t)$, $T(t)$, $I_{avg}(t)$ and $V_{avg}(t)$ represent the voltage, temperature, average current and average voltage of the battery at time step t. The output of the DNN is the estimated SOC at every time step.

$\psi(t) = [V(t), T(t), I_{avg}(t), V_{avg}(t)]$ where $V(t)$, $T(t)$, $I_{avg}(t)$ and $V_{avg}(t)$ represent the voltage, temperature, average current and average voltage of the battery at time step t. The average current and voltage are both calculated over λ precedent time steps, which ranged from 50 to 400 time steps. This is not to be confused with the total dataset time span defined by τ, where $\lambda < \tau$. Many different types of inputs were examined and it was found that I_{avg}, V_{avg} are better values to feed into the network rather than feeding many antecedent values of current and voltage. The number of weights in the network increases proportionally to the number of additional inputs times the number of neurons in the first hidden layer. For example, in a network which has 10 neurons in the first hidden layer, each additional input will require 10 additional weights for a fully-connected layer. Therefore, better computational efficiency and lower memory requirements are achieved by capturing the system dynamics in time through an averaging of current and voltage.

To construct these networks, TensorFlow [154], a machine learning framework, is used in conjunction with two NVIDIA Graphical Processing Units (GPU); a TITAN X and a GeFORCE GTX 1080 TI GPU. The TensorFlow framework provides the ability to quickly prototype and test different network architectures as a result of its ability to automatically compute backpropagation. The latter describes the process by which the network weights can be updated at the end of every training epoch and will be further discussed below. Although training is performed on a GPU to take advantage of parallel computing and to accelerate the training process, when performing real time estimation after training is complete, the algorithms discussed in this section can be flashed on a standard microprocessor.

Feedforward neural networks are matrix-based and can be represented by the below composite function. A few important variables are first defined. Let $w^l_{j,k}$ denote the weight connection between neuron j in layer $l-1$ and neuron k in layer l. Let b^l_k and h^l_k be the bias and the activation function, respectively, of neuron k in layer l . The activation function is computed as follows;

$$h^l_k(t) = \eta\left(\sum_k (w^l_{j,k} h^{l-1}_k(t) + b^l_k)\right) \tag{3.1}$$

where,

$$h^l_k(t) = SOC(t) \quad \text{for } l = L \tag{3.2}$$

and where $\psi(t)$ is provided to the network at the input layer or when $l = 0$. $\psi_i(t)$ is an element of the vector of inputs at every time step t and is given by $\psi(t) = [V(k), T(t), I_{avg}(t), V_{avg}(t)]$. As mentioned above, $V(t)$, $T(t)$, $I_{avg}(t)$ and $V_{avg}(t)$ are the voltage, temperature, average current and average voltage of the battery at time step t. $SOC(t)$ is the estimated state-of-charge at time step t calculated by the network. Due to its simplicity during training as well as during validation, the nonlinearity used in these networks is called Rectified Linear Units (ReLU) and is given by;

$$\eta = max(0, h) \tag{3.3}$$

To understand how accurate the SOC estimate is compared to the ground-truth value, an error signal is generated at the output of the network for each time step and is given by

$$e(t) = SOC(t) - SOC^*(t) \tag{3.4}$$

To gain a good understanding of the overall error energy of the entire network computed at the end of a forward pass, the following loss function is used;

$$\mathcal{L} = [max(e)]^2 + \frac{1}{\tau}\sum_{t=0}^{\tau}\left(e(t)^2\right) \tag{3.5}$$

where τ is the length of the sequence, as mentioned above. Both the average error and the maximum error are included in the overall error function so that the network expends its energy on minimizing both the average error as well as the maximum error value and does not prioritize the minimization of one value over another.

3.2.2 Recurrent Neural Networks

Since 2012, great advancements have been achieved in the field of speech recognition [151; 155; 156]. More recently, a team at Baidu Research has developed a deep learning speech recognition system called Deep Speech 2 which exceeds human-level accuracy [157]. The latter deep neural network was composed of many layers which included layers of RNNs. Recurrent neural networks, shown in Figure 3.3, are a class of ANN geared towards pattern recognition in sequential datasets. Some popular examples of this are speech recognition, natural language understanding and machine translation where characters or words are fed into the network sequentially. Aside from the latter applications, RNNs can be useful for just about any type of time series data [158]. However, classical RNNs are known to have issues with long-range dependencies where the gradient either explodes or vanishes during back-propagation [159; 160]. RNNs with Long Short-Term Memory (LSTM) cells [161] have had better success at capturing long-term dependencies within a sequence and are thus much more widely used for these types of applications.

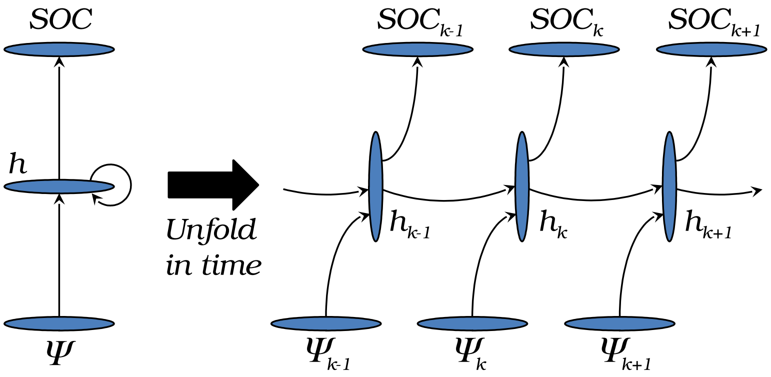


Figure 3.3: Architecture of Recurrent Neural Network (left) and architecture of RNN unfolded in time (right). The input data is given by $\varPsi_k = [V(k), I(k), T(k)]$ where $V(k), I(k)$ and $T(k)$ represent the voltage, current and temperature of the battery at time step k. h_{k-1} denotes the hidden layer of the LSTM-RNN at time step $k-1$. The output of the LSTM-RNN is the estimated SOC at every time step.

3.2.3 Recurrent Neural Networks with Long Short-Term Memory Cells

A LSTM-RNN, whose architecture is also shown in Figure 3.3, can represent non-linear dynamic systems by mapping input sequences to output sequences. When we applied LSTM-RNNs towards SOC estimation, a typical dataset used to train the networks is given by $D = \{(\varPsi_1, SOC_1^*), (\varPsi_2, SOC_2^*), ..., (\varPsi_N, SOC_N^*)\}$, where SOC_k^* is the ground-truth value or the observable state-of-charge value at time step k and $\varPsi_k$ is the vector of inputs also at time step k. The vector of inputs is defined as $\varPsi_k = [V(k), I(k), T(k)]$, where $V(k)$, $I(k)$, $T(k)$ are the voltage, current and temperature of the battery measured at time step k, respectively. The Long Short-Term Memory cell whose schematic representation is shown in Figure 3.4, is equipped with

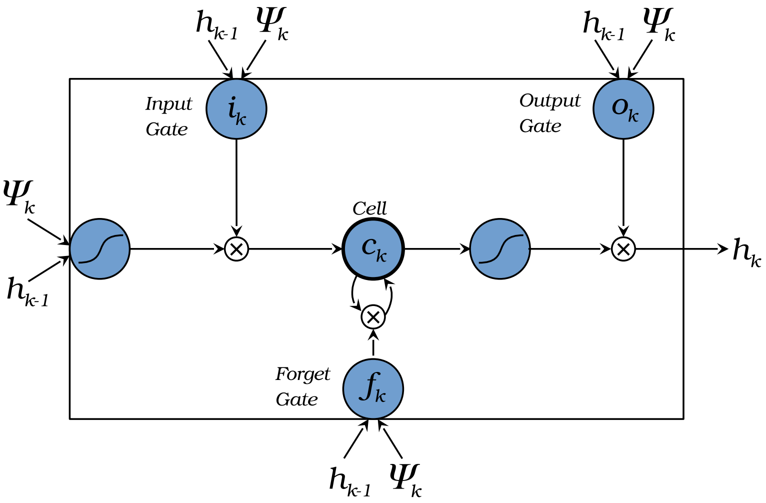


Figure 3.4: Long Short-Term Memory cell. Ψ_k and h_{k-1} are the input data layer at current time step, k, and hidden layer at previous time step, $k-1$, respectively. i_k, o_k, f_k and c_k are the input, output and forget gates as well as the memory cell, respectively. Squiggly lines represent nonlinearities outlined in equation 3.6.

a memory cell, c_k, which stores the long-term dependencies. During training, the input, output and forget gates allow the LSTM to forget or write new information to the memory cell. To construct the LSTM-RNNs, TensorFlow [154], is also used in conjunction with two Graphical Processing Units (GPU); the NVIDIA Titan X Pascal and a GeFORCE GTX 1080 TI. The constructed networks incorporated TensorFlow's basic LSTM cell which is based on the LSTM unit discussed in [162] and can be represented by the following composite function,

$$
\begin{aligned}
i_k &= \eta(W_{\Psi i}\Psi_k + W_{hi}h_{k-1} + b_i) \\
f_k &= \eta(W_{\Psi f}\Psi_k + W_{hf}h_{k-1} + b_f) \\
c_k &= f_k c_{k-1} + i_k tanh(W_{\Psi c}\Psi_k + W_{hc}h_{k-1} + b_c) \\
o_k &= \eta(W_{\Psi o}\Psi_k + W_{ho}h_{k-1} + b_o) \\
h_k &= o_k tanh(c_k),
\end{aligned}
\tag{3.6}
$$

where the initial hidden state, h_0, is set to a zero matrix, η is the sigmoid function and i, f, o and c are the input, forget, output gates and memory cell, respectively. They are called gates since they are a sigmoid function which can be zero valued thus possessing the ability to inhibit the flow of information to the next computational node. Each gate possesses its set of network weights which are denoted by W. The subscripts of W describe the transformation occurring between the two respective components, e.g. the input-output gate matrix is denoted by $W_{\Psi o}$, the hidden-input gate matrix is denoted by W_{hi}, etc. A bias, b, is added to the matrix multiplication at each gate to increase the networks flexibility to fit the data. A final fully-connected layer performs a linear transformation on the hidden state tensor h_k to obtain a single estimated SOC value at time step k. This is done as follows:

$$SOC_k = V_{out}h_k + b_y, \tag{3.7}$$

where V_{out} and b_y are the fully-connected layer's weight matrix and biases, respectively. The disparity between the LSTM-RNN's estimated SOC and the measured one is best represented by the following loss function computed at the end of each forward pass;

$$\mathcal{L} = \sum_{k=0}^{N} \frac{1}{2}(SOC_k - SOC_k^*)^2, \tag{3.8}$$

where N is the length of the sequence and SOC_k as well as SOC_k^* are the estimated and ground truth values of SOC at time step k, respectively.

Many unique drive cycles are concatenated to form the training dataset and when compiled, this typically has a length of over 100,000 time steps. It is not possible to enter a sequence as long as this into our GPU memory during training. Therefore, the LSTM-RNN models are trained by feeding one batch of the sequence at a time which is commonly performed while training LSTM-RNNs. This is referred to as unrolling the LSTM cell in time for $\widetilde{N}$ steps where $\widetilde{N}$ is the batch length holding a smaller value than the total training sequence length, N, such that $\widetilde{N} < N$. Usually, if the time constant of the inherent dynamics within the sequence is shorter than $\widetilde{N}$, then the LSTM-RNN can still capture the time dependencies.

3.2.4 Convolutional Neural Networks for SOH Estimation

Lately, the surge of renewed interest in Artificial Intelligence (AI) has been primarily underpinned by the numerous advancements achieved with deep learning algorithms [147; 150; 151; 163; 152; 153]. One prominent example of these algorithms is the Convolutional Neural Network. This technique showed its prominence in 2012, with AlexNet, a deep learning algorithm composed of many layers of convolutional neural networks. It was able to surpass the accuracy of competing classical pattern recognition algorithms when tasked with classifying 1 million images over 1000 different categories in the ImageNet competition.

Deep learning techniques are disrupting many other fields. In the field of chemical information and modeling, deep learning methods are used as predictive models for prioritizing experiments during the drug discovery process [152]. In bioinformatics and genomics, deep learning has been used to accurately predict alternative splicing patterns in tissue samples as well as in identifying new genetic mutations linked to certain diseases [164; 165]. A more recent pathological study uses CNNs to identify and locate metastatic breast cancer [153]. These reduced a human pathologist's error rate by 85%. In particle physics, deep learning techniques have recently been implemented in a 2014 study for the classification of particles and for the discovery of exotic particles like the Higgs boson using noisy but rich data sets from high-energy particle colliders. An 8% improvement was recorded over incumbent state of the art machine learning techniques [166].

Feedforward neural networks can in principle model any non-linear system by mapping a set of measured quantities to a desired output. Convolutional neural networks, shown in Figure 3.5, are particularly good at performing this mapping function when considering higher dimension datasets having inherent dependencies between adjacent dimensions. The CNNs used in this work, convolve kernels over a two dimensional dataset at each layer instead of fully interconnecting adjacent layers as is typically done in fully connected neural networks. Convolving kernels over the two dimensional dataset, allows these networks to benefit from shared weights and an invariance to local distortions. Thus, nowadays, they are the preeminent algorithm used in image recognition. However, they can be very effective for many other two dimensional datasets which have correlations along one or both dimensions. Once CNNs are trained offline, they can offer fast computational speeds onboard a mobile device or vehicle, since they are formulated by a series of convolution and matrix multiplication operations which are great candidates for parallel computing. The typical dataset used for training and validation in this work is given by $\mathcal{D} = \{(\mathbf{\Psi}(1), SOH^*(1)), (\mathbf{\Psi}(2), SOH^*(2)), ..., (\mathbf{\Psi}(\mathit{\Xi}), SOH^*(\mathit{\Xi}))\}$, where $\mathbf{\Psi}(\xi)$ and $SOH^*(\xi)$ are the the array of input data and the ground-truth state-of-health value, respectively, for each charge profile, ξ. The input charge profiles, $\mathbf{\Psi}(\xi) \in \mathbb{R}^{N\times M}$, which will be discussed further in the next section, can be composed of battery measurements like current, voltage and temperature or, in the case of partial charge profiles, they could also include the SOC values.

The kernels, $\mathbf{w}^{lkm} \in \mathbb{R}^{C_l \times R_l}$, used in each layer of the CNN having height C and width R, are convolved over the input array of height N and width M. The output of this convolution operation is a feature map of size $N-C+1 \times M-R+1$. Depending on the number of kernels per convolution layer, there can be numerous feature maps per layer. For a more formal description, consider the element $x_{ij}^{lk}\left(\xi\right)$ at location (i, j) in the kth feature map of layer l for charge profile ξ, given as follows;

$$\begin{aligned} x_{ij}^{lk}\left(\xi\right) &= \eta\left(\sum_{m}\sum_{c=0}^{C_l}\sum_{r=0}^{R_l}(w_{rc}^{lkm}h_{i+c,j+r}^{l-1,m}\left(\xi\right) + b^{lk})\right) \\ \mathbf{h}^{lk}\left(\xi\right) &= \mathcal{S}\left(\mathbf{x}^{lk}\left(\xi\right)\right) \end{aligned} \tag{3.9}$$

where,

$$\begin{aligned} \mathbf{h}^{lk}\left(\xi\right) &= \mathbf{\Psi}\left(\xi\right), \text{ for } l = 1 \\ \eta &= max(0, y) \end{aligned} \tag{3.10}$$

In the above composite function, m is the feature map in layer $l-1$, b^{lk} is the bias for feature map k in layer l and w_{rc}^{lkm} is the value of the kernel at the (c, r) location connected to feature map k. $\mathcal{S}(\cdot)$ is a sub-sampling function, called max-pooling, which gives the maximum value of a perceived subset $\mathbf{s}$ of a feature map, where

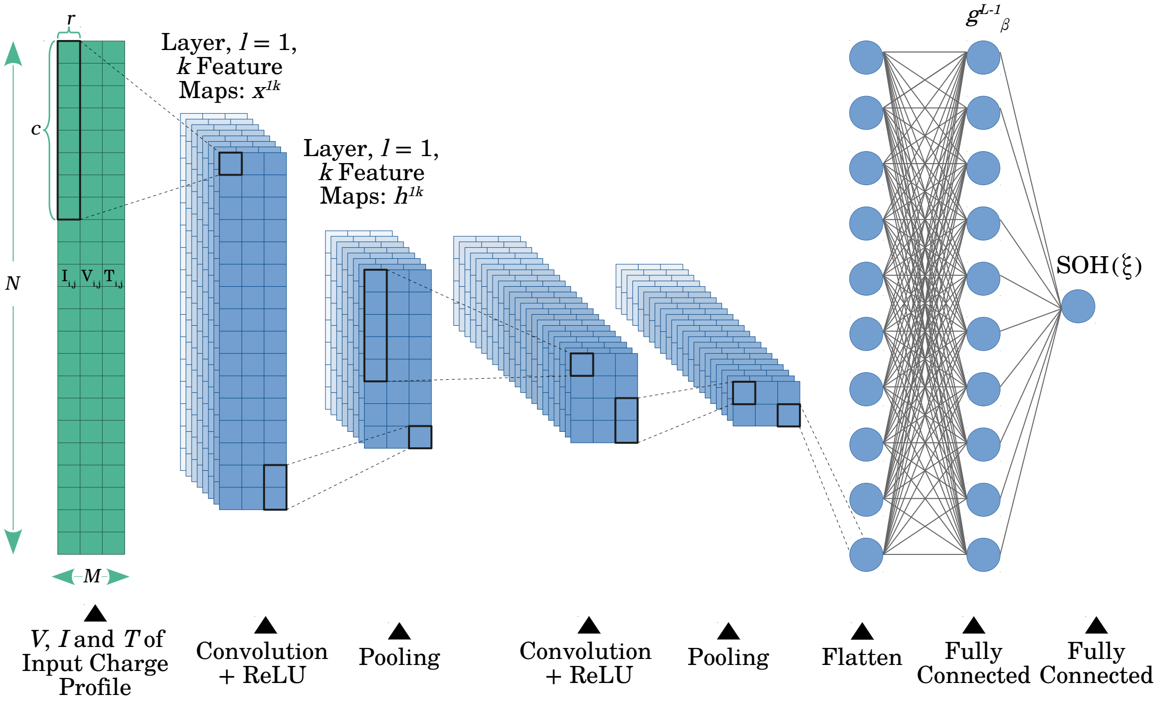

r
c
N
M
$I_{i,j}$ $V_{i,j}$ $T_{i,j}$
Layer, $l = 1$, k Feature Maps: x^{1k}
Layer, $l = 1$, k Feature Maps: h^{1k}
g^{L-1}_{β}
SOH(ξ)
V, I and T of Input Charge Profile
Convolution + ReLU
Pooling
Convolution + ReLU
Pooling
Flatten
Fully Connected
Fully Connected

Figure 3.5: Architecture of a convolutional neural network (CNN) where each layer is composed of a convolution and pooling component and with the last two layers being fully connected. The input data is given by $\mathbf{\Psi}(\xi) \in \mathbb{R}^{N \times M}$ where N=256 and M=3 since $\mathbf{\Psi}(\xi) = [\mathbf{I}(\xi), \mathbf{V}(\xi), \mathbf{T}(\xi)]$ where $\mathbf{I}, \mathbf{V}, \mathbf{T}$ represent the current, voltage and temperature of the ξ charge curve. The output of the CNN is the estimated SOH.

$\mathbf{s} \subseteq \mathbf{h}^{lk}$. The non-linearity used in this work, η, which is called the Rectified Linear Unit (ReLU), is used in this work due to its simplicity and ease of implementation in real time. The last few layers of a CNN, as observed in Figure 3.5, are fully-connected layers, formally described as follows;

$$g^{l}_{\beta}\left(\xi\right) = \eta\left(\sum_{\beta}(w^{l}_{\alpha\beta}h^{l-1}_{\beta}\left(\xi\right) + b^{l}_{\beta})\right) \tag{3.11}$$

where,

$$g^{l}_{\beta}\left(\xi\right) = SOH\left(\xi\right), \text{ for } l = L \tag{3.12}$$

and where $w^{l}_{\alpha\beta}$ denotes the weight connection between neuron α in the $(l-1)$th layer and neuron β in the lth layer, b^{l}_{β} and h^{l}_{β} are the bias and activation function at layer l, respectively. The total number of layers in the CNN is given by L. To determine the SOH estimation performance of the CNN for a particular charge curve ξ, the estimated state of health, $SOH\left(\xi\right)$, is compared to the state of health ground-truth value, $SOH^{*}\left(\xi\right)$, resulting in an error value. The loss function is simply the mean squared error computed from all the individual errors, as follows;

$$\mathcal{L} = \frac{1}{\varXi}\sum_{\xi=0}^{\varXi}\left((SOH(\xi) - SOH^{*}(\xi))^{2}\right) \tag{3.13}$$

where $\varXi$ is the total number of charge curves in the training dataset.

The advantages of convolutional neural networks over other estimation strategies is their ability to detect subtle variations and dependencies in the training data as in the case of the plateau region in the voltage-capacity curve of most Li-ion batteries. As such many strategies, some of which are mentioned in Section 2.3.4, require the use probability density functions and incremental capacity analysis to extract aging signatures. In addition, obtaining the incremental capacity curve directly from noisy battery signals is very difficult and is not a reliable method for on-board application. Therefore, most methods using ICA require further curve fitting or in some cases the use of support vector machines to identify the incremental capacity curves. SOH estimation is then performed by using lookup tables. CNNs can ingest raw battery data and output a single estimated SOH value, removing the need for data preprocessing, battery models, or intermediate calculations like ICA and probability density functions. In addition, the inference accuracy of CNNs is typically increased by adding statistical variances to the measured training data. When used for SOH estimation, this method not only increases inference accuracy but also minimizes the CNNs sensitivity to measurement noise, offsets and gains resulting in a robust algorithm capable of reliable on-board operation.

State of health estimation performance is evaluated with different metrics. These include Mean Absolute Error (MAE), Root Mean Squared Error (RMS), Standard Deviation of the errors (STDDEV) and the Maximum Error (MAX).

The CNN models discussed in this section utilize TensorFlow. To expedite the training process two NVIDIA Graphical Processing Units (GPU) are used to capitalize on the ability to perform parallel computing on CNNs. The GPUs used are the NVIDIA TITAN X and the GeFORCE GTX 1080TI. Although GPUs are used in the training process, this is only to accelerate this process. When performing inference in real time, these CNNs are suitable for traditional embedded devices.

3.2.5 Convolutional and LSTM Recurrent Neural Network Fusion for SOH and SOC Estimation

As an extension to the work performed on SOH estimation using the CNN algorithm, a fused recurrent convolutional neural network (CNN-RNN) is used for the first time to perform combined state-of-health and state-of-charge estimation on controlled charge profiles. Performance and results are shown in Chapter 7. This is conducted by feeding the SOH estimated by the CNN into the LSTM-RNN which, in turn, estimates SOC, as shown in Figure 3.3. For a more in depth look at the theory behind CNNs and LSTM-RNNs, please refer to Section 3.2.4 and Section 3.2.2. Each network is trained separately and applied in unison during testing. It is important to note that the networks operate on separate time-frames. The CNN uses reference charge profiles, ξ, to estimate SOH, hence it awaits a charge event to occur in order to execute an SOH estimation. These charge events can take place once a day or even once every few days. However, in the case of SOC, the LSTM-RNN is required to perform an SOC estimation at 0.1-1Hz depending on the energy management system's sampling frequency. Therefore, between charge events, a fixed SOH value is provided as an input to the LSTM-RNN. The inputs to the LSTM-RNN then becomes $\Psi_k = [V(k), I(k), T(k), SOH(\xi)]$ where $V(k), I(k)$ and $T(k)$ represent the voltage, current and temperature of the battery at time step k and $SOH(\xi)$ is the SOH estimated during the last charge event. Choosing the LSTM-RNN instead of the DNN technique, described in Section 3.2.1, capitalizes on the efficient learning capabilities of the memory cell. In other words, fewer inputs are required from the battery at every time step, k, to perform SOC estimation when utilizing the LSTM-RNN.

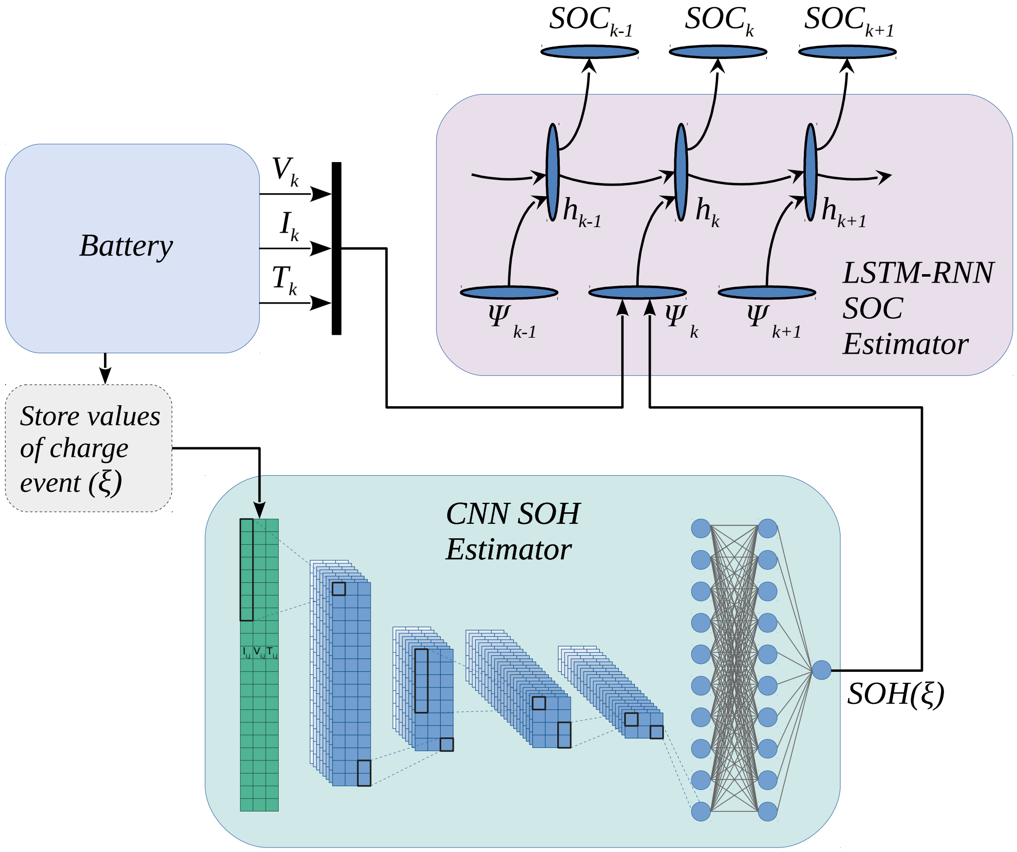


Figure 3.6: Schematic showing fusion of convolutional neural network and LSTM-RNN to perform combined SOH and SOC estimation. Charge event, ξ, data is saved and given to the CNN as inputs, as discussed in Section 3.2.4. The CNN estimates SOH and passes it to the LSTM-RNN which, in turn, estimates SOC at every time instant k, as described in Section 3.2.2. Input vector is given by $\Psi_k = [V(k), I(k), T(k), SOH(\xi)]$ where $V(k), I(k)$ and $T(k)$ represent the voltage, current and temperature of the battery at time step k and $SOH(\xi)$ is the SOH estimated during the last charge event.

3.3 Back-propagation and Optimization

Inference is performed with a single forward pass when data is fed into the network and the estimated SOH or SOC is obtained as the network output. Training is performed when a backward pass is performed following a forward pass where the network weights are updated by minimizing the loss function. One full training epoch, ϵ, describes a single cycle of one forward pass and one backward pass. A forward pass starts when the training data is fed into the network and is complete when the state is estimated at the output and when the overall loss function is determined. A backward pass describes the process whereby the loss signal is sent back through the network and the gradient of the loss with respect to the network weights is computed in order to update the weights. In this work, training does not cease till a specified threshold criteria of loss is attained. In addition, during the backward pass, the gradient of the loss function with respect to the weights is used to update the network weights in an optimization method called *Adam* [167], which is also performed with the biases. This is given in the following composite function;

$$
\begin{aligned}
u_\epsilon &= \gamma_1 u_{\epsilon-1} \nabla \mathcal{L}(w_{\epsilon-1}) \\
v_\epsilon &= \gamma_2 v_{\epsilon-1} \nabla \mathcal{L}(w_{\epsilon-1})^2 \\
\widetilde{u}_\epsilon &= u_\epsilon / (1 - {\gamma_1}^\epsilon) \\
\widetilde{v}_\epsilon &= v_\epsilon / (1 - {\gamma_2}^\epsilon) \\
w_\epsilon &= w_{\epsilon-1} - \rho \frac{\widetilde{u}_\epsilon}{\widetilde{v}_\epsilon - \kappa},
\end{aligned}
\tag{3.14}
$$

where γ_1 and γ_2 are decay rates set to 0.9 and 0.999, respectively, $\rho = 10^{-5}$ is the learning rate and κ is a constant term set to 10^{-8}. The network weights at the present training epoch are given by w_ϵ. A normally distributed random number generator is used to initialize the network weights. During the backward pass, the network self-learns its network weights and biases, a process referred to as backpropagation, which is a remarkable difference to other methods that demand time-consuming hand-engineered battery models and parameter identification.

Once the threshold criteria is met where the loss energy of the network is minimized, training is ceased and the network can be applied in real time. Inference is performed when the network is applied to the validation dataset, which was never seen during training.

3.4 Data Fuels Learning

It's no surprise that with deep learning algorithms like all other learning algorithms, data is as critical as the algorithms themselves. If the data used for learning is not representative of the underlying system, then the model will not output the desired response. Often, this is a result of a misinterpretation or a misunderstanding of the data or features within the dataset rather than the data itself. A lack of knowledge of the data can make it difficult to choose the appropriate algorithm for the task at hand or cause unnecessary preprocessing. Excessive preprocessing or over-processing the data can result in the data loosing its meaning. Also choosing the inputs which most correlate with the desired output is key when choosing features to feed into the network.

3.4.1 Data Preprocessing in the Context of SOC and SOH Estimation

For SOC and SOH estimation, the data was only preprocessed when necessary. In other words, since these algorithms are aimed for on-board application, reducing the number of operations performed to achieve the desired output becomes an important consideration. And this number of operations clearly also includes data preprocessing. Given that these techniques are used with battery data which includes voltage, current and temperature, it was sometimes necessary to scale this data. This is a consequence of these inputs having different scales, e.g. voltage can span 2.5 V to 4.2 V while temperature can span -20 to 25°C. In the following chapters, full disclosure will be given wherever scaling is performed. Scaling, here, is defined by;

$$x_{scaled} = \frac{x_{raw} - x_{min}}{x_{max} - x_{min}} \tag{3.15}$$

where x_{raw}, x_{min} and x_{max}, represent the raw input value, the minimum input value and the maximum input value, respectively. In the chapters discussing SOC and SOH estimation, the actual value which is used to asses the estimation error is often referred to as a ground-truth value or the labeled dataset. The ideal or ground-truth values are written with a star notation, e.g. SOC^* or SOH^* in Equation (3.5), Equation (3.8) and Equation (3.13). The sampling frequency of the raw data is also an important consideration. Although the raw data is sampled at a particular frequency in lab settings, due attention is paid to ensuring that this sampling rate is accurate. Therefore, for most datasets used int his work, a liner interpolation of the data is typically performed so that the networks experience the inputs at a fixed rate.

3.5 Network Construction and Hyperparameter Tuning

3.5.1 Choosing Network Types

An appropriate choice of networks will always depend on the nature of the data as well as the computational and memory constraints. The data used for Li-ion battery SOC estimation is a time series and, therefore, recurrent neural networks are most suitable for this type of application due to their ability to encode dependencies in time. However, as will be shown in Chapter 6, DNNs can achieve similar performance. However, this is only possible if antecedent values are provided to the network as an input. In addition, data augmentation described also in Chapter 6, is also required for DNNs in order to achieve further estimation accuracy and operation at various ambient temperatures.

When considering SOH applications, the strategy defined in Chapter 7, uses reference profiles which reoccur at regular intervals of the battery's lifetime. The objective of the algorithm becomes to issue a desired estimate of SOH when given the full reference profile. This is starkly dissimilar to the objective of the SOC estimation algorithm which issues an estimated SOC when given a discrete value of the battery signals. Therefore, the SOH algorithm must ingest a much larger corpus of data, i.e. the reference profile, before it issues one SOH estimate. This algorithm must be able to detect correlations between features in one or both axis of this two dimensional dataset. These are traits of a convolutional neural network and thus why this algorithm is used in Chapter 7. An LSTM-RNN could equally be used since it could also ingest a complete reference profile before issuing an estimate.

3.5.2 Hyperparameter Tuning

Training deep learning models is an iterative process. Various components are important to consider for different network types. For the DNN, LSTM-RNN and the CNN algorithms utilized in this work, different aspects are considered. We call these aspects hyperparameters. They are entities which are typically not learned during the training process but need to be predefined. They are usually determined by setting various different values, training different models and choosing the values which show best performance on training and test datasets.

Choosing the number of neurons and number of layers of a deep learning model is an iterative process. Starting with a small number of layers and neurons, one can check if the loss over the validation dataset is being minimized and that the network is properly fitting the data. Typically, the learning rate is a value ranging from $1e-2$ to $1e-6$. When testing various network architectures, an appropriate learning rate can be found by iterating its value by $1e1$ in consecutive trials. On average, the loss of the network should always be decreasing from one training epoch to the next. If the loss of the network is not observed to be decreasing, then a smaller learning rate might be necessary.

In the case of DNNs used for SOC estimation, much of the performance gains can be achieved during the data processing phase, where the data is not only cleaned but also scaled and augmented.

The depth in time of an LSTM-RNN is an important hyperparameter to tune since it not only allows the network to converge to a good solution but it also provides an indication of the dynamics in time which dictate the behavior of the underlying system. In the case of this work, this system is a battery and we've seen in Chapter 5 that good performance is typically achieved when the network has a depth in time of 100 to 1000 time steps.

In the case of CNNs used for SOH estimation, there are additional hyperparameters which need to be considered. Firstly, mini-batches are used because of memory constraints on the GPUs. In addition, using mini-batches allows for more frequent updates to the network weights [168]. With standard gradient descent, updates are performed to the network weights after the entire training dataset is processed. With mini-batch gradient descent, smaller batches of the training dataset are selected and processed to issue an update to the network weights. Therefore, the weights can be updated more frequently. Although, for smaller training datasets, below approximately 2000 data points, mini-batches might not offer the same advantage as it can for much larger datasets. In the case of the CNN approach for SOH estimation, in many cases, the training datasets were composed of hundreds of thousands of data points. Hence using mini-batches offered faster training. Mini-batches of size 64 to 1024 is typical when training deep learning algorithms. Setting a mini-batch size value as a power of two can help decrease training time. This can be attributed to the way computer memory is laid out and accessed. The mini-batch size is considered to be another hyperparameter which needs to be varied through different iterations of the network training in order to achieve the size which makes the gradient descent optimization algorithm as efficient as possible.

3.5.3 Maximizing Performance from Deep Networks

Generally, after performing the above steps, if the deep learning model is still not converging to a proper solution, there are additional techniques which can be utilized to generalize the model and allow it to fit the data. One obvious method is to use more data but it is generally not an easy task to obtain additional data. The following techniques can help generalize models and overcome cases of over-fitting.

Batch Normalization

For the DNN algorithm, scaling the inputs to lie between 0-1, helped to quicken the learning process. Similarly, setting the activation values at the output of a hidden layer, to lie in a similar range of values, can also reduce training time. Batch normalization is such a process where the activation values are normalized at the output of each hidden layer to have a certain mean and variance [169]. This process is found to help with the computational time of DNNs. This can be attributed to the fact that the network does not have to learn to represent large covariant shifts in the activation values from one hidden layer to the next which can help accelerate training.

Dropout

Another strategy used to achieve further performance from the CNN used in SOH estimation is dropout [170]. This strategy randomly hides certain nodes of the network during the training process since various nodes connected to these features might be hidden at any given training epoch. Therefore, the training data are exposed to different combinations of hidden nodes within the network throughout training. This regularization technique does not allow the network to settle around one local minimum of the solution space, and requires the network to look for neighboring local minima having lower loss. Dropout also does not allow the network to heavily rely on one feature or one type of input during the training process. Therefore, when various different nodes and features are blocked during training, the network is obligated to spread it's reliance over all or most of the feature space.

Early Stopping

Early stopping is also a method which is used on any deep learning algorithm and was commonly used in this work. This describes the process of monitoring the training loss and the validation loss as a function of the training epochs. Overfitting can be observed, in most cases, when the training loss continues to decrease while the validation loss stars to increase. In early stopping, the training is stopped as soon as the validation loss starts to increase.

Skip Connections

In 2015 winners of the detection task in the ImageNet competition, used a deep residual network [148]. This popularized the idea of residual layers which use skip connections. In this work, in certain cases, skip connections were used with the DNN models to improve representational power. Although they have been useful for CNN models, they can be used for any feedforward-type of network including DNNs. Typically in most feedforward-type models, the activations of the previous hidden layer, $l-1$, is always fed to the next hidden layer. In some cases, using features from other previous hidden layers, say $l-2, l-3, ...$, can be meaningful to the network and help in achieving better representational power and generalization.

Chapter 4

Towards Data-driven and Neural Computation-based State Estimation Framework

4.1 Expressibility of Machine Learning Algorithms

Data-driven methods for battery state estimation are becoming more wide-spread. This is likely due to the increasing number of readily available battery datasets. With a rising influx of open-source battery data, non-parametric models which are not limited to a specific number of parameters, do not have a natural bound on their expressibility. In other words, non-parametric approaches like artificial neural networks allow model expressibility that can be naturally adjusted to the underlying phenomenon. Of course, this is possible when sufficient data is sampled from this underlying process. Thus, problems of variable complexity become addressable by non-parametric methods, given sufficient data is available.

Neural computation-based approaches in machine learning and deep learning are examples of such non-parametric methods. They are able to express complex non-linear systems, provided sufficient data is available. In addition, the number of parameters or network weights expressing the underlying modeled phenomenon is not fixed and can be increased or changed depending on the complexity of the system. In a recent study, it was found that neural networks with finite numbers of network weights can express any multivariate polynomial effectively [171]. Expressibility is an important quality of any battery model or state estimator. As discussed previously, batteries are highly non-linear systems of variable complexity, exhibiting, at one instant, charge/discharge profiles of lower dynamics and constant ambient temperature and, at another instant, highly dynamic drive cycles with fluctuating ambient temperatures. The advantages of non-parametric models like neural networks, outlined above, are a good match for the range of behaviors a battery model or state estimator is required to express at any given moment as well as throughout its lifetime. In the next few chapters, deep learning algorithms are shown to have good expressibility when performing state-of-charge and state-of-health estimation over a range of variable dynamics, ambient conditions and, in certain cases, over partial datasets. This affords a certain practicality to using machine learning models since modeling battery behavior of variable complexity can be expressed with networks of variable size, provided sufficient data is available.

4.2 Efficiency of Deep Learning Models

Like expressibility, efficiency is also critical to the representation of complex battery behaviors. Efficiency in the context of deep learning describes the amount of resources required to express a non-linear system. These resources can include the number of neurons, the number of layers or the number of network weights. Although there is currently no evidence that deeper networks acquire any additional expressibility, there are a few studies which investigate the increase in efficiency accrued with deeper networks [171; 172; 173; 174]. Specifically, these studies show that polynomial or exponential improvements in efficiency are obtained with deeper networks [171; 175].

Efficiency becomes particularly important in the context of SOC and SOH estimation since processing power in mobile applications like electrified vehicles tends to be limited. To understand this further, state-of-charge estimation using feedforward neural networks is used as an example and an analytical representation of the computational efficiency is given. Since most of the computational cost is due to floating-point operations, then it follows that the computational time of a forward

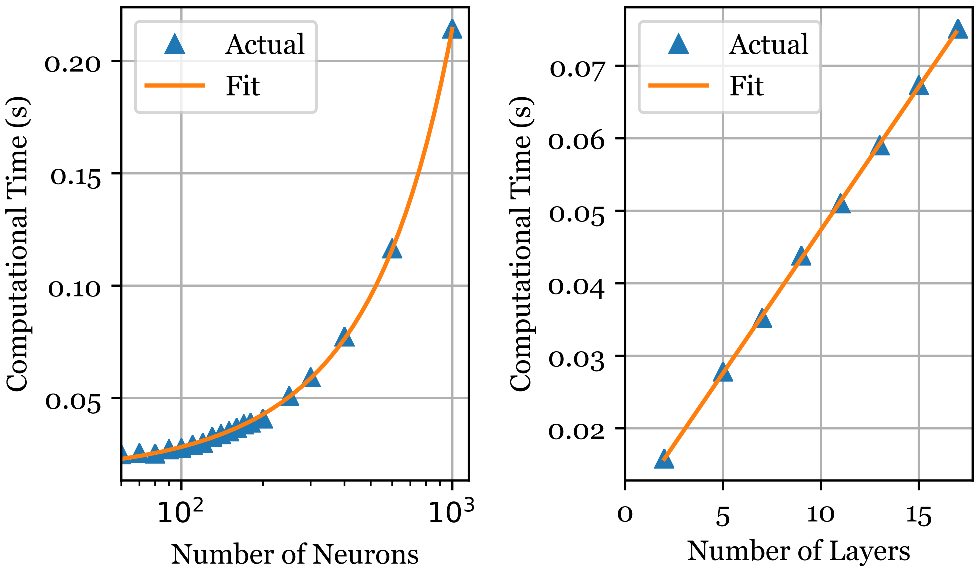


Figure 4.1: Experimentally verified relationships of computational time vs. number of neurons as well as vs. number of layers. (right) Semi-log plot of computational time vs. the number of neurons. The number of layers in the DNN is fixed to 4 layers. (left) Computational time as a function of the number of layers. The number of neurons in each layer of the DNN is fixed to 50 neurons.

pass is proportional to the number of multiplication and addition operations. Referring to Figure 3.2, for each neuron of the DNN, there exists d multiplications and $d-1$ additions resulting, ideally, in $2d-1$ operations per neuron. Let N be the number of neurons in each layer, τ be the total number of time steps in the sequential dataset and L be the number of layers in the DNN, then the number of floating-point operations is given by;

$$F(\tau, L, N) = \tau \left[N^2(2L-4) + N(2d-L+3) - 1 \right] \tag{4.1}$$

Therefore, when letting τ and L remain constant, it is immediately clear that the number of floating-point operations grows as the square of the number of neurons or in Big-O notation as $\mathcal{O}(N^2)$. On the other hand, if τ and N remain constant, and the number of layers is allowed to vary, then it follows that the number of floating-point operations grows linearly with the number of layers, L, or in Big-O notation as $\mathcal{O}(L)$. Hence, it is typically preferred that networks grow in depth rather than in width to reduce computational time. This relationship is tested by timing the forward pass for varying numbers of neurons as well as for varying numbers of layers and are shown in Figure 4.1.

4.3 Automating the Battery Modeling Process

Traditional framework performing SOC and SOH estimation include numerous steps. Generally these steps can be summarized into three main categories; the battery test data generation step is required to perform parameter identification in the battery modeling step which, in turn, is required to conduct battery state-of-charge and state-of-health estimation amongst other tasks in mobile applications. The battery data generation is an important step where a slew of charge and discharge pulsed as well as dynamic profiles are used to characterize the battery. These were shown as the pulse discharge/charge tests in Figure 2.4 where a few hours of relaxation is required for the battery to reach equilibrium. The tests are typically specific in nature so that certain electro-chemical coefficients can be extracted from the data. Furthermore, to estimate the SOC of a battery, the OCV-SOC relationship, shown in Figure 2.5, is often required and is usually obtained at a C-rate of C/10 or C/20 which can be a time-consuming process [176]. The combined time required to perform the pulse tests and the OCV-SOC relationship is on the order of days. In the case of neural computation-based battery models described in Chapter 3, no specific battery tests are required. Any data which spans the lower dynamics as well as the higher ones can be used to train a neural network. This can be drive cycle data, generated in the lab or in real time, as well as pulsed charge/discharge data. There are no constraints on the type of data which can be used to train a neural-based battery model.

On the battery modeling front, most commonly used models are parametric such that the number of model parameters required to express battery behavior is typically fixed. For some of these approaches, adding parameters for better model expressibility requires a better understanding of the underlying electro-physical processes. To improve model expressibility in these types of models, it is often required to further incorporate into the model the representational power of additional electro-physical processes.

In the case of equivalent circuit models, additional resistor-capacitor branches (RC) represent the dynamic potential differences across electrode and electrolyte of the battery. However, internal mass transport through diffusion causes the battery cell to react at a spectrum of time constants ranging from seconds to hours [107; 177]. Therefore, it becomes difficult to capture these wide-ranging time-constants with one or a handful of RC branches. Furthermore, modeling other processes like self-discharge, for example, will also require additional resistance elements, typically, positioned in parallel to the output voltage. Electro-chemical models are considered to have a higher affinity to the underlying electro-physical processes and, hence, can have better representational power with respect to Li-ion batteries, although some proponents of equivalent circuit models might argue against this. Nevertheless, research around the topic of electro-chemical li-ion battery models and their applicability in electrified vehicles is very popular nowadays. Electro-chemical models contain numerous partial differential equations, representing diffusive mass transport, Fick's Law and charge conservation, to name a few. These can be challenging to implement on an embedded device when considering the limited processing power available in energy management systems of electrified vehicles. In addition, a plethora of parameters, like the solid lithium concentration are required to accurately model the battery [108; 18; 98]. Many of these parameters are not only battery chemistry specific but can also be specific to the battery make and model. Further adding to these drawbacks are particular coefficients like the diffusion coefficient and maximum solid lithium concentration which can be proprietary in nature and thus are not typically provided to the public. This can be a large obstacle when formulating these electro-chemical models.

For the machine learning-based models discussed in Chapter 3, additional model parameters or network weights can be used to increase model expressibility. However, in this case, it is not a requirement for these additional parameters to represent any underlying electro-physical process. Therefore, the time consumed by ensuring that the model parameters have some affinity to the underlying electro-physical processes is avoided. In other words, the time-consuming process of hand-crafting battery models can be avoided. Furthermore, neural computation-based models are not dependent on electro-physical coefficients and stoichiometric parameters, as in the case of electro-chemical models, which allows most anyone to achieve a high fidelity model without the need for hard-to-obtain proprietary information.

4.4 One Network to Rule Them All: Unifying Battery Modeling and State Estimation

As mentioned in the previous subsection, traditional SOC and SOH estimation framework are composed of three fundamental steps; the battery test data generation step is required to perform parameterization during the battery modeling step which, in turn, is necessary to conduct SOC and SOH estimation. The third step, where state estimation is performed, usually contains some form of inference or filter algorithm since SOC and SOH are not directly observable quantities. These can include adaptive observers, sliding mode observers and Kalman Filters, which have been discussed in greater detail in Chapter 2. Kalman filters are typically more popular with respect to SOC estimation. In SOH estimation, as discussed also in Chapter 2, many strategies include the use of historical degradation data or probability density distributions and incremental capacity analysis of characteristic battery profiles. These estimation strategies are usually critical in obtaining accurate and reliable SOC and SOH values.

Here, it is proposed that the battery modeling step and the state estimation step can be unified and expressed by a deep learning model. In other words, with machine learning and deep learning algorithms, it becomes possible to map raw battery signals directly to SOC and SOH values. This removes the need to not only hand-engineer a battery model but also the need to hand-craft commonly used estimation methods. For example, the time-consuming co-variance matrix of the Kalman filter is not necessary when using deep learning strategies. Also, as will be discussed further in the next few chapters, very little if any preprocessing of the data is necessary in order to obtain a reliable estimate. From the previous few subsections, it becomes clearer that the representational power of deep learning techniques can allow them to express both the battery model and the inference system necessary to estimate SOC and SOH directly from the raw battery signals. This can be a powerful utility of deep learning algorithms which can not only further automate the steps involved in the estimation process but also reduce the workload necessary to perform all the steps in traditional SOC and SOH estimation frameworks.

4.5 Distilling Information from Irrelevant Noise and Reducing Model Rigidity

Traditional machine learning techniques contain no more than one or two layers of non-linear and linear transformations [149]. Support vector machines (SVMs) are an example of these types of architectures. They are suitable for solving simpler, well-defined problems however they could suffer from a lack of expressibility when applied to dynamic and noisy real world applications [178; 149].

Deeper architectures are capable of taking raw input data and send it through a series of non-linear and linear transformations such that at each transformational layer a larger amount of abstractions are extracted from the input data. These abstractions are generally referred to as features and at each consecutive layer, features of previous features are extracted. Thus, a deep architecture is able to learn a hierarchy of features that represent complex and highly non-linear functions [179; 180]. Deep learning algorithms are particularly adept at representing input data in this hierarchical structure and distilling the important information from irrelevant and noisy data. This becomes particularly important for SOC and SOH estimation in electrified vehicles where most battery signals contain not only measurement noise but also measurement offsets and gains. This feature of deep learning is tested in this thesis by injecting noise, offsets and gains into the measured battery signals and will be discussed further in the next few chapters. Interestingly, two important benefits have surfaced from this feature. As mentioned, the training data was augmented by injecting white Gaussian noise, offsets and gains into the measurement signals to first robustify the deep learning algorithms. As will be shown in the next chapters, augmenting the training data can not only robustify the deep learning algorithms, but can also increase their estimation accuracy. As it turns out, adding statistical variances to the training dataset in order to break the symmetries within the network and reduce model rigidity is a popular practice in the machine learning community and can further reduce the loss of the network [181; 182; 183; 170]. Allowing the facile implementation of robustness while simultaneously increasing estimation accuracy provides yet another powerful utility of deep learning algorithms which can be greatly beneficial in real world and noisy environments.

4.6 Increasing Model Universality through Data

One can capitalize on the expressibility and efficiency of deep learning algorithms to expose a network to all possible scenarios in its perceived environment. In other words, the universality of a deep learning model can be increased by exposing it to any information spanning the environment of the modeled system. In the case, of SOC estimation for Li-ion batteries, this is very beneficial since one deep learning model can be supplied a plethora of training data spanning numerous ambient temperatures and a variety of slow and fast dynamics, as will be shown in the next chapters. In the case of SOH estimation, different ambient temperatures can be used but, more importantly, different ranges of the reference profiles can be used as well. This increases the practicality of the algorithm since it is not required to supply the algorithm with a fixed range of data in order to estimate the SOH. Since the reference profiles used in this work are charge curves, this means that the deep learning model can estimate SOH for most partial charge curves.

Chapter 5

Accurate State-of-Charge Estimation Using Long Short-Term Memory Networks

5.1 Efficacy of Recurrent Neural Networks on Battery Data

As discussed in Chapter 3, Recurrent Neural Networks with Long Short-Term Memory (LSTM-RNN), are effective learning algorithms when considering time series data. Since battery data is very much time-dependent, it is natural to consider LSTM-RNNs when conducting state estimation. This is attempted here for state-of-charge estimation. To be able to estimate the SOC of a battery, it is important to know antecedent information. In most works which use neural network-based techniques for SOC estimation, some form of antecedent information is used to ensure an accurate estimation [133; 131; 184]. However, in the case of the LSTM-RNN, the memory cell stores the characteristic behavior of the battery which it learns from the training data, hence there is no need to use any antecedent data to make an accurate estimation. As will be seen, voltage, current and temperature at the present time step is solely needed. This allows LSTM-RNNs to be more efficient in the type of data they require to learn expressibility. Furthermore, the recurrent structure of the network and the ability to store intrinsic behavior of the battery in its memory cell, can allow representational learning on sparse datasets, as will be shown below. However, because of it's recurrent structure, LSTM-RNNs can take longer to train.

It is important to mention that, there is no preprocessing performed on the data used to train and validated the LSTM-RNN. The raw data is fed through the network to obtain an estimated SOC. This is another benefit of using an LSTM-RNN since this removes some of the necessary computation if this algorithm is implemented in an on-board application. In addition, the LSTM-RNN, as will be shown in the next sections of this chapter, has the ability to encode the characteristic behavior of a battery at numerous ambient temperatures thus reducing the memory required to store different parameters for different ambient temperatures as is typically done for traditional battery models. Therefore, these latter advantages make LSTM-RNN a great candidate to perform estimation on many cells in a battery pack.

To evaluate the performance of the LSTM-RNN at SOC estimation a few different metrics are used. These include the Mean Absolute Error (MAE), the Root Mean Square Error (RMS), the Standard Deviation (STDDEV) and the Maximum Absolute Error (MAX).

5.2 Experimental Data

5.2.1 Drive Cycle Data for Training and Validation

A Panasonic 18650 battery cell with a lithium nickel cobalt aluminum oxide ($LiNiCoAlO_2$ or NCA) chemistry, similar to the cell used in some Tesla vehicles, was tested [185]. The battery, which is rated to have 43 mΩ dc resistance is described in Table 5.1 [185; 186]. All the testing was performed in a thermal chamber with cell testing equipment manufactured by Digatron Firing Circuits, as described in Table 5.2 and shown in Figure 5.1a).

To generate training and validation data for the LSTM-RNN, the battery was exposed to a selection of drive cycles at ambient temperatures ranging from 0 to 25°C. A set experimental procedure was used, as is described in Figure 5.1b). A schematic of the experimental apparatus is shown in Figure 5.1c). The thermal chamber was first set to 25°C followed by a 3 hour pause to allow the battery's internal temperature

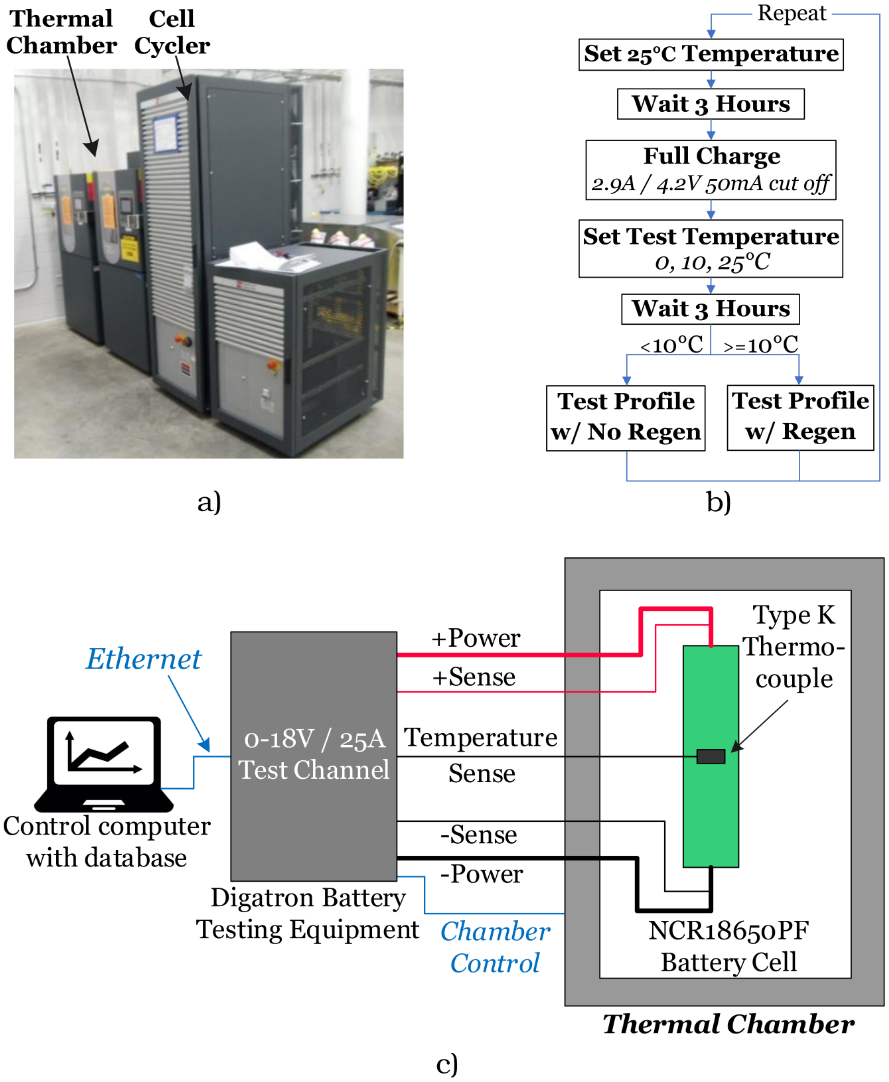


Figure 5.1: a) Equipment used for battery testing, b) battery test procedure and c) schematic of the test bench and data logging system.

Table 5.1: Panasonic 18650PF Cell Parameters

Nominal Open Circuit Voltage	3.6V
Capacity	Min. 2.75 Ah / Typ. 2.9 Ah
Min / Max Voltage	2.5V / 4.2V
Mass / Energy Storage	48g / 9.9Wh
Minimum Charging Temperature	10 °C
Cycles to 80% Capacity	500 (100% DOD, 25 °C)

Table 5.2: Test Equipment Specifications

Cycler Manufacturer	Digatron Firing Circuits
Test Channel Used	25A, 0-18V channel
Voltage / Current Accuracy	+/- 0.1% Full Scale
Data Acquisition Rate Used	10Hz
Thermal Chamber	Cincinatti Sub Zero ZP-8
Size	8 cu. Ft.
Accuracy	+/-0.5 °C

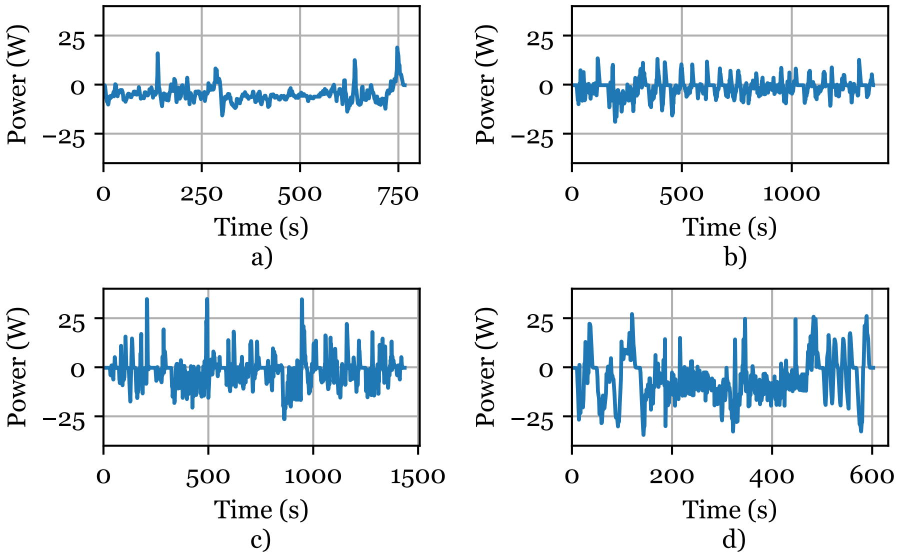


Figure 5.2: Drive cycle power profiles, scaled for a single cell of 35kWh pack for a Ford F150 electric truck. a) HWFET, b) UDDS, c) LA92, d) US06.

to reach 25°C. The battery was then fully charged with a constant current charge rate of 2.9 A (1C) followed by a constant voltage charge at 4.2 V which was terminated when current fell below 50 mA. The thermal chamber temperature was then set to the desired ambient temperature to record the subsequent drive cycle, and, in turn, this was paused for three hours to allow the battery's internal temperature to reach steady state. The battery is then discharged using a drive cycle power profile which includes regenerative braking (i.e. charging) power if the ambient temperature is greater than or equal to 10°C, the minimum temperature for which the battery is rated to charge. The current sensor that is used to measure current and to calculate capacity has an error of less than 25mA. For the typical dataset, this sums to a cumulative error of under 40mAh which is small compared to the 2900mAh of battery capacity.

During experimentation, the battery was exposed to 10 drive cycles. Each dataset consisted of a random combination of different drive cycles which included HWFET, UDDS, LA92 and US06. Constructing these unique datasets which were composed of various drive cycles, having a spectrum of different dynamics, provided the LSTM-RNN with a broad range of realistic driving conditions. These 10 cycles were applied on the battery at three different ambient temperatures (0, 10, or 25 °C). Training of the LSTM-RNN is performed on a subset of these 10 cycles (typically 8 to 9 cycles) and will henceforth be referred to as the training data while validation is performed on a completely different subset of cycles (usually 1 or 2) which are henceforth referred to as test cases. An additional test case, called the Charging Test Case, is recorded at 25 °C to examine the network's performance over a charging profile. Furthermore, a second additional test case is recorded during experimentation which exposes the battery cell to an ambient temperature increasing from 10 to 25 °C and is used

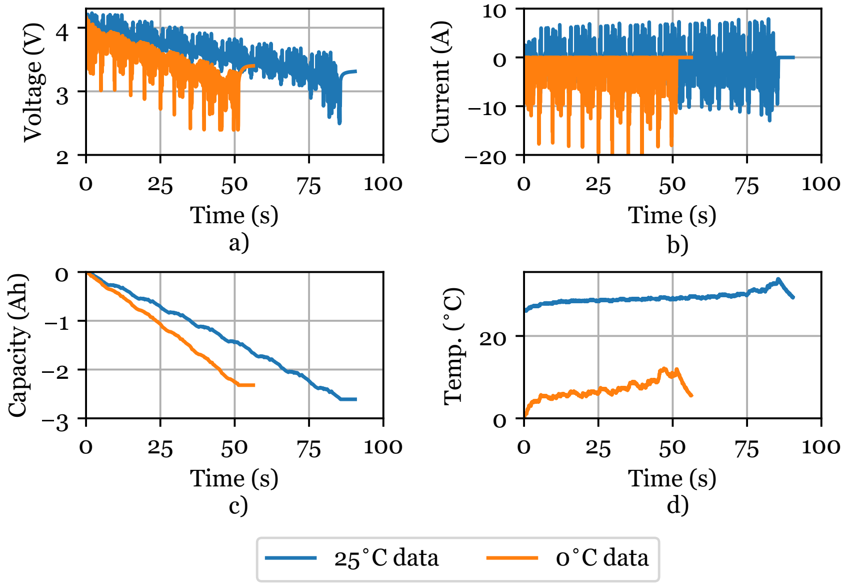


Figure 5.3: US06 drive cycle recorded at an ambient temperature of 25°C and 0°C. The following measured quantities are shown, Voltage (top left), Current (top right), Amp-hours (bottom left) and battery surface temperature (bottom right). Charging is not recommended by manufacturer below an ambient temperature of 10°C thus no regenerative braking is performed at or below this temperature.

to validate the LSTM-RNN's ability to adapt to a varying ambient temperature. The drive cycle power profiles used are for an electric Ford F150 truck [187; 188], with the power profile scaled for a single cell of a 35 kWh pack consisting of 3,680 of the Panasonic NCR18650PF cells. The power profile for the drive cycles has discharge power (negative power) as great as 40 W per cell and charge power (positive power) as great as 35 W per cell, as is shown in Figure 5.2. This translates to peak current of around 18 A, or 6C, which is a fairly typical peak C-rate for an electric vehicle application. The power profile is repeated until the battery's usable capacity is depleted. This is defined as 2.61 Ah when discharged at an ambient temperature of 25 °C, 2.5 Ah at 10 °C and 2.32 Ah at 0 °C.

The measured voltage, current, amp-hours, and battery surface temperature are shown in Figure 5.3 for the US06 drive cycle at an ambient temperature of 25 °C and 0 °C. At 25 °C the current is relatively low, $\leq$3C, and the battery voltage drop is not too significant. The temperature, measured at the surface of the cylindrical cell with a thermocouple, increases by a few degrees over the first 70 minutes of the drive, and then spikes to 34 °C as the terminal voltage falls, resistance increases, and the loss therefore increases. At 0 °C, the battery resistance is much higher, resulting in significant voltage drop as is shown in Figure 5.3. The higher resistance coupled with no regenerative braking energy due to the low temperature state, results in the drive cycle being repeated for 50 minutes at 0 °C, while this was repeated for 85 minutes at 25 °C. The battery also heats significantly, up to 13 °C, due to the higher loss at low temperature.

Although the LSTM-RNN showcased in this section is trained on data obtained from a Panasonic 18650PF cell, the same LSTM-RNN can be trained on any other type of battery cell. The network architecture will not change from one battery cell to another. The network might need to be retrained for a completely different battery, but it's architecture and the values of the network hyperparameters, like the learning rate, will not change. This is not any different than traditional estimation techniques which must re-parameterize the battery models for different types of batteries.

The following two subsections investigate the LSTM-RNN's SOC estimation accuracy when trained on a dataset recorded at a constant ambient temperature and at variable ambient temperatures, respectively.

5.3 State-of-Charge Estimation at Fixed Ambient Temperature

As mentioned above, the vector of inputs fed into the LSTM-RNN is defined as $\Psi_k = [V(k), I(k), T(k)]$, where $V(k)$, $I(k)$, $T(k)$ are the voltage, current and temperature measurements of the battery at time step k, respectively. The mixed drive cycles were logged at a sampling frequency of 1 Hz and they ranged roughly between 4000 and 10000 seconds long.

In this section, the network is trained on up to 8 mixed drive cycles while validation is performed on 2 discharge test cases. In addition, a third test case, called the Charging Test Case, which includes a charging profile is used to validate the networks performance during charging scenarios. The regenerative braking which results in charging currents of over 8 A, as can be seen from Figure 5.3, allows us to test the network's performance at higher momentary charge rates. Regenerative braking charging current is not included for validation test cases recorded at the

Table 5.3: SOC estimation accuracy of LSTM-RNN trained on fixed ambient temperature data

Test Case	MAE(%)	RMS(%)	STDDEV(%)	MAX(%)
Test Case 1 at 10°C *	0.807	1.075	0.710	2.819
Test Case 2 at 10°C *	1.252	1.541	0.898	4.312
Charging Test Case at 25°C **	0.688	0.856	0.510	4.000

*Results obtained with network having a depth in time of $\widetilde{N}$=500

**Results obtained with network having a depth in time of $\widetilde{N}$=1000

lower temperatures since the battery is not rated to charge at temperatures below 10°C. In the Charging Test Case, charging is performed at 1C (2.9A). There was interest in examining an even faster charge rate, but the 1C rate is already twice the datasheet's recommended charge rate so rates that are any higher than this are considered excessive and potentially dangerous. The LSTM-RNN's hidden layer has 500 computational nodes. A smaller number of computational nodes are possible, however the network can converge to a solution much quicker with a larger number of nodes while using a GPU. The computational time needed to train this network is about 5 hours.

The MAE achieved on each of the first two test cases is 0.807% and 1.252%, respectively. The MAE, RMS, STDDEV and MAX performance metrics for these three test cases are outlined in Table 5.3. The LSTM-RNN also showed good performance when tested on the Charging Test Case where the MAE and MAX achieved is 0.688% and 4.000%, respectively. The estimation performance on the Charging Test Case is shown in Figure 5.4. At the final training epoch, the error at every time step of the test case is shown in subplot 2. Also, at every training epoch, these errors over time

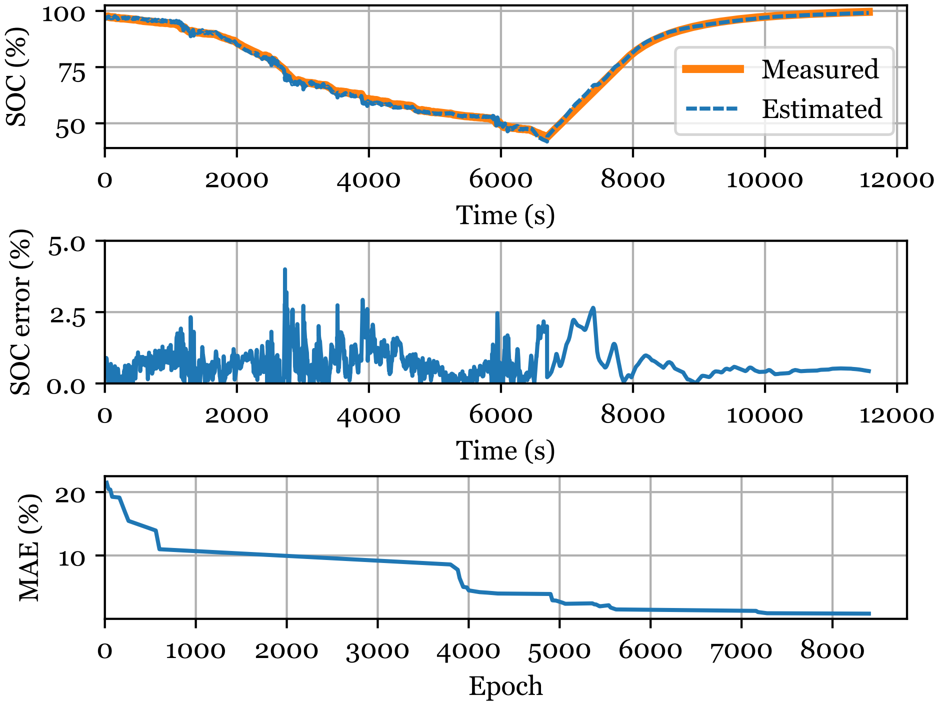


Figure 5.4: LSTM-RNN tested on the Charging Test Case which includes discharge/charge profiles, recorded at ambient temperature of 25°C. LSTM-RNN architecture: network depth in time is $\widetilde{N} = 1000$ and LSTM contains 500 nodes. MAE vs. the training epochs is also shown.

are used to calculate the MAE. Therefore, a plot of MAE as a function of training epoch is shown in subplot 3.

We conduct various tests to understand the factors which influence the LSTM-RNN's estimation performance and to further validate this estimation strategy. In the first test, we trained three LSTM-RNNs having different depths in time, i.e. where $\widetilde{N} = 250$, 500 and 1000 at an ambient temperature of 10°C. The estimated SOC and the error over time of these different LSTM-RNNs are shown in Figure 5.5 and their performance metrics are outlined in Table 5.4. To maintain an unbiased comparison

Table 5.4: SOC estimation accuracy of LSTM-RNN with various network depths in time

Network Depth in Time ($\widetilde{N}$)	MAE(%)	RMS(%)	STDDEV(%)	MAX(%)
250	1.303	1.670	1.043	5.380
500	0.672	0.863	0.293	3.015
1000	0.573	0.690	0.148	2.569

Networks trained on data recorded at ambient temperature of 10°C

between the network architectures tested in Table 5.4 training is stopped at 15000 epochs in each case. It is observed that the networks having larger depths in time which are exposed to more historical data perform better than those exposed to a smaller amount of historical data. However, the increase in estimation accuracy is not linearly proportional to depth in time since going from $\widetilde{N} = 250$ to $\widetilde{N} = 500$ reduces the MAE by about a half however, going from $\widetilde{N} = 500$ to $\widetilde{N} = 1000$ offers only a 15% reduction in MAE.

Another test is performed to measure the amount of training data needed to achieve good estimation accuracy. Therefore, instead of training the LSTM-RNN on a training dataset composed of 8 concatenated mixed drive cycles, as done to achieve the results in Table 5.3, Figure 5.4, Figure 5.5 and Table 5.4, we record the estimation accuracy of LSTM-RNNs trained on different numbers of mixed drive cycles. This is shown in Figure 5.6, where LSTM-RNNs are trained on one to nine mixed drive cycles and the MAE and max error measured over Test Case 1 are plotted in a bar graph. It is clear that the more training data the network is exposed to, the more accuracy is achieved, however after a certain amount of training data (6-8 drive cycles), we obtain diminishing returns. Interestingly, the LSTM-RNN is able to achieve a MAE of below 1% when training is conducted on 3 or more drive cycles. This can be very

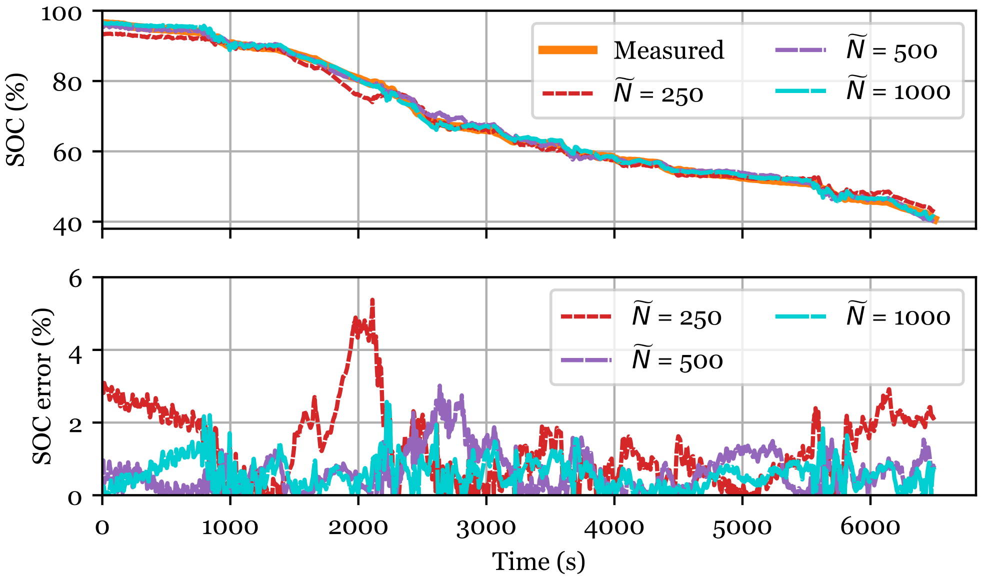


Figure 5.5: Performance of 3 different LSTM-RNNs having depth in time of $\widetilde{N}$ = 250, 500, 1000, tested on Test Case 1 recorded at an ambient temperature of 10°C. Each of the three networks' LSTM cell contains 500 computational nodes. Table 5.4 contains performance metric values.

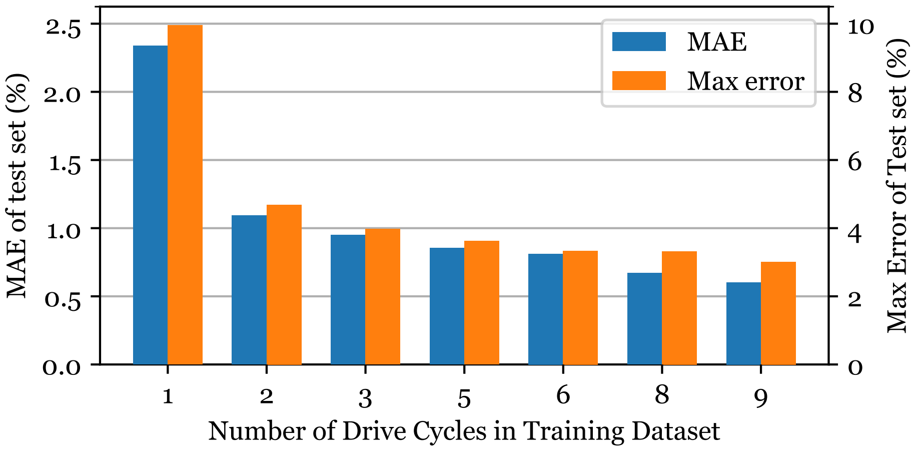


Figure 5.6: LSTM-RNN trained on different amounts of training data at an ambient temperature of 10°C and tested on Test case 1. Training is stopped at 15000 epochs. LSTM-RNN architecture: network depth in time is $\widetilde{N} = 500$ and LSTM contains 500 nodes.

advantageous since the LSTM-RNN can learn it's network parameters and achieve accurate SOC estimates after being exposed to relatively small amounts of training data.

Two additional tests are conducted to examine the LSTM-RNN's performance when either an incorrect initialization is given to the network or when the test drive cycle begins at different SOC levels. Giving an LSTM-RNN an incorrect initialization requires setting the hidden layer state at time step $k = 0$ to zero. This is the only way to test for the case of incorrect initialization since the input vector given to the LSTM-RNN at every time step includes $V(k)$, $I(k)$ as well as $T(k)$. SOC at time step $k-1$ or older are not used as feedback to the network. When correctly initialized, where $h_0 = h^*$, an LSTM-RNN achieves good performance with MAE $= 0.776\%$

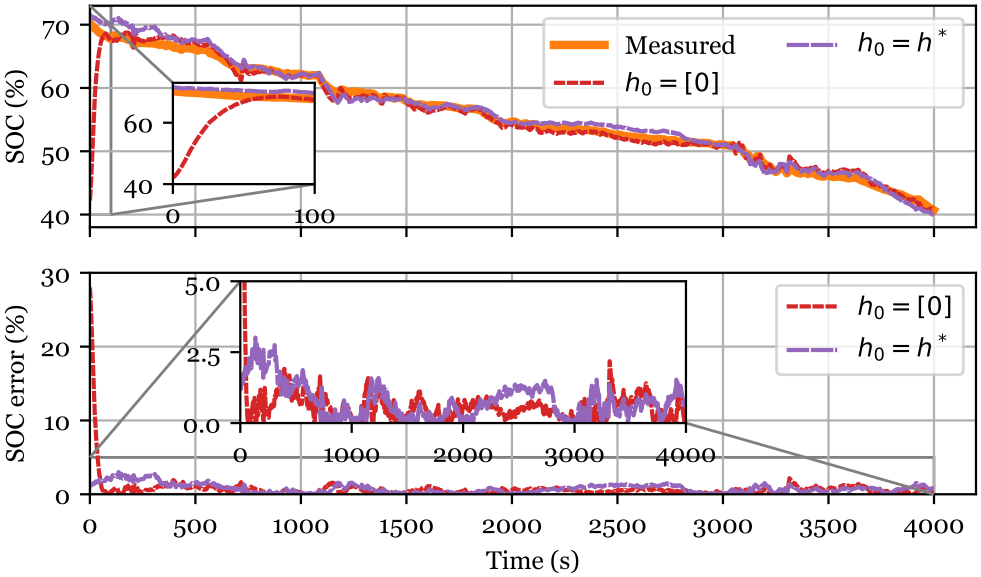


Figure 5.7: Performance of LSTM-RNN with initial state of charge of 70%. Two different cases are tested; one with correct initialization of the hidden layer ($h_0 = h^*$) and the other with an incorrect initialization of the hidden layer ($h_0 = 0$). Please refer to text for further discussion.

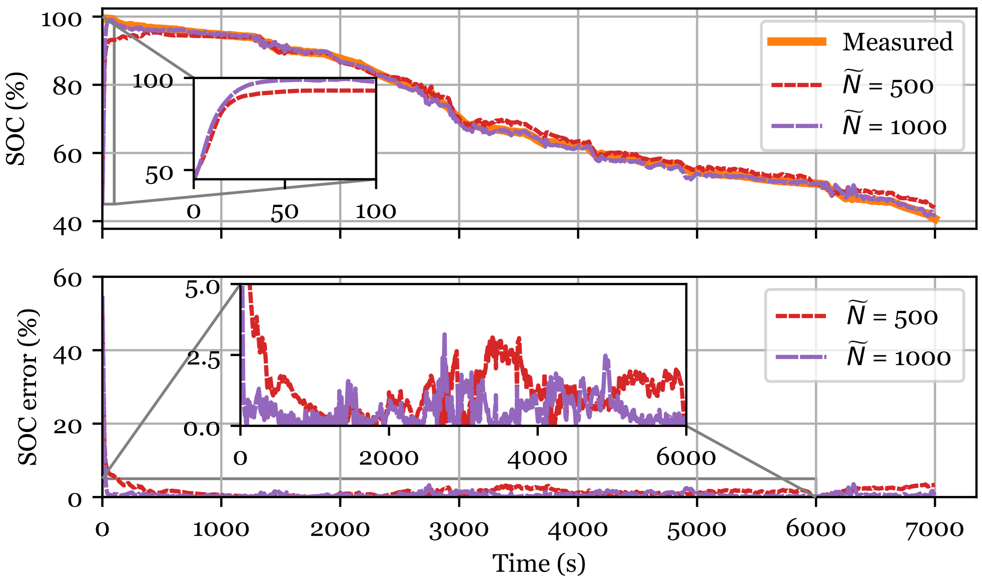


Figure 5.8: Performance of LSTM-RNN given an incorrect initialization and starting at SOC of 100%. Two networks are tested; one having depth in time of $\widetilde{N} = 500$ and the other having depth in time of $\widetilde{N} = 1000$. LSTM contains 500 nodes in both networks.

on Test Case 1 which begins at SOC = 70%, shown in Figure 5.7. When given an incorrect initialization, where $h_0 = 0$, the LSTM-RNN struggles to estimate SOC at the start of the drive cycle (SOC = 70%), where the error is about 27%, but quickly converges to a good state-of-charge estimate within the first 70 seconds of the drive cycle. Further validation is performed on this front by examining whether an LSTM-RNN that is deeper in time can converge quicker if given an incorrect initialization. This is shown in Figure 5.8 where two LSTM-RNNs, one having depth in time of $\widetilde{N} = 500$ and the other having $\widetilde{N} = 1000$, are tested with incorrect initialization on Test Case 1 beginning at a SOC of 100%. It's clear that the LSTM-RNN which has a depth in time of $\widetilde{N} = 1000$ seconds converges in less than 50 seconds which turns

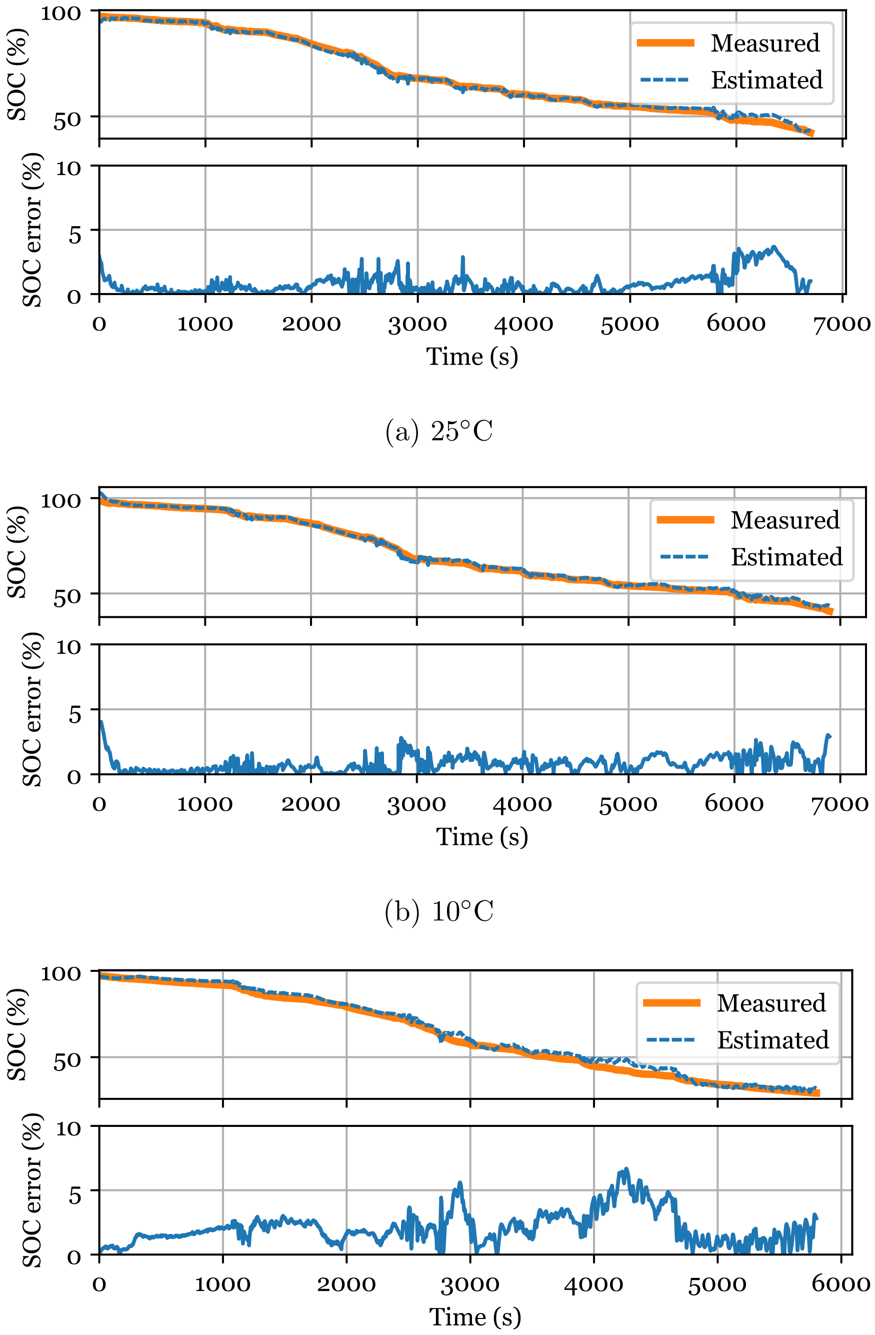
Table 5.5: SOC estimation accuracy of LSTM-RNN trained on multiple ambient temperature data

Ambient Temperature	MAE(%)	RMS(%)	STDDEV(%)	MAX(%)
25°C	0.774	1.110	0.796	3.692
10°C	0.782	0.995	0.616	4.047
0°C	2.088	2.444	1.270	6.687
Varying Temperature	1.606	2.038	1.256	5.815

out to be faster than the LSTM-RNN having $\widetilde{N} = 500$.

5.4 State-of-Charge Estimation at Variable Ambient Temperature

A LSTM-RNN is constructed to handle a larger training dataset which is composed of 27 drive cycles. These 27 drive cycles include three sets of 9 drive cycles; each set is recorded at 0°C, 10°C and 25°C. Another different mixed drive cycle, which is not a part of the training data, is used as a test case to validate the network's performance at each temperature. In particular, there are two goals that we desired to achieve within this second study. The first is to train the LSTM-RNN on datasets recorded at more than one ambient temperature such that one single LSTM-RNN can estimate SOC at different ambient temperature conditions. The second goal is to examine the LSTM-RNN's capability to interpolate its ability to estimate SOC at ambient temperatures different than the ones on which it was trained. The LSTM cell used in this study is unrolled for $\widetilde{N} = 1000$ time steps and the time required to train this network is about 12 hours.



(a) 25°C

(b) 10°C

(c) 0°C

Figure 5.9: Performance of a single LSTM-RNN on Test Case 1 performed at ambient temperatures of a) 25°C, b) 10°C and c) 0°C. LSTM-RNN architecture: network depth in time is $\widetilde{N} = 1000$ and LSTM contains 500 nodes.

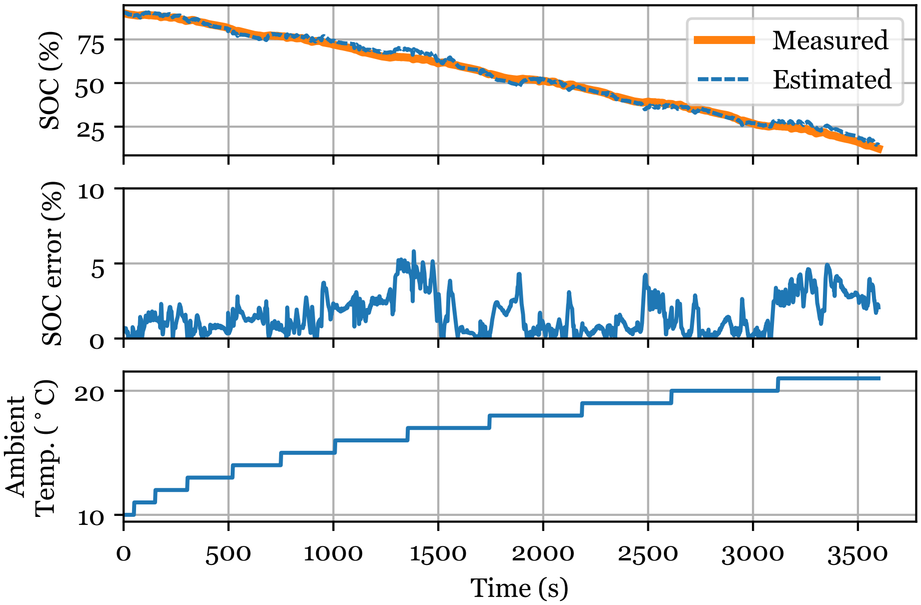


Figure 5.10: LSTM-RNN validated over UDDS test case with rising ambient temperature. From the top; measured SOC compared to estimated SOC, the error and the ambient temperature vs. time. LSTM-RNN architecture: depth in time is $\widetilde{N} = 1000$ and LSTM contains 500 nodes.

Table 5.6: Comparison of SOC estimation error for six studies

Method	Lowest Error	Temp.	Test Case	Li-ion Battery
AEKF w/ ANN battery model [129]	<3% RMS	20°C (ambient)	+/-1A charging pulse profile	1.2Ah
AUKF w/ extreme learning machine battery model [130]	<1.5% MAX	25°C (ambient)	0.52A 50% duty cycle pulse discharge	2.6Ah Samsung
AUKF w/ LSSVM battery model [131]	<2% MAE	25 to 42°C	Two drive cycles, +410 to -224A	70Ah Kokam
Fuzzy NN w/ genetic algorithm [189]	<0.9% APE	25°C (ambient)	Constant resistance discharge (≈13A)	10Ah MRL/ITRI
Radial Bias Function NN [190]	0.02% AAPE	Unspecified	Constant discharge rate (0.3C, 1C, 0.7C)	10Ah Lyno Power LYS347094S
LSTM-RNN (estimator in this thesis)	0.61% MAE 0.78% RMS 2.38% MAX (25°C)	0 to 25°C (ambient)	Dynamic drive cycles, +/- 18A, Range of ambient temperatures	2.9Ah Panasonic NCR18650PF

The estimation performance of this single LSTM-RNN is shown in Figure 5.9 where estimation performed at 25°C, 10°C and 0°C are plotted in a), b) and c), respectively. The MAE achieved at these three ambient temperatures is 0.774, 0.782 and 2.088. The MAE, RMS, STDDEV and MAX metrics achieved over all three ambient temperatures are outlined in Table 5.5. The performance is good and validates the LSTM-RNN's ability to encode the dynamics experienced by a Li-ion battery at various ambient temperatures into the parameters of a single network.

The single LSTM-RNN performed well for estimation on the validation test cases recorded at three different constant ambient temperatures however, battery-powered vehicles can undergo a change in ambient temperature of more than 10°C over the course of one day depending on the climate or the geographical location within which they operate. Hence, an interesting test is to examine its performance on a test case, not included in the training data, which is recorded at a changing ambient temperature. Therefore, the LSTM-RNN's performance over a test case where the ambient temperature in the thermal chamber is increased from 10°C to about 25°C is shown in Table 5.5 and in Figure 5.10. The MAE achieved on this test case is 1.606% and showcases the LSTM-RNN's ability to perform estimation at ambient temperatures between 10°C and 25°C to which it is not exposed during the training process. This can be a great advantage if training data is scarce.

In summary, this chapter highlights the elegance of the LSTM-RNN model for SOC estimation. The contribution on the battery modeling front is how the LSTM-RNN maps battery measurements like voltage, current and temperature directly to SOC. The LSTM-RNN offers competitive estimation performance when compared to other algorithms mentioned in literature which are shown in Table Table 5.6. The contribution on the parameterization front is how the training algorithm self-learns all the network parameters; freeing researchers from hand-engineering and parameterizing the models themselves. In traditional methods, only after the parameters are fitted to a battery model, and after the time-consuming covariance matrix in a Kalman filter is determined, for example, can the algorithm operate in the field. In the third contribution, the elegance of this machine learning technique is that it allows us to encode the characteristic behavior at different ambient temperatures within the network while maintaining good estimation accuracy at these different ambient temperatures.

Chapter 6

Feedforward and Deep Neural Networks for State-of-Charge Estimation

6.1 Trade-offs Between DNN and LSTM-RNN for SOC Estimation

This chapter explores the use of Feedforward Neural Networks and Deep Feedforward Neural Networks (DNN) for state-of-charge estimation of Li-ion batteries. Many works have previously attempted to perform SOC estimation while using a neural network, as discussed in Chapter 3, however most have required the use of a filter in order to achieve estimation errors below 2-3% [133; 131; 184]. In Chapter 5, it was demonstrated that SOC estimation can be performed without the need of any additional filters with an LSTM-RNN. In this chapter, the use of DNNs performing SOC estimation is examined. These can also be used with the need for filters and have their own advantages when used for SOC estimation.

In the previous chapter, Chapter 5, an LSTM-RNN is used to estimate SOC. The results showed competitive performance and the only inputs required by the LSTM-RNN are the voltage, the current and the temperature at the present time step. There is no requirement to feed antecedent values of any of these battery measurements. Therefore, for on-board applications, this means that there is no need to save antecedent values in order to perform an estimation calculation. However, as mentioned previously in Chapter 5, the training process of the LSTM-RNN is slower than the training process of a DNN. This results from the recurrent structure of the LSTM-RNN, illustrated in Figure 3.3, which needs to have the previous hidden states, h_{k-1}, computed in order to compute the next hidden state, h_k. Therefore, operations for this type of network cannot be parallelized as much as can be for other types of neural networks like DNNs. The latter networks are not time dependent, therefore the entire set of computations between the input layer and the output layer can be parallelized. Therefore, the training process for DNNs can be much faster as a result of its non-recurrent structure. However, the latter feature, while being an advantage during training, can be a drawback with respect to efficiency. As will be seen in this chapter, when considering the number of inputs required to achieve the same accuracy, the DNN will always require more than the LSTM-RNN. This is of course due to its lack of recurrent structure; since the DNN is not time dependent, it cannot hold the characteristic behavior of the battery in its memory, hence, that time dependent characteristic behavior must be supplied to the DNN through its inputs. Therefore, when using DNNs, antecedent values must be stored and saved during on-board operation in order to perform the calculation in the forward pass.

In the case of the LSTM-RNN, preprocessing is not used. This can be a result of how effective LSTM-RNNs are at representational learning. In the case of DNNs, it is observed that scaling is required. Therefore, this meant that the LSTM-RNN will not need to perform additional operations related to preprocessing of the raw data. Although some preprocessing is required and additional antecedent information is required for DNNs, they can offer ease of implementation for on-board applications. This of course results from the simplicity of a DNN layer versus the compositional intricacy of the LSTM-RNN layer.

To evaluate the SOC estimation performance of the DNN, a few different metrics are used, as was used in Chapter 5. These include the Mean Absolute Error (MAE), the Standard Deviation (STDDEV) and the Maximum Absolute Error (MAX).

6.2 Experimental Data

6.2.1 Drive Cycle Data for Training and Validation

A series of vehicle drive cycle and charging profiles were performed on a battery cell, and the resulting measured data is used to train and validate the DNNs. The nine different drive cycles used are given in Table 6.1. The cycles were chosen to cover a range of mean, RMS, and peak power values for the battery, as would be experienced in a vehicle application, and to provide enough data to train the neural network. The tests were also performed at a range of temperatures, down to -20°C where nonlinear resistance, battery self-heating, and diffusion effects make SOC estimation more tedious.

Four of the cycles, which are the US06, HWFET, UDDS, and LA92, are standard drive cycles, and one drive cycle, called the NN, was designed for this work and has additional dynamics useful for training the DNNs. The other four cycles, cycles 1 through 4, are composed of a mixture of the four standard drive cycles performed in a random, repeating order. The battery power command for the tests was calculated for a large light duty passenger vehicle with a 35kWh battery pack, and scaled for a single battery cell.

The DNNs are validated using the US06 cycle, an aggressive and highly dynamic profile, and the HWFET cycle, a profile with moderate dynamics. The data for the remaining seven tests are used to train the neural network, as is indicated in Table 6.1.

The test procedure used is as follows: (1) set the thermal chamber temperature to 25°C, (2) wait ten minutes or until the measured battery temperature is greater than 10°C, (3) perform a full charge, (4) set the chamber to the current test temperature,

Table 6.1: Tested Drive Cycles and Corresponding Battery Cell Power Characteristics

Test	Use	Mean Power(W)	RMS Power(W)	Peak Power(W)
Cycle 1	Training	3	7	35
Cycle 2	Training	3	7	35
Cycle 3	Training	3	7	35
Cycle 4	Training	3	7	35
US06	Validation	6	13	34
HWFET	Validation	5	6	19
UDDS	Training	2	4	19
LA92	Training	2	7	35
NN	Training	3	10	39

(5) wait for the battery temperature to stabilize, and (6) perform the drive cycle. The drive cycle is repeated until the battery voltage reaches 2.5V for the 25 and 10°C tests. For the lower temperature tests, the battery resistance is much higher, causing the terminal voltage to hit 2.5V periodically throughout the test. To accommodate the higher resistance, an amp-hour cut off 2.32Ah at 0°C, 2.03Ah at -10°C, and 1.74Ah at -20°C is used instead. The battery tester still has a lower voltage limit of 2.5V, but the test is not terminated when this voltage is hit. The tester reduces current to prevent voltage from falling below the limit. The battery may hit the lower voltage limit several times when at higher currents and low SOC. The nine drive cycle tests were performed in the order given in Table 6.1, first at 25°C, and then at 10, 0, -10, and -20°C. Following those tests, the nine drive cycles were also performed at a variable ambient temperature where the temperature was increased from 10°C to about 25°C. These varied tests are used to evaluate the SOC estimation algorithm for cases when the battery is warming up throughout the drive.

The tested battery cell is a Panasonic NCR18650PF, nickel cobalt aluminum

Table 6.2: Panasonic 18650PF Cell Parameters

Nominal Open Circuit Voltage	3.6V
Capacity	Min. 2.75 Ah / Typ. 2.9 Ah
Min / Max Voltage	2.5V / 4.2V
Mass / Energy Storage	48g / 9.9Wh
Minimum Charging Temperature	10°C
Cycles to 80% Capacity	500 (100% DOD, 25°C)

Table 6.3: Test Equipment Specifications

Cycler Manufacturer	Digatron Firing Circuits
Test Channel Used	25A, 0-18V channel
Voltage / Current Accuracy	+/- 0.1% Full Scale
Data Acquisition Rate Used	10Hz
Thermal Chamber	Cincinatti Sub Zero ZP-8
Size	8 cu. Ft.
Accuracy	+/-0.5°C

(NCA) chemistry Li-ion battery, with a nominal capacity of 2.9Ah and other specifications as described in Table 6.2 [185; 186]. The battery cell is tested with a 25 amp, 0 to 18V rated channel of a Digatron Firing Circuits Universal Battery Tester, and placed in a Cincinnati Sub Zero thermal chamber, as is described in Table 6.3. The battery tester is very accurate; rated for less than 0.1% error, which is important because the measured battery amp-hours is used to calculate the reference or ground truth SOC value. The maximum error of 0.1% translates to a 25mA current error, which would result in a maximum of 25mAh of error for a one hour drive cycle test; which can be considered to have a negligeable effect on the results. The battery system and instrumentation is illustrated in Figure 6.1, which shows that the voltage sensing leads are connected directly to the battery terminals and that a thermocouple is connected to the center of the case.

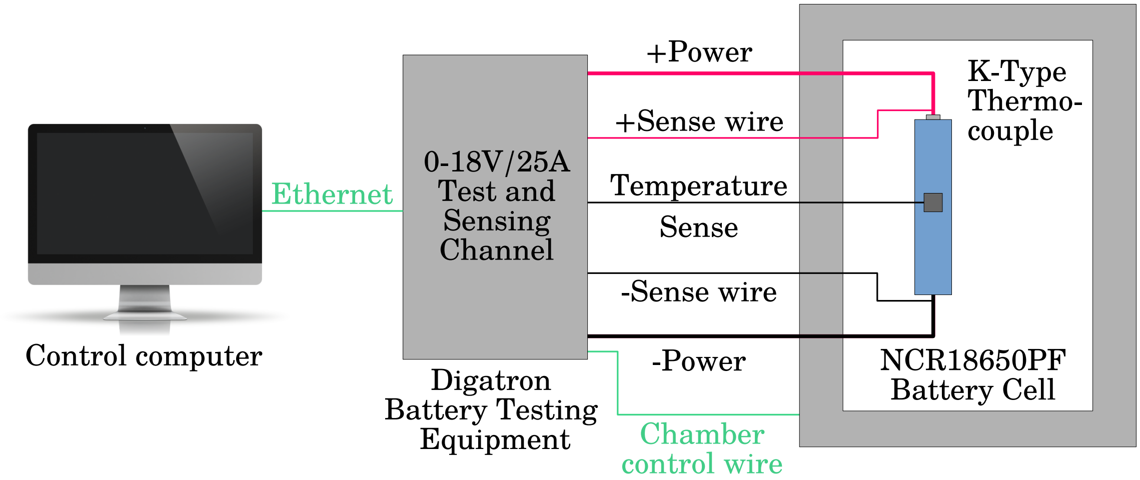


Figure 6.1: Schematic of the battery test bench and data logging system.

6.2.2 Data Augmentation for Robust Deep Neural Networks

To robustify the deep neural network, the training data is augmented by injecting white Gaussian noise into the battery measurement signals. In particular, Gaussian noise with 0 mean and a standard deviation of 2-4% is injected into the voltage, current and temperature measurements. In addition, to robustify against offsets and gains inherent in battery measurement devices, an offset is applied to all measurement signals and a gain is applied to the current measurement. Specifically, an offset of up to ±150mA and a gain of up to ±3% is applied to the current measurements, an offset of up to ±5mV is applied to the voltage measurement and an offset of up to ±5°C is applied to the temperature measurement. New versions of the initial recorded training data is created with different noise, offset and gain levels. We create up to 20 new versions of the training data.

All the results showcased in this work are based on DNNs trained on data obtained from Panasonic 18650PF cells. Nevertheless, the DNNs discussed in this work can be trained on any other type of battery cell. The network architectures will remain unchanged if used to perform SOC estimation on another battery however, the network might need to be retrained or better yet transfer learning can be used to simply retrain the last one or last few layers of the network. This would be less time-consuming than re-parameterizing a model for an alternative battery cell as would be typically performed for traditional models.

The following two subsections examine the SOC estimation accuracy of the DNN at fixed ambient temperature and, thereafter, at variable ambient temperatures.

6.3 State-of-charge Estimation Results

As mentioned earlier, the vector of inputs fed into the DNNs is given by $\psi(t) = [V(t), T(t), I_{avg}(t), V_{avg}(t)]$ where $V(t)$, $T(t)$, $I_{avg}(t)$ and $V_{avg}(t)$ represent the voltage, temperature, average current and average voltage of the battery at time step t. The output of the DNN is the estimated SOC at t. The drive cycles used for training and validation are recorded at a sampling frequency of 1Hz and are between 4000 and 13000 seconds long.

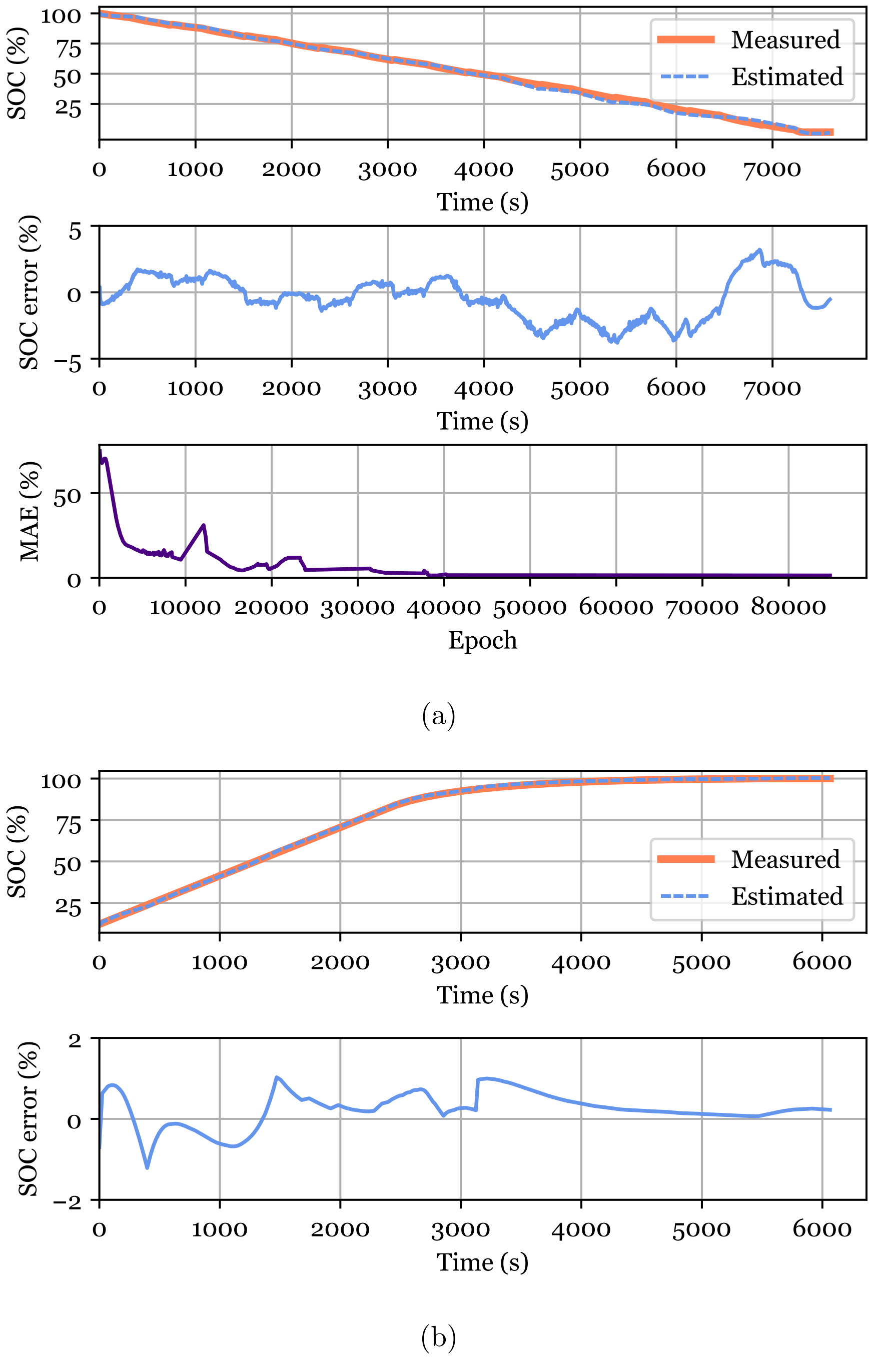
6.3.1 SOC Estimation at Constant Ambient Temperature

In this section, the DNN is trained on up to seven complete discharge datasets which are augmented as described in subsection 3.3. To verify the DNN's performance in both fast and slow dynamics, validation is conducted on the US06 and HWFET datasets, respectively. These latter datasets as well as all other datasets used to obtain performance results are only used during validation and never during training.

Regenerative breaking is also allowed for the discharge datasets recorded at the higher temperatures however not at lower ambient temperatures since the battery is not rated for charging at temperatures lower than 10°C. When regenerative braking occurs at the higher temperatures, the charging current is seen to spike up to 8 A. This is useful in order to ascertain the DNN's performance at higher charging currents even if momentary in nature. These higher charging spikes cannot be maintained for longer than a few seconds since this would also exceed the peak charge current rating of the battery. To evaluate the DNN's performance in charging scenarios for longer periods of time, the DNN is applied on a charging validation dataset. The charge rate used in this dataset is 1C. There is interest in using a charge rate that is higher than this however 1C is already twice the recommended charge rate and selecting an even higher rate is excessive and potentially dangerous.

Good performance is observed both in slow and fast dynamics as well as in the charging validation dataset, as can be seen in Figure 6.2 and in Table 6.4, where an MAE of 1.35%, 1.85% and 0.39% is achieved over the HWFET, US06 and charging datasets, respectively. The DNN used to obtain these results is composed of 3 layers where the first two layers each contains 4 neurons and the last layer contains 1 neuron. The network is trained for up to 85000 epochs which culminates to 5 hours of training time on the GPU. The MAE calculated over the HWFET validation dataset is plotted as a function of the training epochs and is also shown in Figure 6.2(a).

Different tests are performed to examine effects of various structural aspects of the DNN on SOC estimation accuracy. The first of such tests evaluates the influence of the λ parameter on estimation accuracy. The λ parameter, used when computing the compositional inputs which are the average voltage and current is explained in



(a)

(b)

Figure 6.2: (a) From top to bottom; DNN estimation accuracy, estimation error over the HWFET discharge validation dataset recorded at 25°C and the mean absolute error as a function of training epochs. (b) Estimation accuracy and estimation error over validation charging profile also recorded at 25°C. DNN is composed of 3 layers and 4 neurons in each layer.

Table 6.4: SOC estimation accuracy of deep neural network trained on fixed ambient temperature data of 25°C during validation

Validation Dataset	MAE(%)	STDDEV(%)	MAX(%)
HWFET Dataset	1.35	0.94	3.80
US06 Dataset	1.85	1.20	5.14
Charging Dataset	0.39	0.26	1.21

DNN architecture: $3L$ and $4N$/layer

Table 6.5: SOC estimation accuracy using different averaging parameter λ in training data

Validation Dataset	MAE(%)	STDDEV(%)	MAX(%)
HWFET with $\lambda = 100$	0.96	0.76	3.91
HWFET with $\lambda = 400$	0.61	0.49	2.38
US06 with $\lambda = 100$	1.16	0.86	4.54
US06 with $\lambda = 400$	0.84	0.61	3.14

All validation results are obtained on 25°C datasets
DNN architecture: $3L$ and $4N$/layer

more detail in Section 3.2.1. In effect, this would examine how exposure to increased amounts of historical data could increase estimation accuracy of the DNN. A DNN is trained on input data with $\lambda = 100$ and another is trained on input data with $\lambda = 400$. The results from these two networks are shown in Figure 6.3a) and summarized in Table 6.5. Exposing the DNN to 400 historical data points as opposed to 100 offers good performance gains with a reduction in the MAE and MAX of approximately 30% and 30-40%, respectively. To ensure an unbiased comparison, training is stopped at 160,000 epochs.

The battery measurements performed in the lab are obtained in isolated and controlled environments to ensure good quality data. However, in the real world, battery measurements, like voltage, current and temperature measurements, can be of variable quality. A DNN can be taught to handle such noisy environments at training

Table 6.6: SOC estimation accuracy of DNN trained over augmented and unaugmented data

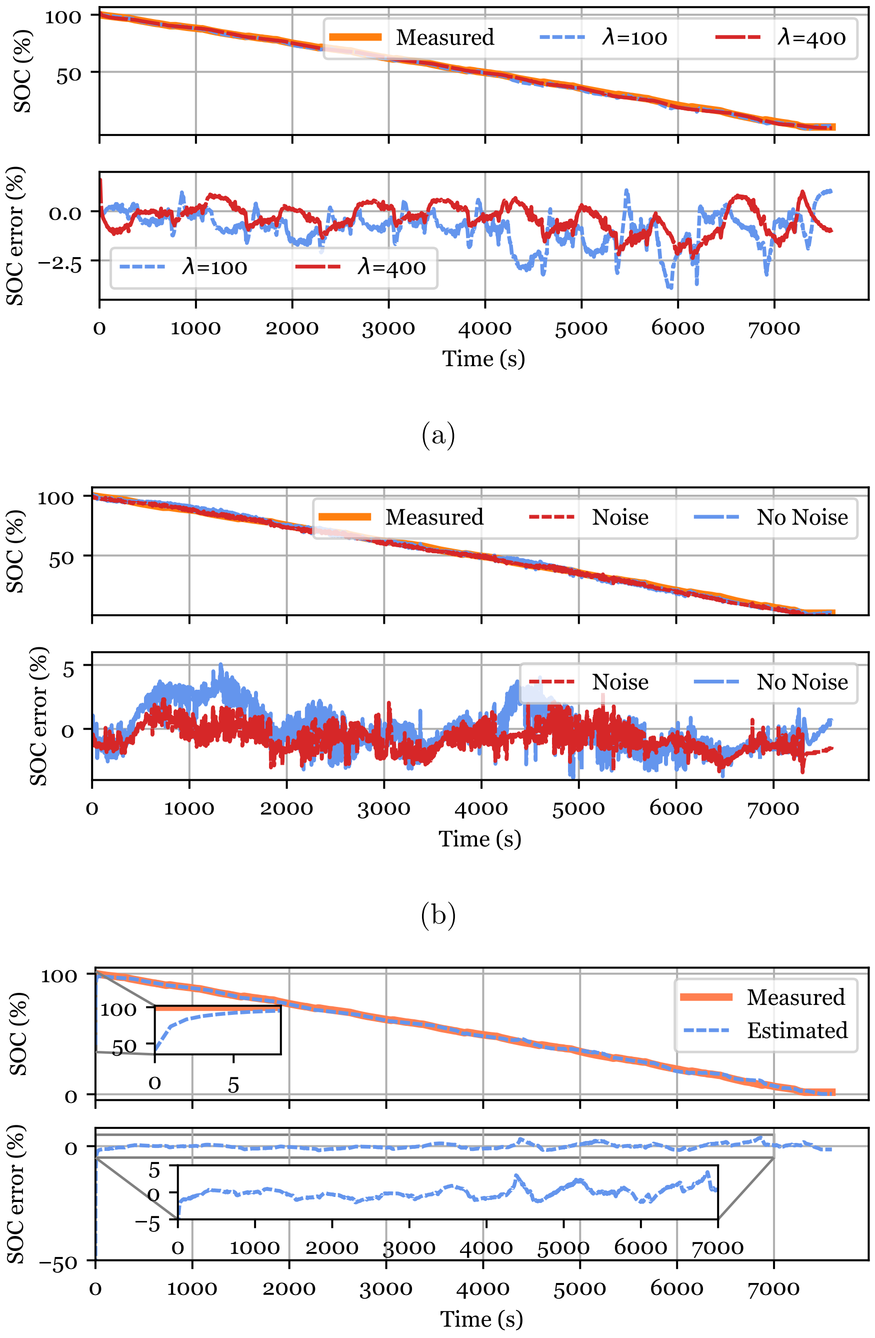
Training Data Augmentation	Validation Dataset	MAE(%)	STDDEV(%)	MAX(%)
No	HWFET	1.35	0.94	5.04
Yes	HWFET	1.06	0.49	3.41
No	US06	2.67	2.08	15.14
Yes	US06	1.59	1.23	7.14

All validation results are obtained on 25°C datasets
DNN architecture: $3L$ and $4N$/layer

time. As a result, a comparison showing the estimation performance achieved on the augmented training dataset versus the unaugmented training dataset is shown in Table 6.6 and in Figure 6.3b). From the latter table and figure, it becomes evident that using the augmented dataset for training achieves a significant reduction in MAE and MAX of 21-41% and 32-43%, respectively. By injecting Gaussian noise, offsets and gains on the measurement devices, not only is the accuracy of the DNN improved but the network is robustified for real world application.

Another important test is to verify the DNN's performance when it is initialized incorrectly. To that end, instead of providing the correct measured initial battery voltage of 4.2 V to the network, an incorrect initial value of 3.6 V is given. This is shown in Figure 6.3c), where the DNN struggles to output a good estimate of SOC at the beginning of the dataset, where the error is over 50%, but quickly converges to good SOC estimation within the first 10 seconds of the dataset.

A large fraction of the performance results discussed above are obtained on DNNs with 3 layers; where the first two layers contain 4 neurons and the last fully-connected layer contains 1 neuron. Interestingly, this corresponds to 36 network weights which is relatively small in size and can be flashed on any embedded processor when operating

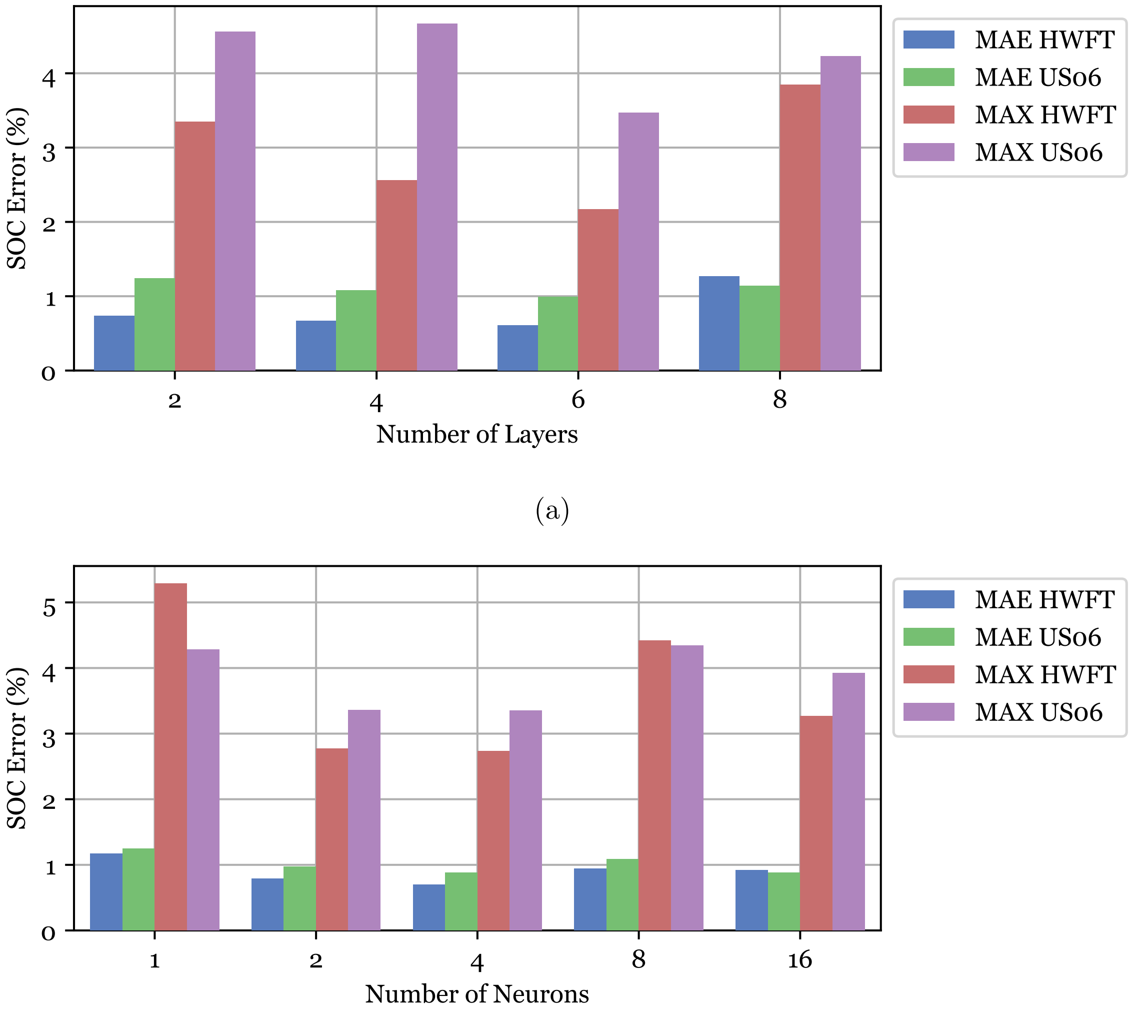


(a)

(b)

(c)

Figure 6.3: (a) Performance of DNN with λ=100 and with λ=400. (b) Performance of DNN trained over augmented training data and over unaugmented training data. (c) Performance over incorrectly initialized network. DNNs have 3 - 8 layers and 4 - 8 neurons per layer. Validation is performed over 25°C HWFET dataset.



(a)

(b)

Figure 6.4: (a) Estimation accuracy measured during validation versus number of layers in DNN. The number of neurons per layer is fixed to 4 neurons. (b) Estimation accuracy measured during validation versus number of neurons. The number of layers in the DNN is fixed to 6 layers. All tests are performed over validation datasets recorded at 25°C and $\lambda = 400$.

in real time. To determine an optimal network architecture, the estimation accuracy is evaluated on networks with different numbers of layers and neurons. This is shown in Figure 6.4a) and in Figure 6.4b), respectively. For each test case shown in Figure 6.4, $\lambda = 400$ and training is stopped at 100,000 epochs to ensure an unbiased comparison. It is clear that a deeper network can achieve better estimation accuracy as going from 2 layers to 4 offers a 10% and a 24% reduction in MAE and MAX error, respectively. In the same vein, going from 4 to 6 layers reduces the MAE and MAX error by 9% and 15%, respectively, however, increasing the depth up to 8 layers achieves no reduction in MAE or MAX. This could very well be a result of the gradients, defined in Equation (3.14), becoming much smaller as more layers are added to the network which leads to vanishing gradients and the inability to update the network weights during training. With respect to the number of neurons in the network, Figure 6.4b) indicates that going from 1 neuron to 4 per layer offers a reduction in MAE and MAX of 32-35% and 16-45%, respectively. However, 8 or more neurons offer no additional reduction in MAE or MAX error. This could very well be as a result of the having a fixed number of neurons in each layer as opposed to an increased number of neurons which allows the network to learn more features at each layer.

6.3.2 Computational Efficiency and Real Time Operation

The computational complexity of the DNN as a function of neurons and layers was presented in Section 4.2. Nevertheless, it is also essential to compare the computational efficiency of the DNN with respect to other competing algorithms. To this end, the computational time of the DNN is compared to that of an EKF approach used in [128].

Table 6.7: Computational time of a DNN and of an extended Kalman filter

SOC Estimation Technique	Computational Speed (seconds)
DNN (2 layers, 256 neurons)	0.07
Combined Model + EKF	0.66

The goal of this test was to measure the time required by these two techniques in order to compute through 13000 data points. This is performed in MATLAB on a machine with an Intel Xeon E5630 processor having 4 cores with a base clock frequency of 2.53 GHz and with 32 Gb of RAM. The results are shown in Table 6.7 and are based on an average of 20 separate executions of the two algorithms over the same dataset. Therefore on average, the DNN is almost an order of magnitude faster than the EKF. This can be attributed to the fact that the EKF is required to linearize around each operating point while the DNN mainly performs a series of matrix multiplications to achieve the desired output which is an inherently much simpler series of operations.

Moreover, the operation of the DNN is validated in real time. This allows us to validate it's operation in real world noisy environments with noisy measurement devices and allows us to validate it's robustness against such noisy signals. To that end two different current measurement devices are used to measure the current from the battery cell. A highly accurate measurement device with less than 25mA per hour of error is used to perform Coulomb counting which provides the ideal SOC value. The second current measurement device is the LEM DHAB sensor, which a standard automotive grade off-the-shelf current measurement device which an offset error of 150 mA - 350 mA depending on the ambient temperature and a sensitivity error which can span 0.5% - 1.5%.

In Figure 6.5, it is shown that an off-the-shelf standard current sensor performing

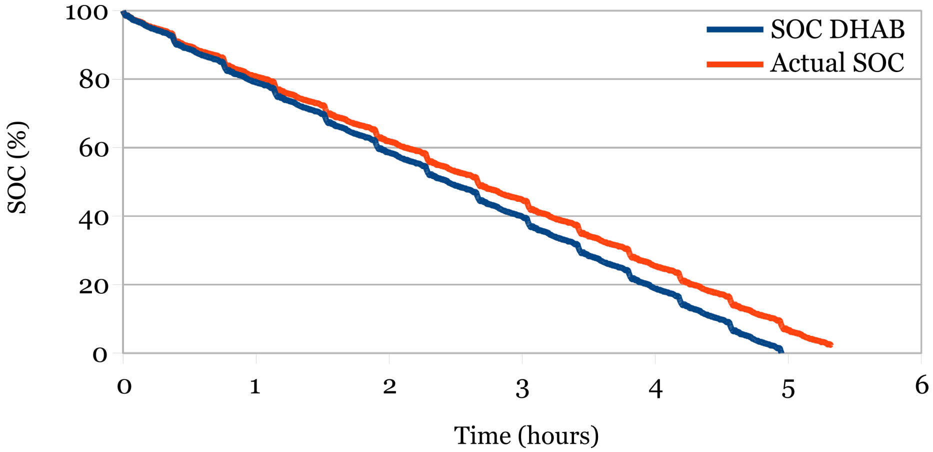


Figure 6.5: Actual or ideal SOC computed using Coulomb counting from high accuracy current sensor and SOC computed using automotive grade LEM DHAB current sensor. Errors from DHAB sensor aggregate over time such that SOC estimate diverges away from the ideal or actual SOC.

simple Coulomb counting does not possess the accuracy to issue a good estimate of SOC. This reinforces the need for SOC estimation strategies with good representational power.

After being trained, the DNN algorithms are exported for use in real time in a hardware-in-the-loop setup. This consisted of the same experimental setup as shown in Figure 6.1, although the LEM DHAB current sensor is now also connected to the battery cell. Therefore, the accurate sensor as well as the LEM DHAB sensor are both connected to the battery. The current measurement from the LEM DHAB sensor is provided as an input to the DNN algorithm. The DNN issues an estimate of SOC in real time and this is then compared to the ideal or actual SOC estimate provided by the accurate current sensor.

The DNN used is a simple 2 layer neural network with 256 neurons in the hidden layer. The performance in real time of this algorithm is shown in Figure 6.6 where

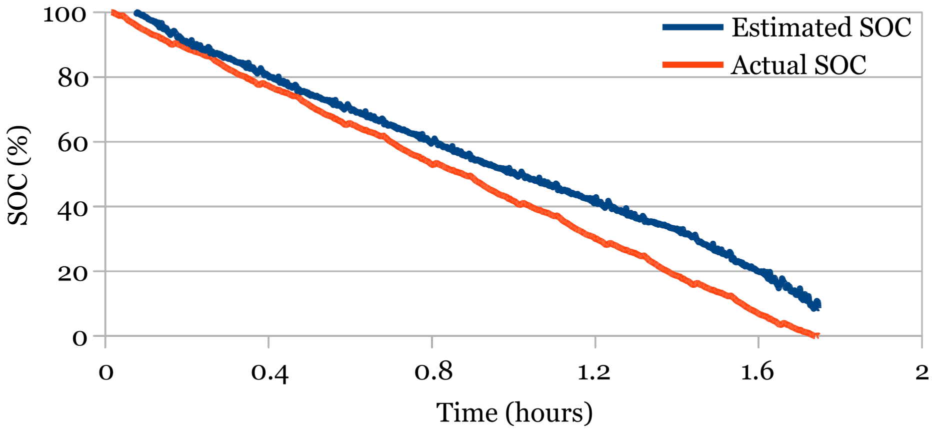
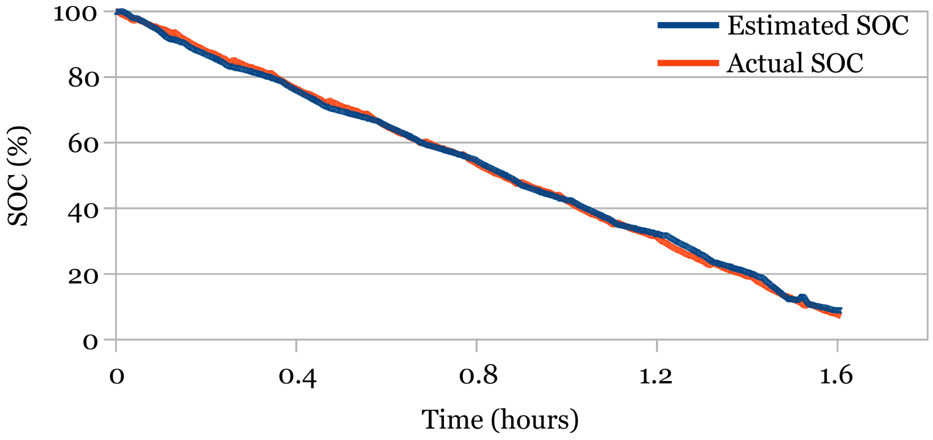


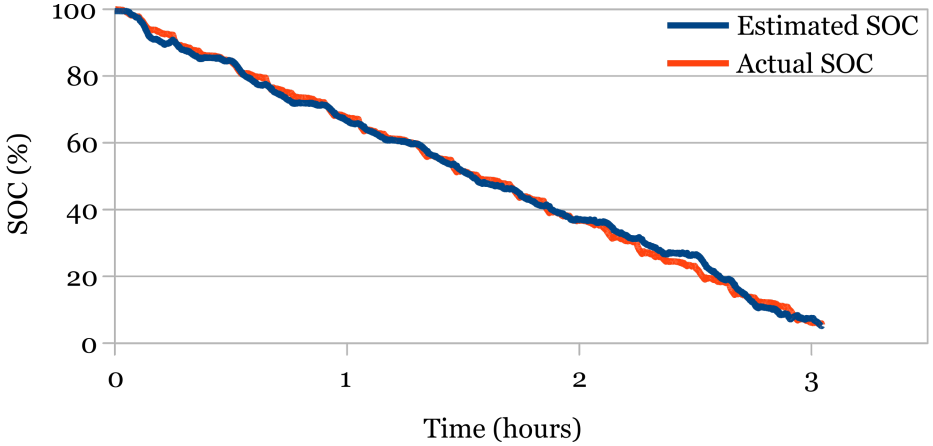
Figure 6.6: Actual or ideal SOC computed using Coulomb counting from high accuracy current sensor and SOC computed using automotive grade LEM DHAB current sensor for HWFET drive cycle. Low performance due to the DNN being trained over data from a battery that had seen significant aging in comparison to the battery used in this real time test.

the MAE = 7.55%. This algorithm is not very accurate in real time. The low performance can be attributed to the fact that the data used during training is that of an aged battery while the battery cell used during the real time testing was that of a new battery. Therefore, after training the DNN on data from a new battery, the performance of the DNN, observed in Figure 6.7 is much better and is very similar to the results achieved above in the fixed ambient temperature results of Section 6.3.1.

Using the MATLAB FLOPs counter, the DNN algorithm which includes the data processing phase, shows a FLOPs count of under 3000. This is well below most automotive grade embedded processors which can handle a computing speed ranging from megaflops to gigaflops.



(a) MAE = 1.01% over HWFET drive cycle



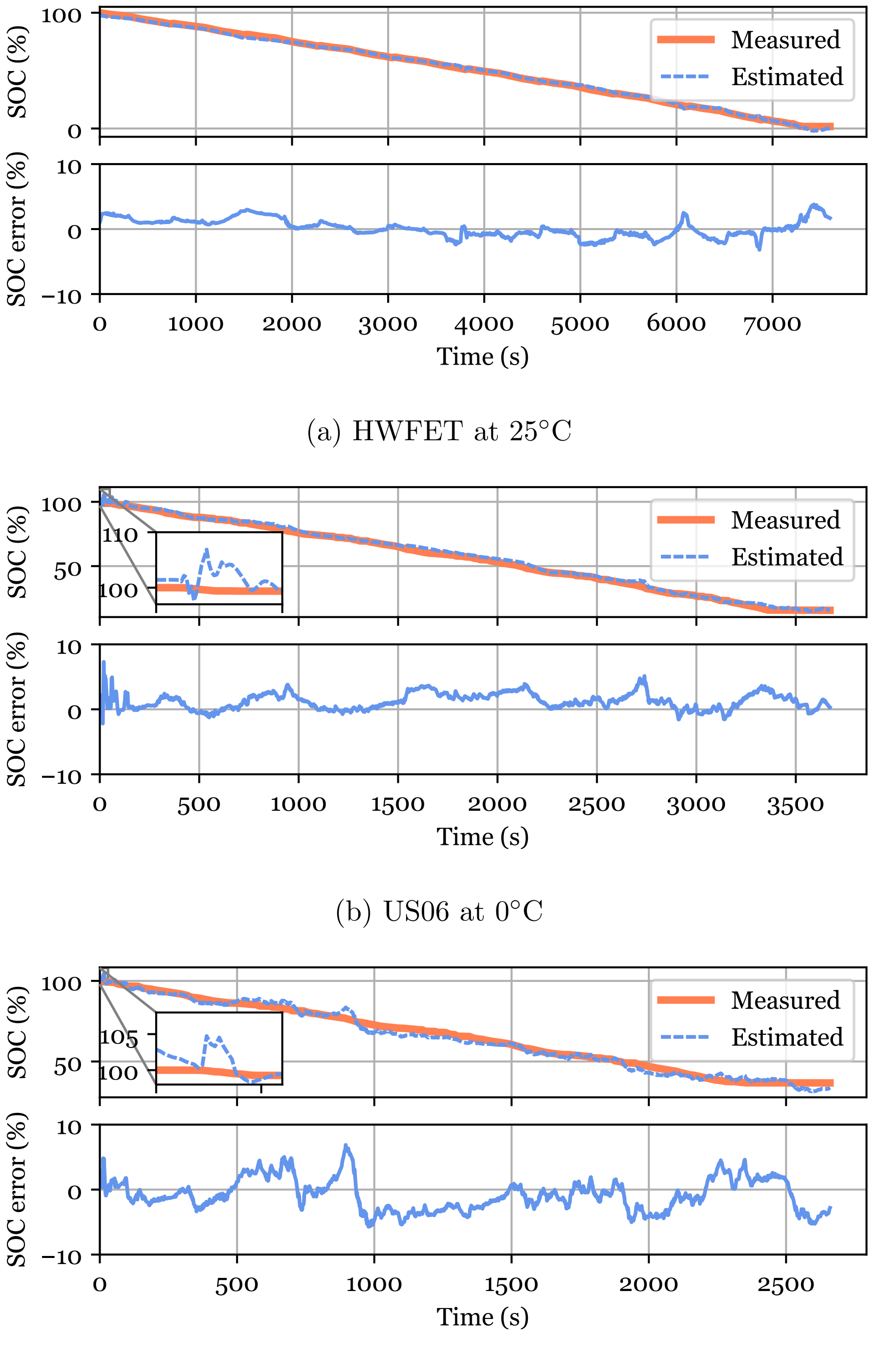
(b) MAE = 1.23% over LA92 drive cycle

Figure 6.7: SOC computed using automotive grade LEM DHAB current sensor over (a) HWFET drive cycle and (b) LA92 drive cycle compared to actual or ideal SOC computed using Coulomb counting from high accuracy current sensor. Real time testing performed at ambient temperature of 25°C.

6.3.3 SOC Estimation at Variable Ambient Temperature

In this section, two objectives are pursued. The first is to train the DNN on data taken at various ambient temperatures such that a single DNN can estimate SOC at different ambient temperatures. The seven training datasets recorded at the 5 different ambient temperatures (25°C, 10°C, 0°C, -10°C and -20°C), mentioned in Table 6.1, are used for training. These datasets are augmented, as described in section III, to increase the robustness of the DNN which raises the training dataset size to 20 times the initial size. Two validation datasets from each of the 5 ambient temperatures are then used to evaluate the DNN's performance. The second objective is to investigate the DNN's capability to interpolate it's ability to estimate SOC at ambient temperatures other than the ones one which it is trained, which is also performed with the LSTM-RNN in Chapter 5. The DNN used in this section is 4 layers deep and has 8, 16, 32 and 1 neurons in the respective layers. The training data is composed of the battery voltage, temperature, average voltage and average current, as is used in the previous section, and the averaging parameter, λ is set to 400. The time required to train this DNN is about 40 hours.

Figure 6.8 showcases the DNN's estimation performance over one validation dataset from 3 different ambient temperatures; 25°C, 0°C and -20°C. In addition, Figure 6.9 shows a bar graph outlining the DNN's performance over the two validation datasets for every one of the five ambient temperatures. Typically, higher estimation error is expected for the lower ambient temperatures since the disparity between the measured surface temperature of the battery and its internal temperature increases for lower ambient temperatures. Though, interestingly, the lowest MAE is achieved over a 0°C dataset at 0.91%. This can be attributed to the added



(a) HWFET at 25°C

(b) US06 at 0°C

(c) US06 at -20°C

Figure 6.8: (a) Estimation performance of one DNN at 3 different ambient temperatures. DNN is composed of 4 layer, where 8, 16, 32 and 1 neurons are used in the respective layers.

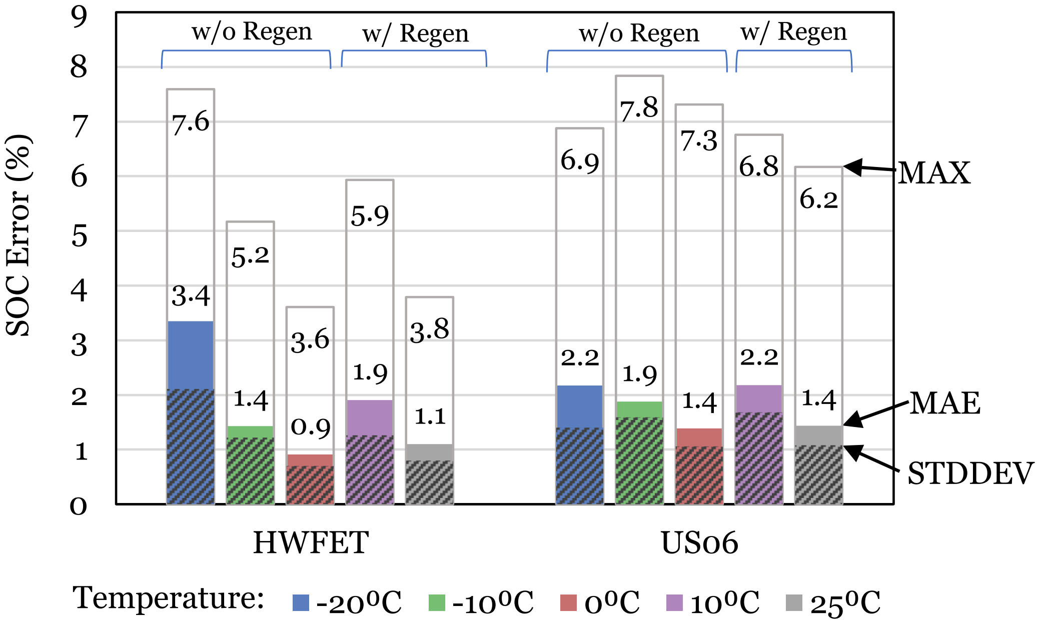


Figure 6.9: SOC estimation accuracy of DNN trained on varying ambient temperature data during validation.

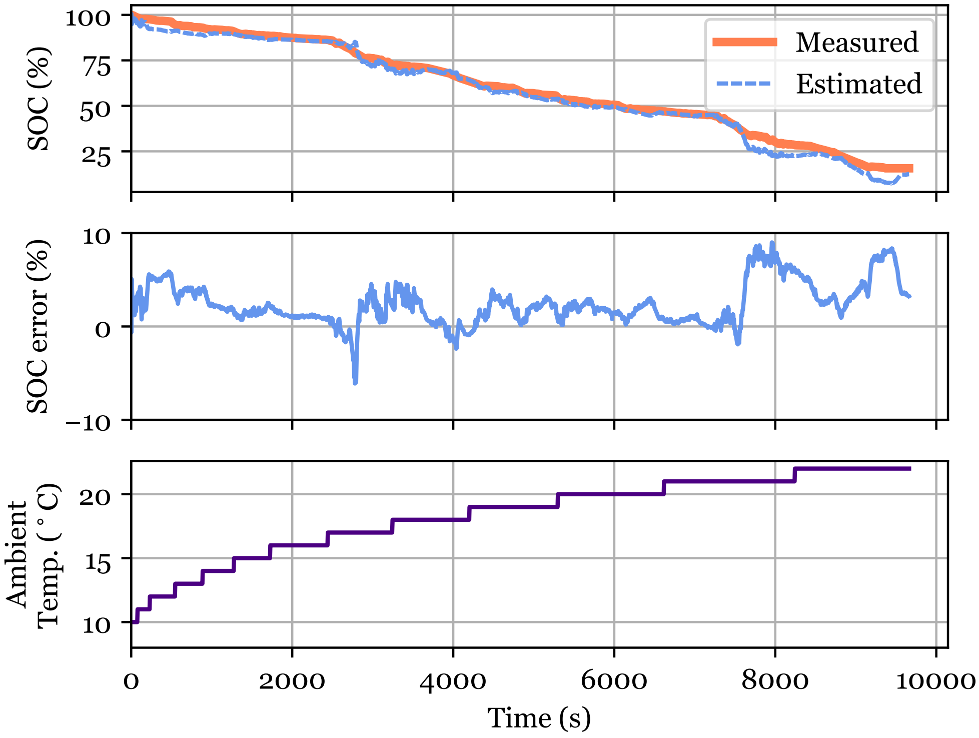


Figure 6.10: Estimation performance of DNN on validation set recorded at an increasing ambient temperature. From top to bottom, the plots shown are of the estimated SOC compared to the ground-truth SOC, the resulting errors at each time step and the ambient temperature. The validation dataset is composed of a mixture of HWFET, US06, LA92 and the NN drive cycle.

regenerative braking at the higher ambient temperatures which can add uncertainty. However, the MAE achieved at 25°C is still quite low at 1.44%.

In most cases, the MAE over the HWFET datasets which are less dynamic in nature are lower than those obtained over the US06 datasets which are much more dynamic in nature. However, this phenomenon is not exhibited for the case of −20°C. For the −20°C ambient temperature, the MAE over the HWFET dataset is higher. This is likely because the average battery temperature for the HWFET test is −14°C, which is much lower than the average temperature of −6.7°C for the US06 test. While the US06 temperature is greater for all of the test cases, the temperature difference has more significance at these lower temperatures where battery resistance increases dramatically. The maximum error is not entirely representative of the performance of the DNN since a few outliers over thousands of estimated values do not represent the overall performance of the network, however MAX is provided in the results for completeness.

The DNN had good estimation performance when validated over constant ambient temperatures. However, depending on the geographical location within which a battery-powered vehicle may operate, a variation of 5 to 10°C in ambient temperature is possible over the course of one day. Hence, a worthwhile exercise would be to validate the DNN over a dataset which has an ambient temperature that changes over time. Therefore, the DNN's estimation performance is tested over a validation cycle which increases from 10°C to 25°C over the course of the dataset, which is shown in Figure 6.10. As can be seen, the DNN performs well over this validation dataset, maintaining good estimation accuracy even at ambient temperatures lying between 10°C and 25°C, on which the network is not trained. This interpolating ability can be a great advantage when training data is scarce.

Table 6.8: Comparison of SOC estimation error for eight studies

Method	Lowest Error	Temp.	Test Case	Li-ion Battery
Model Adaptive-Improved EKF [27]	<1.5% MAE	25°C	Two drive cycles, +2 to -1A	1.1Ah A123 APR18650m1
AEKF w/ ANN battery model [129]	<3% RMS	20°C (ambient)	+/-1A charging pulse profile	1.2Ah
AUKF w/ extreme learning machine battery model [130]	<1.5% MAX	25°C (ambient)	0.52A 50% duty cycle pulse discharge	2.6Ah Samsung
AUKF w/ LSSVM battery model [131]	<2% MAE	25 to 42°C	Two drive cycles, +410 to -224A	70Ah Kokam
Fuzzy NN w/ genetic algorithm [189]	<0.9% APE	25°C (ambient)	Constant resistance discharge (≈13A)	10Ah MRL/ITRI
Radial Bias Function NN [190]	0.02% AAPE	Unspecified	Constant discharge rate (0.3C, 1C, 0.7C)	10Ah Lyno Power LYS347094S
LSTM-RNN (estimator in this thesis)	0.6% MAE 0.7% RMS 2.6% MAX (25°C)	0 to 25°C (ambient)	Dynamic drive cycles, +/- 18A, Range of ambient temperatures	2.9Ah Panasonic NCR18650PF
DNN (estimator in this thesis)	0.6% MAE 0.8% RMS 2.4% MAX (25°C)	-20 to 25°C (ambient)	Dynamic drive cycles, +/- 18A, Range of ambient temperatures	2.9Ah Panasonic NCR18650PF

In summary, the DNN algorithms have also proven to perform well with respect to SOC estimation. Firstly, on the modeling and estimation front, the DNNs used in this chapter map the measured battery signals like voltage, current and temperature directly to SOC and achieves competitive estimation performance with MAEs below 1%. The DNN's SOC results performance are compared to other techniques discussed in different studies as well as to the those from the LSTM-RNN in Chapter 5 in Table 6.8. Secondly, on the parametrization front, the DNN self-learns all its weights, eliminating the need to hand-engineer and parametrize traditional models, which is a very time-consuming and costly process. Thirdly, on the inference front, once trained, a DNN can operate online with relatively low computational time. As seen above, they can offer competitive SOC estimation results with as few as 36 network weights. In addition, the DNNs presented in this chapter are robustified against measurement offsets, gains and noise such that they can retain great estimation performance regardless of the imperfections found in a vehicle's measurement devices. Whether LSTM-RNNs or DNNs are used for SOC estimation, they both offer different advantages. One is more efficient during the learning process and allows the raw measurements to be used with no preprocessing and the other offers simplicity and smaller network sizes.

Chapter 7

State-of-Health Estimation for Li-ion Batteries Using Deep Convolutional Neural Networks

7.1 Pragmatism of Convolutional Neural Networks for SOH Estimation

This chapter looks at using a popular deep learning algorithm, the Convolutional Neural Network (CNN), for state-of-health estimation of Li-ion batteries. The CNN, which is described in more detail in Section 3.2.4, convolves kernels or filters over a two dimensional dataset. Model expressibility is learned through backpropagation which is also discussed in more detail in Section 3.3. Since the technique convolves kernels over two dimensional datasets, CNNs benefit from shared weights which reduces the amount of weights necessary to estimate SOC. Because of this, convolutional neural networks scale well in depth or when adding additional layers.

The experimental data used for SOC estimation in Chapter 5 and Chapter 6 is structured in a way where battery signals recorded for a certain discrete time step, usually at a sampling frequency of 0.1-1Hz, are fed into the networks and an estimated SOC is computed for that same time step. In contrast, the randomized battery usage dataset used in this chapter, obtained from the NASA Prognostics Center of Excellence [112], is structured in a way where the two dimensional dataset of the reference charge profile is provided to the network and an estimated SOH is computed for that particular reference dataset. Hence data on an complete or partial charge curve is fed into the network to estimate SOH instead of a discrete instance of battery measurements. In addition, since the reference charge profile is a time series, it is important to use an algorithm which is sensitive to correlations along one or both axis of the dataset. For these reasons, and for its shared weights, the CNN technique is chosen to be used for SOH estimation. The LSTM-RNN can also be used to analyze larger two dimensional input arrays to perform SOH estimation and has good performance over time series datasets, however LSTM-RNNs do not benefit from the efficiency of shared weights.

This chapter is completed with a final study which examines the importance of SOH estimation for SOC estimation. As such, this study will perform combined SOC and SOH estimation while using a fusion of the CNN SOH estimator and the LSTM-RNN SOC estimator. The CNN algorithm is used to estimate SOH which is used as an input to the LSTM-RNN which, in turn, estimates SOC.

7.2 Randomized Battery Usage Dataset

The dataset used in this work is the Randomized battery Usage Dataset obtained form the NASA Prognostics Center of Excellence [112]. This large repository of data contained seven files which each included aging data on 4 different Li-ion battery cells. The parameters of most of these datasets are shown in Table 7.1. Various LG Chem 18650 Li-ion cells are aged by undergoing a randomized load; ranging between 0.5 A and 5 A for some datasets and -4.5 A to 4.5 A for others, often at two different ambient temperatures; 25°C and 40°C. The random walk steps last for about 5 minutes. This dataset is unique since randomized usage serves as a better representation of real world loading profiles which makes for better training data. Reference charge and discharge profiles were conducted every 50 random cycles or every 5 days to characterize the cell's aging. These were sampled at 0.1 Hz and, as is typically performed, the charge curves in these characterization steps included constant current (CC) and constant voltage (CV). The CC segment of the charge profile typically consisted of the first 60% to 70% of the state-of-charge. This work

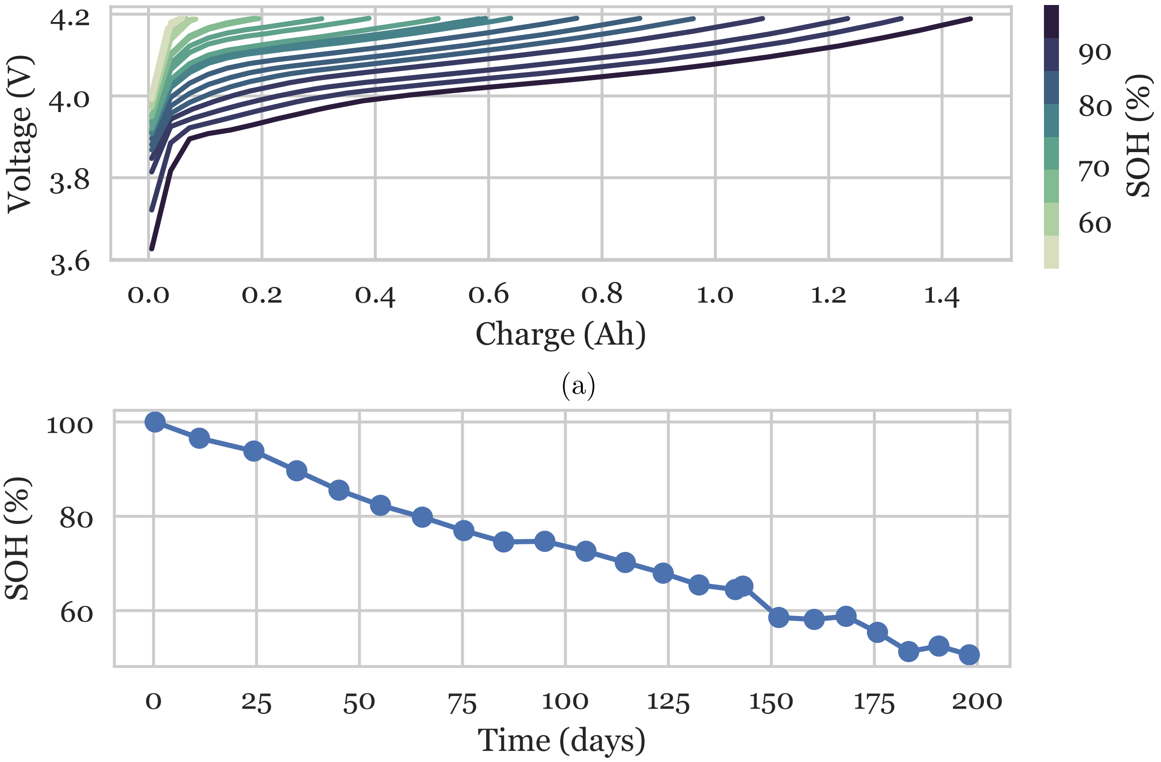
Table 7.1: Randomized Battery Usage Dastset Parameters

Capacity @ SOH=100%	Min. 2.08 Ah / Typ. 2.15 Ah
Min / Max Voltage	3.2V / 4.2V
Min / Max Temperature	15°C / 43°C
Charge Current (CC)	2.0 A
Random Walk Current	Min. -4.5 A / Max. 5.0 A

considers the fixed and partial SOC ranges where both the CC and the CV segments of the charge curves are utilized. Therefore, the CNN algorithm will not need to see full charge data in order to estimate SOH and can do so with both complete and partial charge profiles. For most of the 28 aging datasets in this repository, the cells are aged to at least SOH = 80% and in some cases are aged to less than SOH = 40%. An example of one aging dataset is shown in Figure 7.1.

7.2.1 Data Parsing and CNN Model Development

Data processing is a very critical step which is performed before constructing any deep learning algorithm. If not given its due process and due time, an undesired result is almost a certainty. However, the data processing performed in this work is unlike the processing performed in other works, where data-preprocessing includes scaling, normalizing, dimensionality reduction, etc., which can all increase the computational load on an algorithm aimed for on-board operation. Data preprocessing like scaling and normalization for deep learning algorithms is less of a requirement since these networks can adjust their weights to match most inputs. Raw data inputs can be used instead, reducing the computational load for on-board applications. In traditional machine learning techniques where non-linearities used were the hyperbolic tangent function or the sigmoid function, the activation functions would saturate at a value of



(a)

(b)

Figure 7.1: (a) CC segment of charge profile of a Li-ion cell throughout randomized usage aging process; color spectrum of profiles indicate SOH. (b) Recorded capacity-based SOH for each of the charge profiles in (a).

1. However, the ReLU non-linearity does not saturate for positive input values, which is partly the reason for the rise in its popularity observed in deep learning algorithms. This is not to say that this type of data preprocessing, e.g. scaling and normalizing, is not needed for deep learning methods, since in many cases some data processing can help a network achieve the desired response with less workload.

What is more demanding is data parsing and data purity, where much of the data-processing time was spent. This describes the process of parsing the data into its constituent loading cycles which include reference charge and discharge data as as well as the random walk data. To render the algorithm insensitive to historical loading conditions, like ambient temperature, cycling history and historical degradation data, the reference charge profiles are used to find a characteristic aging signature. The entire process from data parsing to the training an validation of the CNN are represented in a flowchart shown in Figure 7.2. Furthermore, outliers and incorrectly recorded data points are removed during this step. In addition, the stream of recorded data can sometimes be interrupted causing some datatsets to be disjointed. Therefore manual stitching of the data is often required in order to ensure proper continuity of the time dynamics inherent in the data. After parsing of the datasets is complete, the data is segmented into training and testing data files to ensure that a different dataset is used during training than is used during testing.

7.2.2 Data Augmentation

Adding statistical variances to the input data can increase the inference accuracy of the CNN. Interestingly, for SOH estimation application, data augmentation not only

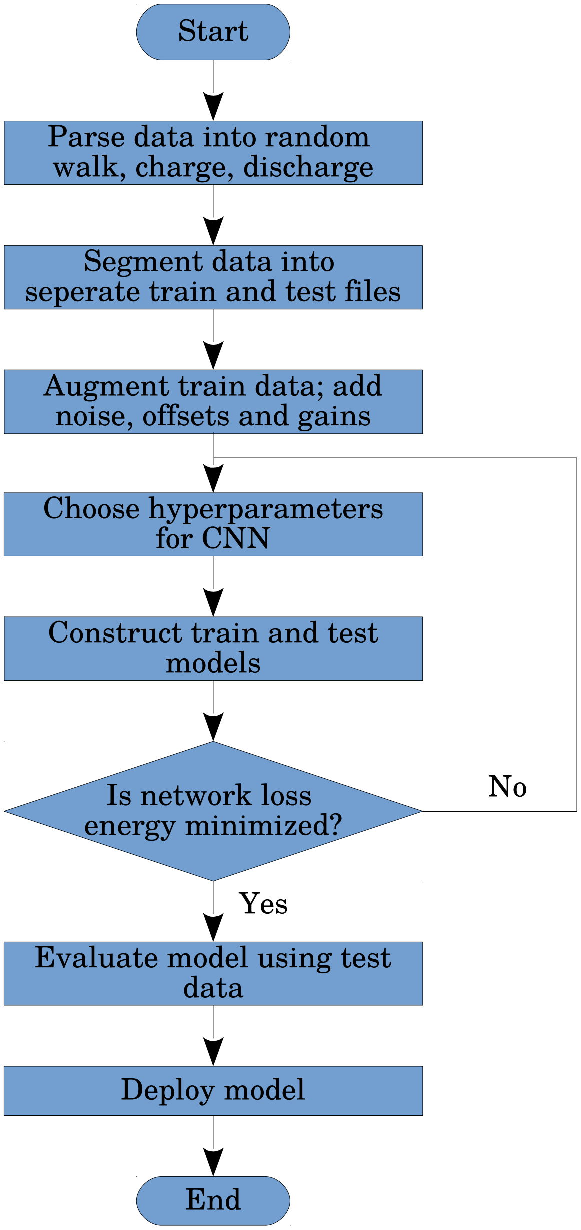

Start
Parse data into random walk, charge, discharge
Segment data into seperate train and test files
Augment train data; add noise, offsets and gains
Choose hyperparameters for CNN
Construct train and test models
Is network loss energy minimized?
No
Yes
Evaluate model using test data
Deploy model
End

Figure 7.2: Flowchart of the data parsing, training and validation of the convolutional neural network SOH estimation algorithm.

increases the estimation accuracy but also renders the CNN insensitive to measurement noise, gains and offsets. In effect, augmenting the training data can robustify the CNN, as discussed in Section 4.5.

Training data augmentation is performed by injecting white Gaussian noise into the measured battery signals. Specifically, Gaussian noise with 0 mean and a standard deviation of 1-4% is injected into the voltage, current and temperature measurements. Moreover, to robustify against offsets and gains inherent in battery measurement devices, an offset is applied to all measurement signals and a gain is applied only to the current measurement since current measurements are more susceptible to gains. An offset of up to ±150mA and a gain of up to ±3% is applied to the current measurements, an offset of up to ±5mV is applied to the voltage measurement and an offset of up to ±5°C is applied to the temperature measurement. Alternate copies of the training data are created with varying levels of noise, offsets and gains within the limits described above. Up to 80 augmented copies of the training data are created for training.

Although the CNNs in this work are trained with LG Chem cell aging data, other aging datasets from cells with different makes and models can be used for training. The CNN architectures will be unchanged if used to estimate SOH for a different battery however it might require to be retrained with data from that specific cell. Nevertheless, transfer learning, a useful property of deep learning algorithms, can be used to instead simply re-train the last one or few layers of the CNN to achieve similar estimation performance with a new battery. This would consume much less time than would otherwise be necessary to re-parameterize a battery model in traditional SOH estimation strategies.

7.3 State-of-Health Estimation Results

The state-of-health estimation performance of the deep convolutional networks are outlined in this section. As previously mentioned, this work uses charge profiles which include both fixed and variable SOC ranges. There are trade-offs which need to be made in either case and these will be discussed in the following two sections. The networks discussed in this section use a learning rate of 1×10^{-5}.

7.3.1 State-of-Health Estimation using Fixed Charge Profiles

In a mobile device or in an electric vehicle, periodic test cycles can be performed on a Li-ion battery. These can consist of fixed charge cycles like the ones used to obtain the results in this subsection. These fixed charge curves consist of a two dimensional array of inputs given by $\mathbf{\Psi}(\xi) \in \mathbb{R}^{N\times M}$ where N=256 and M=3, which include the measured battery current, voltage and temperature or more formally; $\mathbf{\Psi}(\xi) = [\mathbf{I}(\xi), \mathbf{V}(\xi), \mathbf{T}(\xi)]$. For a more formal description of the CNN and it's associated figure, please refer to Section 3.2.4 and Figure 3.5. Training is conducted on up to 26 of the aging datasets and validation is performed on 1 or 2 datasets, depending on the tests being performed. The validation datasets are never seen by the CNN during the training process. The output of the CNN is the estimated SOH for a particular charge profile. The time required to train the CNNs used to obtain the results in this subsection is 4 - 9 hours, depending on its size and depth.

The CNN is first validated on two aging datasets, referred to as dataset RW4 and RW23 in the NASA repository; there are recorded at 25°C and at 40°C, respectively. This CNN is composed of 6 convolution layers and 2 fully-connected layers (FC). The results, shown in Figure 7.3, Figure 7.4 and in Table 7.2, point to an MAE of

Table 7.2: SOH estimation accuracy of CNN validated on 25°C and 40°C RW aging datasets

Validation Dataset	MAE(%)	STDDEV(%)	MAX(%)
Validation RW dataset (25°C)	1.51	0.97	3.27
Validation RW dataset (40°C)	1.21	0.70	2.44

L1&L2: 32@(32,1), L3 - L6: 64@(32,1), FC1: 256 neurons

Table 7.3: Results showing inference for different case studies

Case Study	MAE(%)	STDDEV(%)	MAX(%)
Input: Voltage*	1.49	1.13	3.90
Input: Voltage, Current, Temperature*	1.03	0.62	2.24
No Pooling*	1.30	0.79	3.44
Pooling*	1.03	0.62	2.24
Unaugmented Train Data**	2.41	1.14	4.19
Augmented Train Data**	1.23	1.07	3.60
Smallest CNN***	1.85	1.62	6.06

*L1 - L6: 64@(32,1), FC1: 256 neurons
**L1&L2: 32@(32,1), L3 - L6: 64@(32,1), FC1: 256 neurons
***L1&L2: 4@(4,1), FC1: 32 neurons

1.51% and 1.21% for the 25°C and 40°C, respectively. The network is trained for up to 175,000 epochs although the largest decrease in error is observed within this first 100,000 training epochs. From Figure 7.4, it is clear that the 40°C aging dataset has fewer data points than the 25°C dataset but this is to be expected since the battery tends to age much quicker at higher ambient temperatures.

The CNN algorithm is put through further testing to investigate its performance. Its performance over these test cases are shown in Table 7.3. For most of these tests, training is stopped at 100,000 epochs to maintain testing objectivity. The first of

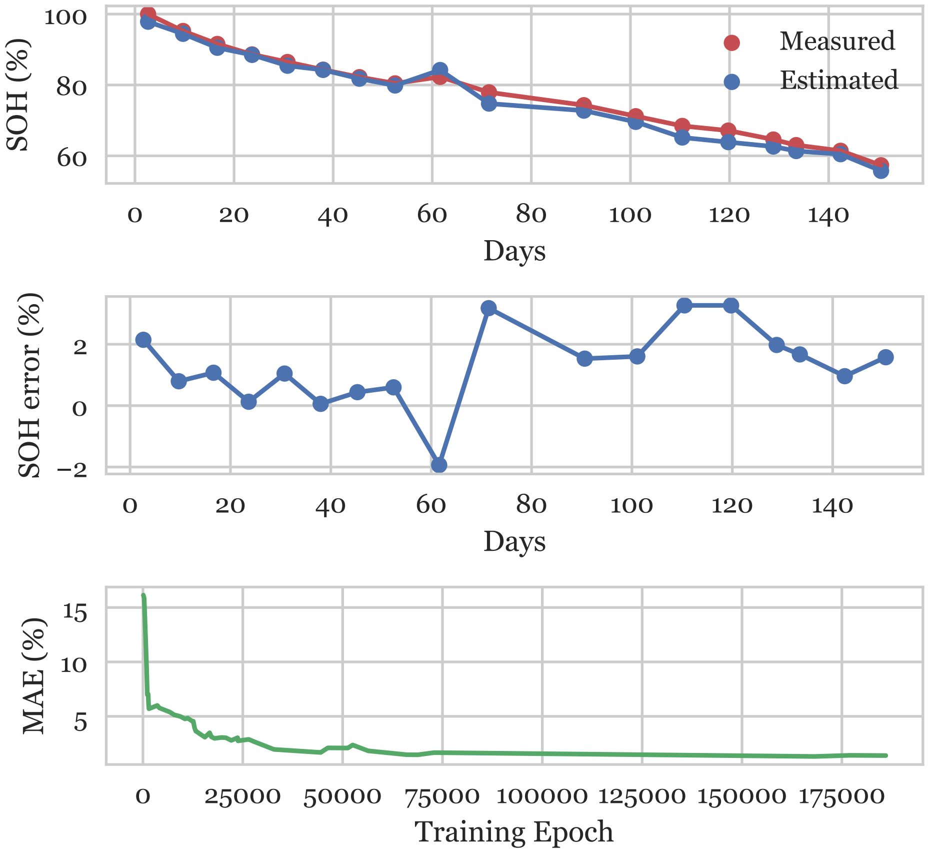


Figure 7.3: CNN estimation accuracy, estimation error over the 25°C validation dataset and the mean absolute error as a function of training epochs. MAE, MAX and network architecture are described in Table 7.2.

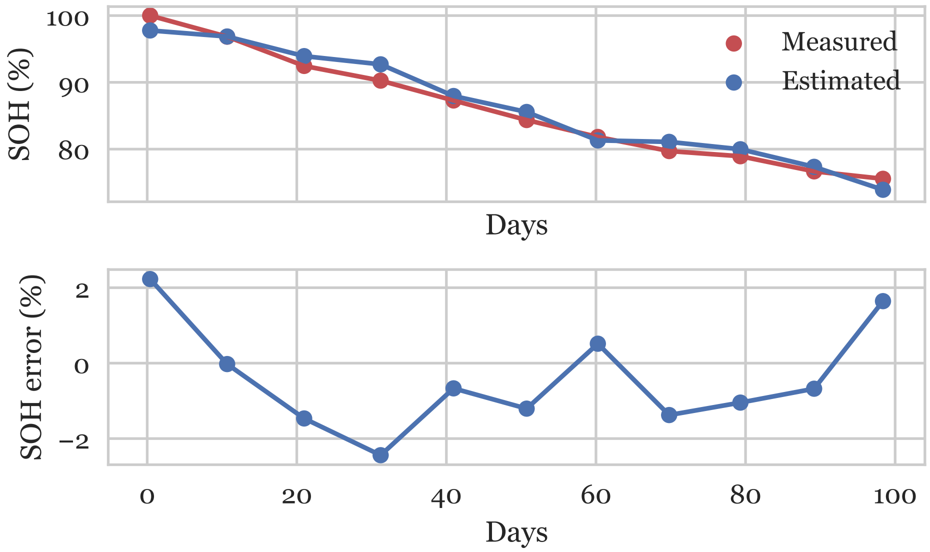


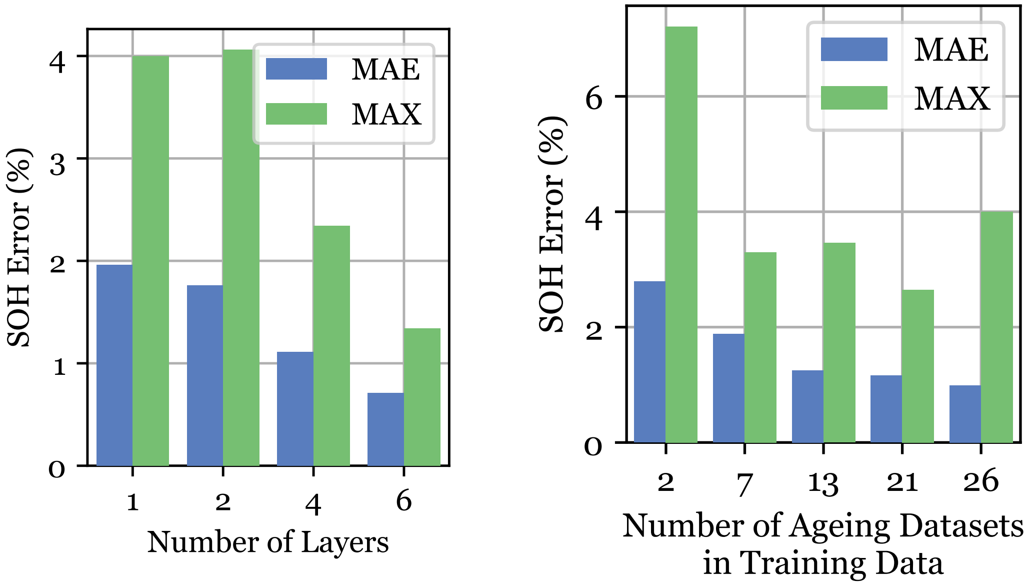
Figure 7.4: Estimation accuracy and estimation error over 40°C validation dataset. MAE, MAX and network architecture are described in Table 7.2.

such tests examines the accuracy of the CNN for inputs which include solely voltage as compared to inputs which include battery current, voltage and temperature. Reducing the dimensionality of the input data, reduces the number of computations which the CNN is required to perform during operation. The results show that the error when using only voltage as an input is satisfactory with an MAE and MAX of 1.49% and 3.90% respectively. However, the MAE and the MAX are reduced by 31% and 43%, respectively, when using all three input signals.

As described in Equation (3.9) of Section 3.2.4, pooling, $\mathcal{S}(.)$, is performed after a convolutional layer to subsample the layer activations. Although this is a layer that is often used for other applications, its efficacy in reducing the SOH estimation error was unknown. Therefore, to examine this for SOH estimation, two CNNs, where one has pooling layers and where the other has no pooling layers are used. The results of this case study are also shown in Table 7.3. It is found that a CNN with no pooling offers an MAE of 1.30% however a CNN with pooling has an MAE of 1.03%. Therefore, pooling improves the accuracy of the CNN for SOH estimation by about 21% however when considering the application of this algorithm in on-board systems the accuracy of the CNN with no pooling might be sufficient.

In the third test, we investigate if augmented training data, described in Section 7.2.2, affects the estimation accuracy of the CNN. This is performed by training two identical CNNs with identical architectures over augmented and unaugmented datasets. Augmentation describes the injection of Gaussian random noise as well as offsets and gains into the training data, as described in section III. Augmentation can offer the statistical variance for the input dataset to span more of the dynamics exhibited by the battery. An MAE and MAX of 2.41% and 4.19% is obtained with the unaugmented training dataset as opposed to the MAE and MAX of 1.23% and 3.60% obtained with the augmented dataset. Therefore, exposing the CNN to augmented training datasets offers good performance gains with a reduction in MAE and MAX of 49% and 15%, respectively.

In the final test case, the estimation performance is examined for a much smaller network since on-board applications often cannot allocate a large amount of computation to the SOH algorithm. Therefore, a network with 2 convolutional layers where each one is composed of 4 filters each having dimensions 4×1 and two fully-connected layers having 32 and 1 neurons, respectively, is used to estimate SOH. When trained with augmented training data, the MAE and MAX achieved by this small CNN over a validation dataset is 1.85% and 6.06%, respectively. Although, this network has adequate performance, further tests are conducted to asses the impact of network



(a)

(b)

Figure 7.5: (a) Estimation accuracy measured during validation versus number of layers in CNN. The number of neurons per layer is fixed to 4 neurons. (b) Estimation accuracy measured during validation versus number of training datasets. All tests are performed over validation datasets recorded at 25°C.

depth on the CNN. In Figure 7.5, the accuracy of convolutional neural networks at estimating SOH is recorded, first, as a function of network depth (number of layers) and, second, as a function of the amount of training data used during the training process. For each of these two test cases, the training data is augmented and training is stopped at 100,000 epochs. Clearly, deeper networks achieve increased estimation accuracy since going from 1 convolutional layer to 6 reduces the MAE by more than 60%. However, if an SOH estimation error of less than 0.8% is unnecessary and an error of less than 2% is sufficient for a particular application, than the computational burden can be lessened by opting for a 1 or 2 layer CNN.

In Figure 7.5(b), the estimation accuracy over a validation dataset is examined

Table 7.4: Results showing inference on corrupted validation dataset versus uncorrupted validation dataset

Validation Data Corruption	MAE(%)	STDDEV(%)	MAX(%)
No	1.40	0.94	3.17
Yes*	1.66	1.28	3.99

Architecture; L1 - L4:32@(31,1), FC1: 64 neurons
*Corrupted by injection of noise, gains and offsets.

as a function of the amount of training data required to achieve good estimation performance. Therefore, instead of training the CNN over 20 or more dataset as is performed in the previous results, different numbers of aging datasets are included during the training process and the CNN is evaluated over a validation dataset. In general, the more training data used to train the network, the lower of an MAE is achieved by the CNN. However, the argument to use more data than 13 to 21 datasets during training becomes hard to substantiate given the diminishing returns achieved when more than 21 datasets are used.

The battery measurements used for aging studies are typically obtained in controlled environments with high accuracy measurement devices. Therefore signal purity is ensured and noise is minimized. Although, this is necessary when measuring signals like the battery current, to ensure an accurate capacity measurement, it might reduce from the robustness of these algorithms, making them highly sensitive to measurement noise, offsets and gains. The latter become a concern when considering operation of the battery in real world applications where signal purity is of varying quality. The elegance of machine learning algorithms, like the CNN, is that they can learn to handle such measurement noise. Therefore, as performed in previous results, the CNN is trained on augmented training data where Gaussian noise, offsets and

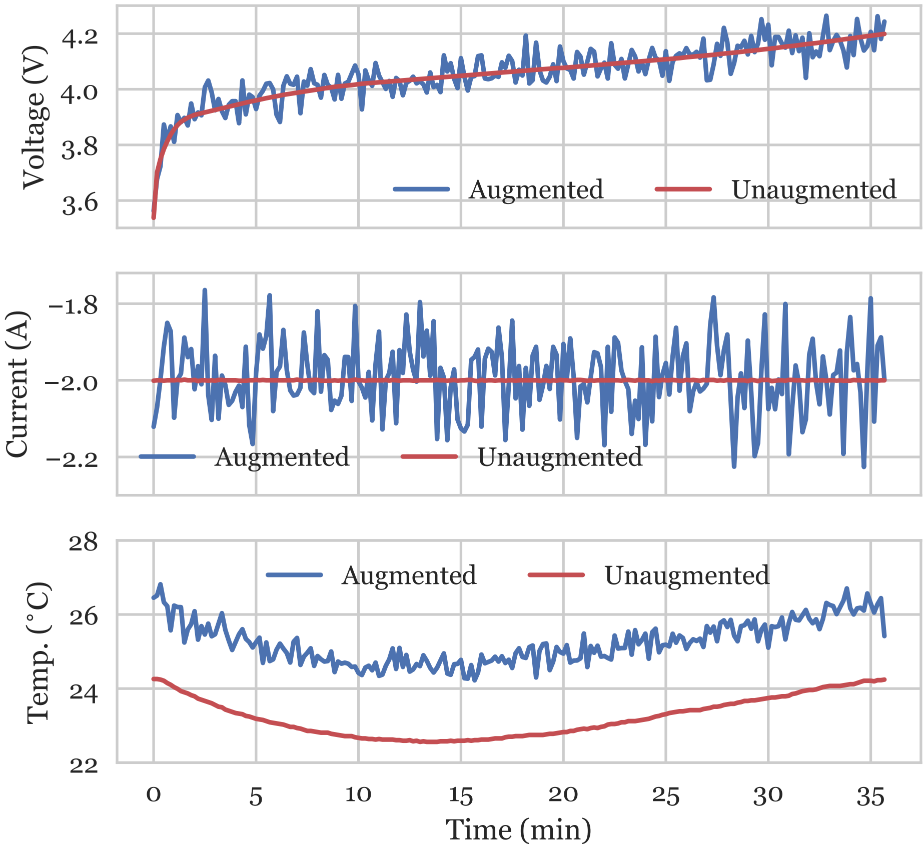


Figure 7.6: Corrupted and uncorrupted voltage, current and temperature battery signals of a reference charge profiles.

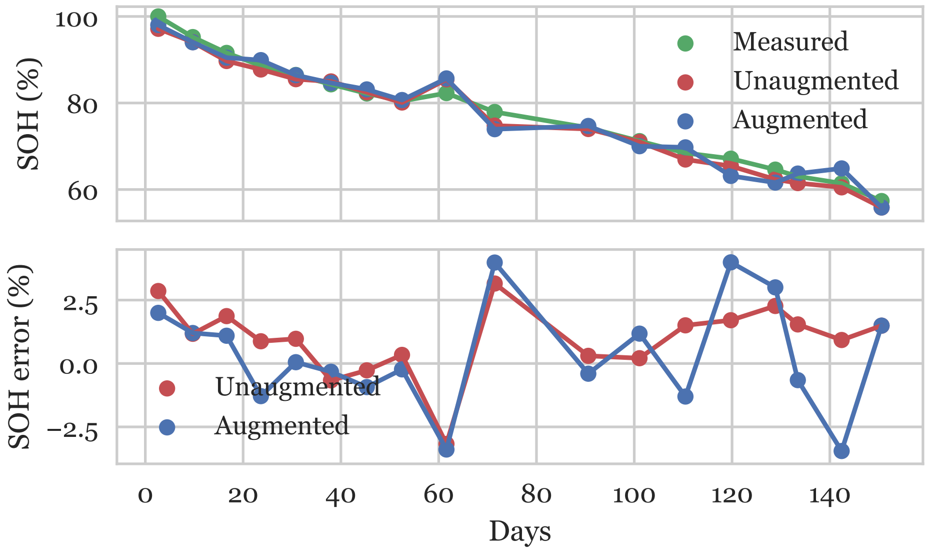


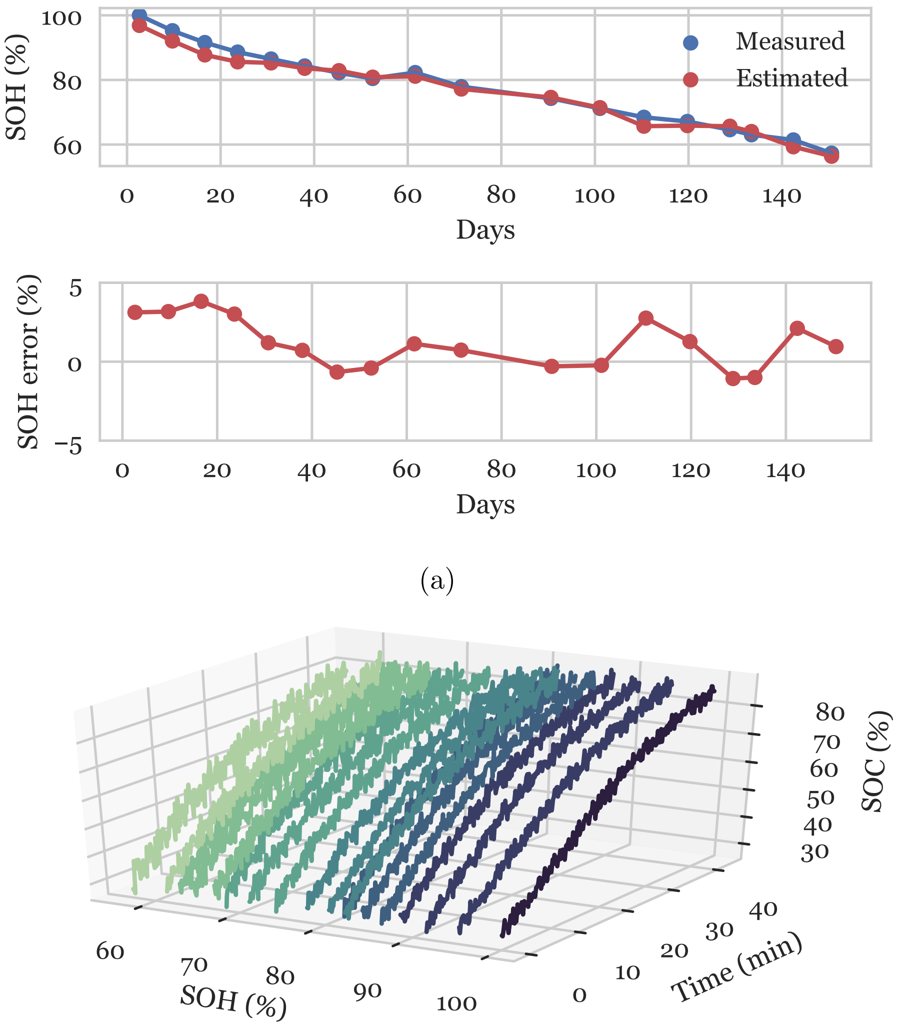
Figure 7.7: Estimation accuracy and estimation error over augmented and unaugmented battery validation data. Please see Table 7.4 for detailed results.

gains are injected into the data. However, the CNN is now tested over validation data which is also corrupted with injected noise, offsets and gains to mimic real world scenarios. This corrupt validation data is shown in Figure 7.6. In Figure 7.7 and in Table 7.4, the results for SOH estimation over corrupt validation data are shown. Specifically, normally distributed random noise with mean 0 and standard deviation of 1%, 1.5% and 5% is added to the voltage, current and temperature measurements, respectively. An offset of 5mV, 50mA and 2°C is added to the voltage, current and temperature measurements, respectively. A gain of 2% is only applied to the current measurements. Results with the corrupt validation data show good performance with an MAE of 1.66% which is only a 15% increase in error from the uncorrupted validation dataset.

7.3.2 State-of-Health Estimation using Partial Charge Profiles

Although the CNN achieves great performance in the previous subsection, the charge profiles used are fixed, such that the SOC ranges between 0% and about 60%. Although periodic reference profiles can be performed during normal operation to supply the SOH estimation algorithm with the required fixed charge profile, this can be avoided if the CNN is able to estimate SOH from partial charge curves. This facilitates the estimation of SOH for charge curves having varying SOC ranges. Therefore in this subsection, partial charge curves are used as inputs to the CNN during training as well as during validation. However, differences between charge curves having different SOC ranges around the same SOH value can be subtle. Therefore, it becomes necessary to include something other than voltage, current and temperature as an input so that these subtle differences can be recognized by the CNN. Hence, SOC, which is assumed to be continuously monitored by an electric vehicle, for example, is included as an input to the CNN.

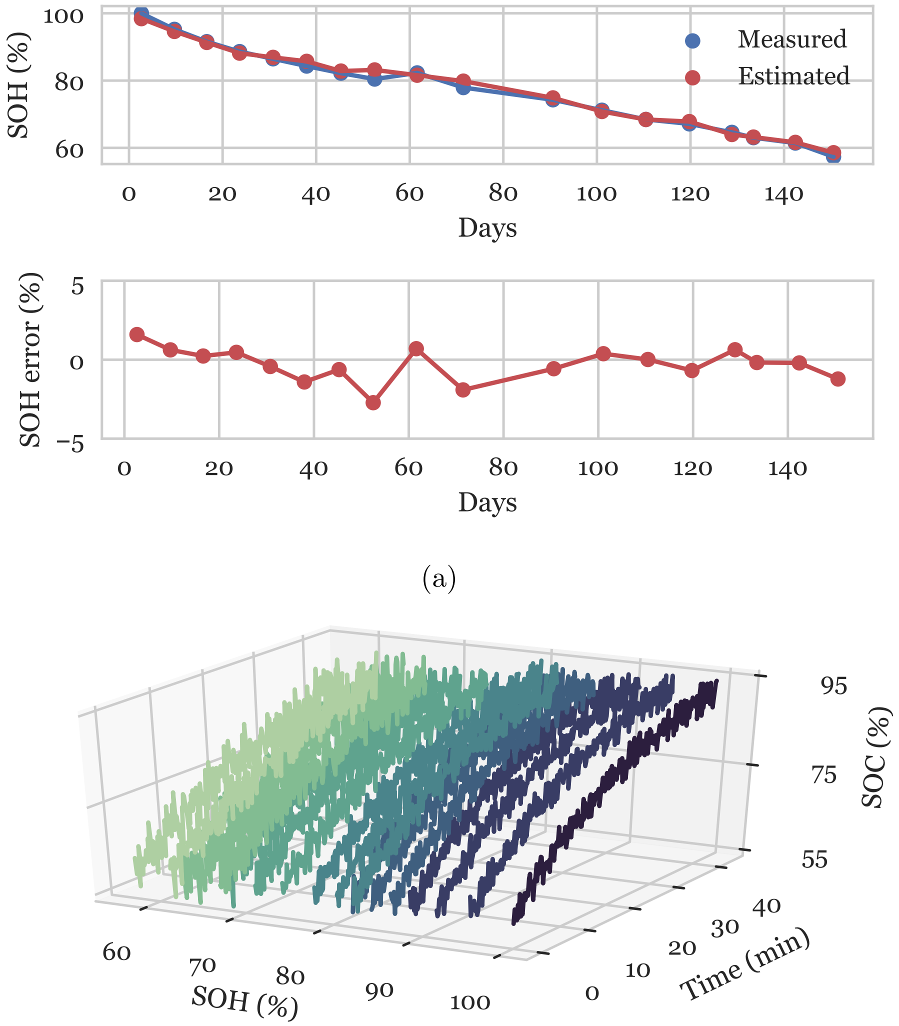
In Figure 7.8 and in Figure 7.9, SOH is estimated by the CNN using partial charge curves. The SOC range of the partial reference charge curves is shown in these figures as well. In Fig. 7.8, the SOC range of 30% to about 80% is used while in Figure 7.9 an SOC range of 60% to about 90% is utilized. Table 7.5 shows further results from other partial charge profiles having different SOC ranges. The SOC in these validation datasets is assumed to have an error of under 4% to simulate a real world scenario. Although not overwhelmingly obvious, the results show that the larger SOC ranges generally render better SOH estimation results. This is most likely attributed to the longer ranges of data which reveal more of the battery's aging signature. However,



(a)

(b)

Figure 7.8: (a) Plot showing SOH estimation results from CNN when given a partial charge profile as well as the corresponding error curve. (b) Partial charge profile beginning at SOC=30% and ending at about SOC=80% recorded at different states-of-health.



(a)

(b)

Figure 7.9: (a) Plot showing SOH estimation results from CNN when given a partial charge profile as well as the corresponding error curve. (b) Partial charge profile beginning at SOC=60% and ending at about SOC=95% recorded at different states-of-health.

Table 7.5: Results showing inference performed at different SOC ranges

SOC range	MAE(%)	STDDEV(%)	MAX(%)
25% - 80%	1.58	1.11	3.55
40% - 80%	1.56	1.02	3.46
60% - 95%	0.81	0.68	2.72
85% - 95%	1.60	0.88	3.53

Architecture; L1&L2: 64@(31,1), L3&L4: 64@(16,1)

the smaller SOC range of 85% to 95% achieves an MAE and a MAX of 1.60% and 3.53% which is nevertheless still competitive performance.

Figure 7.10 gives even greater resolution to the MAE and MAX values of SOH estimation performed over charge profiles beginning at various SOC values. The charge profiles used begin at different SOC values however they all have the same length of time. This time span measures 42.7 minutes. Interestingly, charge profiles beginning at either SOC=10% or around SOC=50-60% have the lowest MAE and MAX values. Since the time span of each partial charge profile is fixed, the partial charge curves beginning at those SOC values can be exposing less of the voltage plateau region where the aging signature is harder to discern. In addition, it's also clear that with higher SOC starting points, especially for those higher than SOC=80%, SOH estimation has a higher error. This can be due to less data being available to estimate SOH.

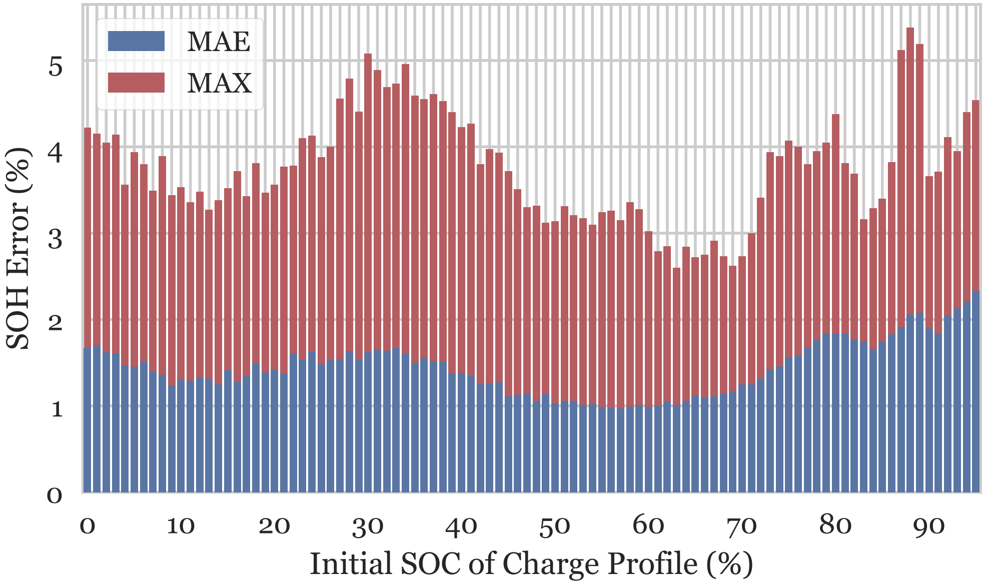


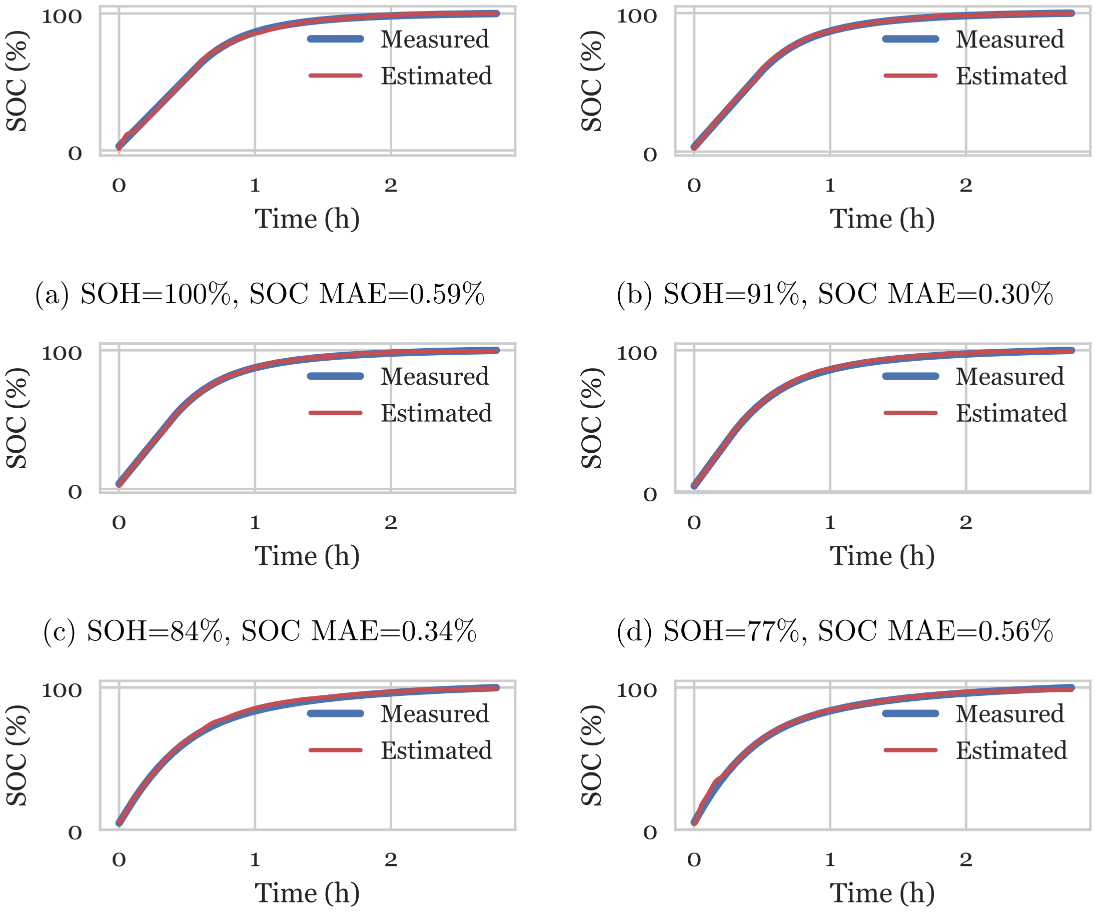
Figure 7.10: MAE and MAX values of SOH estimation performed over charge profiles beginning at various SOC values. The charge profiles are obtained from the NASA prognostics repository and are 42.7 minutes long.

7.4 A Fused Convolutional LSTM Recurrent Neural Network for Combined State-of-Health and State-of-Charge Estimation

This section furthers some of the previous works on SOH as well as SOC estimation by performing a combined SOH and SOC estimation strategy using a fusion of a Convolutional Neural Network and a Long Short-Term Memory Recurrent Neural Network (CNN-RNN). This strategy, explained in more detail in Section 3.2.5, utilizes the same CNN used in Section 7.3.1 to perform SOH estimation on fixed reference. The estimated SOH is then used as an input to a LSTM-RNN, similar to those used in Chapter 5, along with battery voltage, current and temperature to estimate SOC.

The two networks are trained separately and are tested in unison. The CNN model is developed in the same way as described in section Section 7.3.1, where fixed charge reference profiles are used. Training data is also augmented as done with CNNs previously. The LSTM-RNN is chosen for SOC estimation rather than the DNN, utilized in Chapter 6, because of its efficient representational power, specifically on time series data. In addition, the LSTM-RNN requires no preprocessing of the training and testing data. There are, of course, drawbacks to the use of LSTM-RNNs in comparison to DNNs and a discussion on this comparison is given previously in Section 6.1. Essentially, when using an LSTM-RNN for SOC estimation, there is no need to toil with the inputs in order to create secondary compositional inputs, like voltage and current averages for example.

The data used in this section is the same as that used in Section 7.3.1, where SOH estimation is performed over reference charge profiles from datasets found in



(a) SOH=100%, SOC MAE=0.59%

(b) SOH=91%, SOC MAE=0.30%

(c) SOH=84%, SOC MAE=0.34%

(d) SOH=77%, SOC MAE=0.56%

(e) SOH=68%, SOC MAE=0.92%

(f) SOH=63%, SOC MAE=0.71%

Figure 7.11: Examples of SOC estimation using the CNN-RNN. SOC is estimated over constant charge profiles when the battery is at different SOH levels. The data is recorded at 25°C.

the NASA prognostics repository. The reference charge profile's voltage, current and temperature are augmented as is previously performed. To perform SOH and SOC estimation in unison, it becomes necessary to use data from the same repository, hence removing the possibility of using the SOC drive cycle data used in Chapter 5 and Chapter 6, which do not contain aging information. Also, to ensure that the SOC estimation performance is correctly evaluated over the course of the battery's lifetime, controlled constant charge profiles are used. As mentioned previously in Section 3.2.5, the input vector which is given to the LSTM-RNN at every time instant is given by $\varPsi_k = [V(k), I(k), T(k), SOH(\xi)]$ where $V(k), I(k)$ and $T(k)$ represent the voltage, current and temperature of the battery at time step k and $SOH(\xi)$ is the SOH estimated during the last charge event. The estimated $SOH(\xi)$ remains the same between charge events which can be a length of time of a few hours to a few days. Meanwhile, SOC estimation occurs at a rate of 0.1Hz which is the sampling rate of the charge curves used for SOC estimation. For this section, a 25°C and a 40°C dataset are used for testing of the fused CNN-RNN model which are referred to as datasets RW4 and RW23 in the NASA prognostics repository. About 14 - 20 other datasets from this repository are used for training.

The CNN used in this study has the same architecture as the CNN used to obtain the results shown in Figure 7.3, Figure 7.4 and outlined in Table 7.2, which has 6 layers of convolution (kernel size of 32×1) and 2 fully connected layers with the first of these FC layers containing 256 neurons. On the other hand, the LSTM-RNN is composed a single LSMT layer having 500 nodes and a depth in time of $\widetilde{N} = 500$.

In Figure 7.11, SOC estimation is performed over charge profiles recorded at various instants of the battery's lifetime. A few examples are shown from dataset RW4

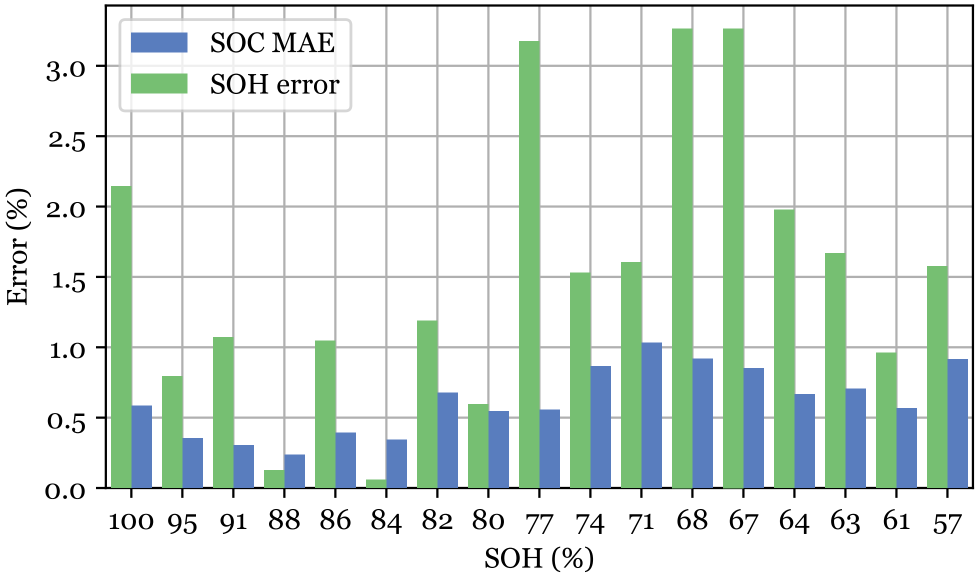


Figure 7.12: Histogram of the SOH estimation error for a charge profile and the Mean Absolute Error (MAE) of the SOC estimation for different values of SOH. SOH and SOC estimation performance is evaluated over the RW4 dataset recorded at 25°C.

recorded at 25°C. SOC estimation performance is very competitive for these charge curves but that's not a surprise given the great performance observed in Figure 5.4 of Section 5.3 where the LSTM-RNN is tested in a charging scenario. It's clear, however, that the SOC estimation performance is generally better earlier in the battery's lifetime than latter as SOC MAE is higher for smaller SOH values. This can be due to the fact that not all datasets include aging of the battery below SOH=80%, therefore there is less training data available at these lower SOH levels which is reflected in the error values observed below SOH=80%. However, the SOH estimation error is still quite competitive, especially above SOH=80% which is where most vehicle manufacturers like to declare the battery's End of Life (EOL).

For a more in depth look at the SOH and SOC estimation performance, Figure 7.12 and Figure 7.13, show a histogram of the SOH estimation error for an instance in the battery's lifetime and subsequently, the SOC MAE calculated over a complete charge curve. It's important to note that one value is an average value (SOC MAE) while the other (SOH error) is not. Generally, as expected, a lower SOC MAE is achieved for lower SOH errors and this same relative behavior is observed in both Figure 7.12 and Figure 7.13, recorded at 25°C and 40°C, respectively.

To validate the notion that state-of-health is an important value for SOC estimation, a final test is performed. This test evaluates the SOC estimation performance of the LSTM-RNN using only voltage, current and temperature as inputs, same as done in Chapter 5 instead of using the estimated SOH as an additional input. In other words, SOH estimation from the CNN is bypassed and instead the battery measurements are fed directly into the LSTM-RNN to perform SOC estimation. Thereafter, the SOC MAE is calculated for charge cycles taken at the various SOH points. The

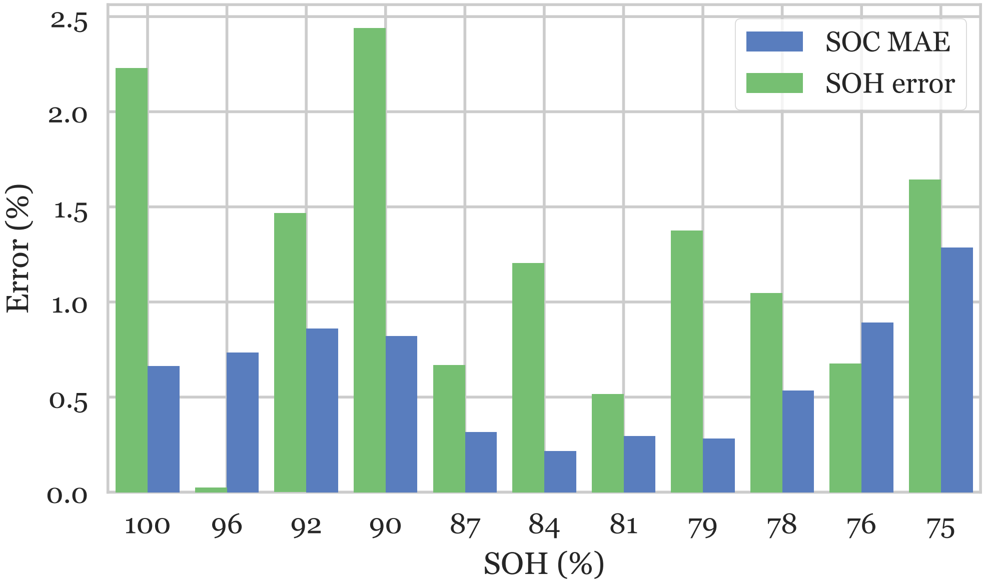


Figure 7.13: Histogram of the SOH estimation error for a charge profile and the Mean Absolute Error (MAE) of the SOC estimation for different values of SOH. SOH and SOC estimation performance is evaluated over the RW23 dataset recorded at 40°C from the NASA repository.

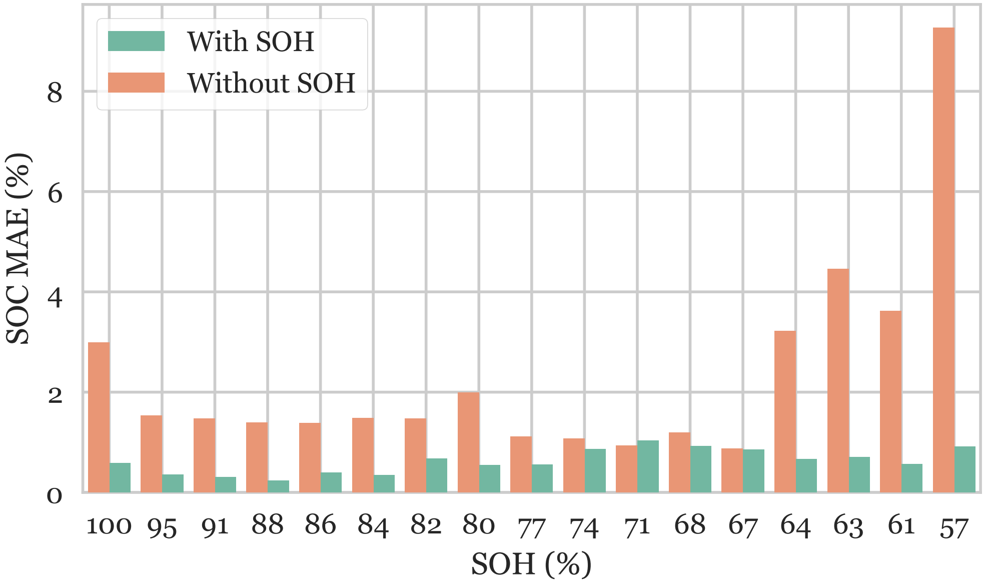


Figure 7.14: SOC estimation results from fused CNN-RNN with and without estimated SOH values supplied to the LSTM-RNN at 25°C.

results for the 25°C and the 40°C datasets are shown in Figure 7.14 and in Figure 7.15, respectively. Training of the CNN with and without estimated SOH as an input is stopped at 20000 epochs in both cases.

The results are undeniably in favor of using state-of-health for SOC estimation. These results are further evidence of the importance of SOH estimation to Li-ion batteries and energy management systems. Aside from the fact that SOH is very important to ensure safe operation of the battery and the vehicle, SOH is also key to the accurate estimation of SOC. For almost all of the battery's lifetimes, the SOC MAE is almost always lower when the estimated SOH is used as an input to the LSTM-RNN.

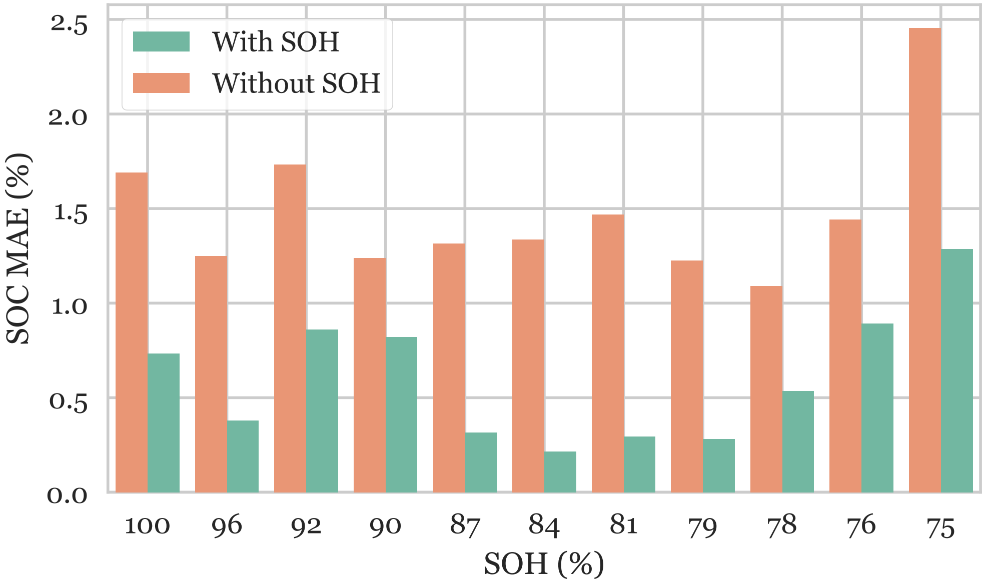


Figure 7.15: SOC estimation results from fused CNN-RNN with and without estimated SOH values supplied to the LSTM-RNN at 40°C.

Chapter 8

Conclusions, Future Research and Publications

8.1 Conclusions

In the final analysis, a new framework for SOC and SOH estimation is proposed in this research work using deep learning techniques. Li-ion batteries are increasingly becoming an important asset to humanity's overgrowing and dire need for clean energy. There are critical issues surrounding climate change and its catastrophic effects on our collective well-being, especially in major metropolitan areas which are experiencing the highest population growth rates. Li-ion batteries used in electrified vehicles offer much more efficiency than their petrol-powered counterparts. In addition, the steadily decreasing cost of Li-ion batteries is enticing a slow but steady adoption of electrified vehicles, whether land-, air- or sea-faring.

Accurate SOC estimation is critical to gauge a good measure of a vehicle's remaining driving range and is critical for battery balancing and for battery safeguards. State-of-Health estimation, on the other hand, is critical to the safe and reliable operation of any battery. Understanding the amount of degradation that the battery has experienced, is key in ensuring safe and reliable operation as well as customer satisfaction. In many earlier XEVs like the first generation Chevrolet Volt, the battery is significantly oversized given the electric driving range that it is claimed to have. Although this can be one way to account for a less effective SOH estimation strategy, it can definitely be an expensive one. The reliability and chemistries of modern Li-ion batteries is steadily improving. A review of most popular Li-ion battery chemistries and a thorough comparison over numerous USABC metrics and others is given in this work.

Energy management systems of electrified vehicles are also reviewed where protection of Li-ion batteries as well as various notable battery modeling techniques are discussed. Battery modeling techniques are crucial to the accurate determination of SOC and SOH in traditional approaches. They are similarly important to the approach proposed herein using deep learning techniques. However, the substantial representational power and expressibility of deep learning techniques allow these battery models to be learned through experience along with the SOC and SOH estimation routines. This not only removes the need to invest time and energy into parameter identification and hand-crafted battery models but allows the battery modeling and state estimation to be performed in one single step. This is also done without compromising on estimation accuracy. Deep learning methods not only allow for competitive estimation performance but also allow important information to be distilled from noisy environments. As shown in numerous examples and interesting outcome of using deep learning methods is that they can both easily be robustified against measurement noise, offsets and gains, and through this process, they can achieve better estimation accuracy. Corrupting the input data allows for statistical variances and for symmetries learned through training to be broken. This, in effect, requires the neurons in the networks to find new computational pathways which can lead to lower local minima of the loss function. Results from this phenomena can be seen in Figure 6.3, Figure 6.7 and Figure 7.7.

The efficiency of deep learning techniques is also a notable feature allowing same expressibility, if not better, with deeper layers. This is shown in Figure 6.4 although is more clear in Figure 7.5. Thus, this allows for higher fidelity models to be achieved by sensibly adding layers and adjusting network architectures instead of manually crafting the models to achieve better performance, as is done in more traditional methods. Additionally, universality of deep learning techniques can be increased by exposing the networks to data which increasingly spans the full gamut of possible environmental conditions. This is exemplified through the DNN and the LSTM-RNN models accruing the ability to estimate SOC at numerous ambient temperatures effectively, as seen in Figure 5.9 and in Figure 6.9. To further this point, the CNN performing SOH estimation is also generalized by learning to estimate SOH given partial charge profiles, shown in Figure 7.8, Figure 7.9 and in Figure 7.10.

This research work is completed with a fused convolutional recurrent neural network which performed combined SOC and SOH estimation. This strategy capitalizes on the expressibility of LSTM-RNNs over time series data when estimating SOC and the efficiency as well as the representational power of the CNN when estimating SOH. State-of-health is estimated with a CNN which, in turn, is fed into the LSTM-RNN to perform SOC estimation. This is performed over controlled constant charge profiles to asses SOC over the lifetime of the battery. The results from this fused CNN-RNN are strikingly impressive. SOH and SOC estimation performance is competitive. However, the most engaging result from this study is the direct evidence that knowledge of SOH, allows for sustained accurate SOC estimation throughout the lifetime of the battery. This is shown in Figure 7.14 and Figure 7.15.

8.2 Future Research

This area of battery research performing battery modeling and battery state estimation involving deep learning techniques is still quite new and as such numerous future projects can be discussed. For this discussion, the author believes that there are three important areas of future work which can be conducted based on the research outlined in this thesis. These include sensor modeling, prolonged on-board testing and integration into autonomous car routines as well as achieving increased universality through cloud-based deep learning models.

Although deep learning algorithms are used in this thesis, these algorithms can, generally, be regarded as supervised learning algorithms. These are based on teacher-learner strategies which require labeled or measured observable data to learn a representation of the underlying system. As such, since these observables are not available in a mobile application like electrified vehicles, it is not possible to perform the learning process in the vehicle. Thus why learning is conducted offline and when loss energy is minimized, and the network is able to express the underlying system with sufficient accuracy, can the model be applied online. Allowing these algorithms to learn online requires accurate measurements of battery observables. This is generally not possible since the measurement devices used are not of the high accuracy sensors used in the laboratory. For example, Coulomb counting from an off-the-shelf automotive grade current sensory, as shown in Figure 6.5, does not provide a good estimate of SOC. Therefore, using a deep learning technique to model the behavior of the current sensor and correct for its inaccuracies can enable the use of an error-prone sensor in order to perform Coulomb counting in an online fashion.

Although experimental data is gathered, and processed for this research work, more rigorous testing performed on-board a vehicle is important to the development of these algorithms for operation in various terrains, environmental stresses and for prolonged periods of time. Developing an extended testing profile on-board a vehicle will allow for even further real world testing. Vehicles are becoming increasingly autonomous with IHS Markit stating that level 4 autonomous vehicles can be deployed as early as 2021 and level 5 as early as 2025 although they claim that market penetration of these two levels of autonomous vehicles will not be witnessed till the 2030s. Since the Li-ion battery is responsible for the driving range available in a vehicle, SOC and SOH estimation can very well be included in route optimization algorithms for autonomous vehicles. Hence, streams of data from prolonged testing can be collected and fully utilized with data-driven approaches like deep learning algorithms.

An obvious direction to further this research but one whose advantages are not initially entirely clear is the application of these deep learning SOC and SOH estimation techniques in the cloud. Li-ion batteries are becoming ubiquitous given the increased popularity of mobile devices, electric vehicles and renewable energies, in general. As such a pragmatic SOC and SOH estimation framework is necessary to ensure their safe and reliable operation. Furthermore, with the large amounts of data which can be generated by batteries, it becomes important to consider data-driven approaches like deep learning techniques to accurately and reliably estimate SOC/SOH and ensure the safe operation of these devices. Therefore, streaming the battery data to a cloud computer which updates the model weights based on the new information can be a milestone in the continued real time improvement of estimation accuracy. The algorithms do not have to be stored in the cloud, however learning can be performed in the cloud and the model weights stored in the vehicle can be updated in real time. Therefore, this ensures model execution to be unaffected by intermittent interruptions to the communications system between the vehicle and the cloud computer.

The performance of these deep learning models can be generalized and universality can be achieved with the collection of large corpora of data. Nowadays, vehicle manufacturers can collect vehicle and battery data from most if not all the vehicles driven by their customers. Continuous streams of real world data can be added to the training routine of these deep learning models. In other words, a feedback loop can be created whereby data from the batteries is collected and added to the training and validation datasets used to train the deep learning models. In addition, data from different battery cells can be also used in order to generalize the model to two or more battery types.

Finally, a clear extension of this work can be to use these deep learning algorithms for energy and power management over the grid. The market share of electrified vehicles is increasing. In addition, we are currently witnessing an insurgence of solar as well as battery-powered systems for consumers. Therefore, many new sources and sinks of power are being added progressively to the grid. Managing load and power demand in light of this can be a high dimensional problem that is very suitable for data driven approaches like deep learning. Deep learning algorithms can learn from utilities data and become autonomous systems able to perform energy management and operational planning for the grid.

8.3 Publications

The following publications have resulted from this Ph.D. study:

Published journal papers

1. Chemali, Ephrem, Phil Kollmeyer, Matthias Preindl, Ryan Ahmed, and Ali Emadi. ”Long Short-Term Memory-Networks for Accurate State of Charge Estimation of Li-ion Batteries.” IEEE Transactions on Industrial Electronics

(2017).

2. Chemali, Ephrem, Matthias Preindl, Pawel Malysz, and Ali Emadi. ”Electrochemical and electrostatic energy storage and management systems for electric drive vehicles: State-of-the-art review and future trends.” IEEE Journal of Emerging and Selected Topics in Power Electronics 4, no. 3 (2016): 1117-1134.

Journal papers under review

1. Chemali, Ephrem, Phil Kollmeyer, Matthias Preindl, and Ali Emadi. ”State-of-Charge Estimation of Li-ion Batteries using Deep Neural Networks: A Machine Learning Approach.” Journal of Power Sources (submitted 2017).

2. Chemali, Ephrem, Matthias Preindl, Phil Kollmeyer, and Ali Emadi. ”A Deep Learning Approach for State of Health Estimation of Li-ion Batteries.” Applied Energy (submitted 2018).

International refereed conference papers

1. Chemali, E. and Emadi, A., 2017, June. On the concept of a novel Reconfigurable Multi-Source Inverter. In Transportation Electrification Conference and Expo (ITEC), 2017 IEEE (pp. 707-713). IEEE.

2. Chemali, E., McCurlie, L., Howey, B., Stiene, T., Rahman, M.M., Preindl, M., Ahmed, R. and Emadi, A., 2015, November. Minimizing battery wear in a hybrid energy storage system using a linear quadratic regulator. In Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE (pp. 003265-003270). IEEE.

References

[1] J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer, "The contribution of outdoor air pollution sources to premature mortality on a global scale," *Nature*, vol. 525, no. 7569, p. 367, 2015.

[2] A. Emadi, *Advanced Electric Drive Vehicles.* New York: CRC Press, 2015.

[3] E. Chemali, M. Preindl, P. Malysz, and A. Emadi, "Electrochemical and electrostatic energy storage and management systems for electric drive vehicles: State-of-the-art review and future trends," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 4, no. 3, pp. 1117–1134, 2016.

[4] R. Hemmati and H. Saboori, "Short-term bulk energy storage system scheduling for load leveling in unit commitment: modeling, optimization, and sensitivity analysis," *Journal of advanced research*, vol. 7, no. 3, pp. 360–372, 2016.

[5] E. Reihani, S. Sepasi, L. R. Roose, and M. Matsuura, "Energy management at the distribution grid using a battery energy storage system (bess)," *International Journal of Electrical Power & Energy Systems*, vol. 77, pp. 337–344, 2016.

[6] A. Cherp and J. Jewell, "The three perspectives on energy security: intellectual

history, disciplinary roots and the potential for integration," *Current Opinion in Environmental Sustainability*, vol. 3, no. 4, pp. 202–212, 2011.

[7] C. Winzer, "Conceptualizing energy security," *Energy policy*, vol. 46, pp. 36–48, 2012.

[8] T. Couture and Y. Gagnon, "An analysis of feed-in tariff remuneration models: Implications for renewable energy investment," *Energy policy*, vol. 38, no. 2, pp. 955–965, 2010.

[9] W. Sierzchula, S. Bakker, K. Maat, and B. van Wee, "The influence of financial incentives and other socio-economic factors on electric vehicle adoption," *Energy Policy*, vol. 68, pp. 183–194, 2014.

[10] B. Bilgin, P. Magne, P. Malysz, Y. Yang, V. Pantelic, M. Preindl, A. Korobkine, W. Jiang, M. Lawford, and A. Emadi, "Making the Case for Electrified Transportation," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1, pp. 4–17, 2015.

[11] B. Nykvist and M. n. Nilsson, "Rapidly falling costs of battery packs for electric vehicles," *Nature Climate Change*, vol. 5, no. 4, pp. 329–332, 2015.

[12] "us advanced battery consortium ev battery goals". (2013)

[13] O. Egbue and S. Long, "Barriers to widespread adoption of electric vehicles: An analysis of consumer attitudes and perceptions," *Energy policy*, vol. 48, pp. 717–729, 2012.

[14] M. K. Hidrue, G. R. Parsons, W. Kempton, and M. P. Gardner, "Willingness to

pay for electric vehicles and their attributes," *Resource and energy economics*, vol. 33, no. 3, pp. 686–705, 2011.

[15] P. Malysz, R. Gu, J. Ye, H. Yang, and A. Emadi, "State-of-charge and state-of-health estimation with state constraints and current sensor bias correction for electrified powertrain vehicle batteries," *IET Electrical Systems in Transportation*, p. 9 pp., 2016.

[16] L. McCurlie, M. Preindl, and A. Emadi, "Fast model predictive control for redistributive lithium-ion battery balancing," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1350–1357, 2017.

[17] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles," *Journal of Power Sources*, vol. 258, pp. 321 – 339, 2014.

[18] R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, and S. Habibi, "Reduced-Order Electrochemical Model Parameters Identification and SOC Estimation for Healthy and Aged Li-Ion Batteries. Part I: Parameterization Model Development for Healthy Batteryies," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 659–677, 2014.

[19] Z. Li, J. Huang, B. Y. Liaw, and J. Zhang, "On state-of-charge determination for lithium-ion batteries," *Journal of Power Sources*, vol. 348, no. Supplement C, pp. 281 – 301, 2017.

[20] M. A. Roscher and D. U. Sauer, "Dynamic electric behavior and open-circuit-voltage modeling of LiFePO4-based lithium ion secondary batteries," *Journal of Power Sources*, vol. 196, no. 1, pp. 331 – 336, 2011.

[21] J. Li, J. K. Barillas, C. Guenther, and M. A. Danzer, "A comparative study of state of charge estimation algorithms for lifepo4 batteries used in electric vehicles," *Journal of Power Sources*, vol. 230, pp. 244 – 250, 2013.

[22] L. Liu, L. Y. Wang, Z. Chen, C. Wang, F. Lin, and H. Wang, "Integrated system identification and state-of-charge estimation of battery systems," *IEEE Transactions on Energy Conversion*, vol. 28, no. 1, pp. 12–23, Mar. 2013.

[23] M. Gholizadeh and F. Salmasi, "Estimation of state of charge, unknown non-linearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1335–1344, Mar. 2014.

[24] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles," *Journal of Power Sources*, vol. 246, pp. 667 – 678, 2014.

[25] C. Hu, B. D. Youn, and J. Chung, "A multiscale framework with extended kalman filter for lithium-ion battery {SOC} and capacity estimation," *Applied Energy*, vol. 92, pp. 694 – 704, 2012.

[26] F. Sun, X. Hu, Y. Zou, and S. Li, "Adaptive unscented kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles," *Energy*, vol. 36, no. 5, pp. 3531 – 3540, 2011.

[27] S. Sepasi, R. Ghorbani, and B. Y. Liaw, "Improved extended kalman filter for state of charge estimation of battery pack," *Journal of Power Sources*, vol. 255, no. Supplement C, pp. 368 – 376, 2014.

[28] S. M. Rezvanizaniani, Z. Liu, Y. Chen, and J. Lee, "Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (ev) safety and mobility," *Journal of Power Sources*, vol. 256, pp. 110 – 124, 2014.

[29] P. Cicconi, D. Landi, A. Morbidoni, and M. Germani, "Feasibility analysis of second life applications for li-ion cells used in electric powertrain using environmental indicators," in *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*, pp. 985 – 990, 2012.

[30] C. K. Dyer, P. T. Moseley, Z. Ogumi, D. A. J. Rand, and B. Scrosati, *Encyclopedia of Electrochemical Power Sources*. Elsevier Science, 2009.

[31] W. Wang, J. Ye, P. Malysz, H. Yang, and A. Emadi, "Sensitivity analysis of Kalman Filter based capacity estimation for electric vehicles," in *2015 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–7. IEEE, 2015.

[32] P. Malysz, J. Ye, R. Gu, H. Yang, and A. Emadi, "Battery State-of-Power Peak Current Calculation and Verification using an Asymmetric Parameter Equivalent Circuit Model," *IEEE Transactions on Vehicular Technology*, vol. 9545, no. c, pp. 1–1, 2015.

[33] N. Omar, M. Daowd, P. V. D. Bossche, O. Hegazy, J. Smekens, T. Coosemans, and J. V. Mierlo, “Rechargeable Energy Storage Systems for Plug-in Hybrid Electric VehiclesAssessment of Electrical Characteristics,” *Energies*, vol. 5, no. 12, pp. 2952–2988, 2012.

[34] J. L. Sudworth, “The sodium/nickel chloride (ZEBRA) battery,” *Journal of Power Sources*, vol. 100, pp. 149–163, 2001.

[35] J. Dixon, I. Nakashima, E. Arcos, and M. Ortuzar, “Electric vehicle using a combination of ultracapacitors and zebra battery,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 3, pp. 943–949, Mar. 2010.

[36] H. Chen, T. N. Cong, W. Yang, C. Tan, Y. Li, and Y. Ding, “Progress in electrical energy storage system: A critical review,” *Progress in Natural Science*, vol. 19, no. 3, pp. 291–312, 2009.

[37] S. Sabihuddin, A. Kiprakis, and M. Mueller, “A Numerical and Graphical Review of Energy Storage Technologies,” *Energies*, vol. 8, no. 1, pp. 172–216, 2014.

[38] T. Ying, X. Gao, W. Hu, F. Wu, and D. Noreus, “Studies on rechargeable NiMH batteries,” *International Journal of Hydrogen Energy*, vol. 31, no. 4, pp. 525–530, 2006.

[39] P. Albertus, J. Christensen, and J. Newman, “Modeling Side Reactions and Nonisothermal Effects in Nickel Metal-Hydride Batteries,” *Journal of The Electrochemical Society*, vol. 155, no. 1, p. A48, 2008.

[40] W. Kempton and T. Kubo, "Electric-drive vehicles for peak power in Japan," *Energy Policy*, vol. 28, no. 1, pp. 9–18, 2000.

[41] P. Bäuerlein, C. Antonius, J. Löffler, and J. Kümpers, "Progress in high-power nickelmetal hydride batteries," *Journal of Power Sources*, vol. 176, no. 2, pp. 547–554, 2008.

[42] F. Torabi and V. Esfahanian, "Study of Thermal-Runaway in Batteries: II. The Main Sources of Heat Generation in Lead-Acid Batteries," *Journal of the Electrochemical Society*, vol. 160, no. 2, pp. A223–A234, 2012.

[43] D. Pavlov, "Energy balance of the closed oxygen cycle and processes causing thermal runaway in valve-regulated lead/acid batteries," *Journal of Power Sources*, vol. 64, pp. 131–137, 1997.

[44] P. H. Smith, T. N. Tran, T. L. Jiang, and J. Chung, "Lithium-ion capacitors: Electrochemical performance and thermal behavior," *Journal of Power Sources*, vol. 243, pp. 982–992, 2013.

[45] G. Mulder, N. Omar, S. Pauwels, M. Meeus, F. Leemans, B. Verbrugge, W. De Nijs, P. Van Den Bossche, D. Six, and J. Van Mierlo, "Comparison of commercial battery cells in relation to material properties," *Electrochimica Acta*, vol. 87, pp. 473–488, 2013.

[46] Y. Mikhaylik, I. Kovalev, R. Schock, K. Kumaresan, J. Xu, and J. Affinito, "High energy rechargeable li-s cells for ev application: Status, remaining problems and solutions," *ECS Transactions*, vol. 25, no. 35, pp. 23 – 34, 2010.

[47] Department of Energy, “Advanced Battery Development - 2013 Annual Progress Report,” 2013.

[48] M. Hagen, D. Hanselmann, K. Ahlbrecht, R. Maca, D. Gerber, and J. Tubke, “Lithium-sulfur cells: The gap between the state-of-the-art and the requirements for high energy battery cells,” *Advanced Energy Materials*, vol. 5, no. 16, 2015.

[49] H. Popp, J. Attia, F. Delcorso, and A. Trifonova, “Lifetime analysis of four different lithium ion batteries for (plug in) electric vehicle,” in *Transportation Electrification Conference and Expo (ITEC), 2015 IEEE*, 2015.

[50] W. Xu, H. Peng, J. Huang, C. Zhao, X.-B. Cheng, and Q. Zhang, “Towards stable lithiumsulfur batteries with a low self-discharge rate: Ion diffusion modulation and anode protection,” *ChemSusChem*, vol. 8, no. 17, pp. 2892–2901, 2015.

[51] F. Xu, H. He, Y. Liu, C. Dun, Y. Ren, Q. Liu, M.-x. Wang, and J. Xie, “Failure Investigation of LiFePO4 Cells under Overcharge Conditions,” *Journal of The Electrochemical Society*, vol. 159, no. 5, p. A678, 2012.

[52] A. W. Golubkov, D. Fuchs, J. Wagner, H. Wiltsche, C. Stangl, G. Fauler, G. Voitic, A. Thaler, and V. Hacker, “Thermal-runaway experiments on consumer Li-ion batteries with metal-oxide and olivin-type cathodes,” *RSC Advances*, vol. 4, no. 7, pp. 3633–3642, 2014.

[53] A. W. Golubkov, S. Scheikl, R. Planteu, G. Voitic, H. Wiltsche, C. Stangl, G. Fauler, A. Thaler, and V. Hacker, “Thermal runaway of commercial 18650

Li-ion batteries with LFP and NCA cathodes impact of state of charge and overcharge," *RSC Adv.*, vol. 5, no. 70, pp. 57 171–57 186, 2015.

[54] J. Seo, S. Sankarasubramanian, C.-S. Kim, P. Hovington, J. Prakash, and K. Zaghib, "Thermal characterization of Li/sulfur, Li/ SLiFePO4 and Li/SLiV3O8 cells using Isothermal Micro-Calorimetry and Accelerating Rate Calorimetry," *Journal of Power Sources*, vol. 289, pp. 1–7, 2015.

[55] H. A. Catherino, "Complexity in battery systems: Thermal runaway in VRLA batteries," *Journal of Power Sources*, vol. 158, no. 2 SPEC. ISS., pp. 977–986, 2006.

[56] M. Takahashi, K. Komatsu, and K. Maeda, "The Safety Evaluation Test of Lithium-Ion Batteries in Vehicles - Investigation of Overcharge Test Method -," *ECS Transactions*, vol. 41, no. 39, pp. 27–41, 2012.

[57] Q. Wang, P. Ping, X. Zhao, G. Chu, J. Sun, and C. Chen, "Thermal runaway caused fire and explosion of lithium ion battery," *Journal of Power Sources*, vol. 208, pp. 210–224, 2012.

[58] C. Y. Wen, C. Y. Jhu, Y. W. Wang, C. C. Chiang, and C. M. Shu, "Thermal runaway features of 18650 lithium-ion batteries for LiFePO4 cathode material by DSC and VSP2," *Journal of Thermal Analysis and Calorimetry*, vol. 109, no. 3, pp. 1297–1302, 2012.

[59] K. Yeow and H. Teng, "Characterizing Thermal Runaway of Lithium-ion Cells in a Battery System Using Finite Element Analysis Approach," *SAE Int. J. Alt. Power.*, vol. 2, no. 1, pp. 179–186, 2013.

[60] T. J. Knipe, L. Gaillac, and J. Argueta, "100,000-Mile Evaluation of the Toyota RAV4 EV," *Southern California Edison*, pp. 1–12, 2005.

[61] T. Reddy, *Linden's Handbook of Batteries, 4th Edition.* McGraw-Hill Education, 2010, Chapter 22.

[62] D. Wetz, P. Novak, B. Shrestha, J. Heinzel, and S. Donahue, "Electrochemical energy storage devices in pulsed power," *IEEE Transactions on Plasma Science*, vol. 42, no. 10, pp. 3034–3042, 2014.

[63] C.-H. Dustmann, "Advances in ZEBRA batteries," *Journal of Power Sources*, vol. 127, no. 12, pp. 85 – 92, 2004.

[64] K. B. Hueso, M. Armand, and T. Rojo, "High temperature sodium batteries: status, challenges and future trends," *Energy & Environmental Science*, vol. 6, p. 734, 2013.

[65] M. R. Palacín, "Recent advances in rechargeable battery materials: a chemist's perspective." *Chemical Society reviews*, vol. 38, pp. 2565–2575, 2009.

[66] I. J. Fernández, C. F. Calvillo, A. Sánchez-Miralles, and J. Boal, "Capacity fade and aging models for electric batteries and optimal charging strategy for electric vehicles," *Energy*, vol. 60, pp. 35–43, 2013.

[67] T. Reddy, *Linden's Handbook of Batteries, 4th Edition.* McGraw-Hill Education, 2010, Chapter 26.

[68] Panasonic, "NCR18650 Catalog," 2012, Version 13.11 R1.

[69] Y. Su, S. Cui, Z. Zhuo, W. Yang, X. Wang, and F. Pan, "Enhancing the High-Voltage Cycling Performance of LiNi0.5Mn0.3Co0.2O2by Retarding Its Interfacial Reaction with an Electrolyte by Atomic-Layer-Deposited Al2O3," *ACS Applied Materials and Interfaces*, vol. 7, no. 45, pp. 25 105 – 25 112, 2015.

[70] M. Anderman, "Assessing the Future of Hybrid and Electric Vehicles: The xEV Industry Insider Report," *Advanced Automotive Batteries*, 2013.

[71] M. Barghamadi, A. Kapoor, and C. Wen, "A Review on Li-S Batteries as a High Efficiency Rechargeable Lithium Battery," *Journal of The Electrochemical Society*, vol. 160, no. 8, pp. A1256–A1263, 2013.

[72] P. Bruce, S. Freunberger, L. Hardwick, and J.-M. Tarascon, "Li-O2 and Li-S batteries with high energy storage," *Nature Materials*, vol. 11, no. 1, pp. 19 – 29, 2012.

[73] P. G. Bruce, L. J. Hardwick, and K. Abraham, "Lithium-air and lithium-sulfur batteries," *MRS Bulletin*, vol. 36, no. 7, pp. 506–512, 2011.

[74] L. Chen and L. L. Shaw, "Recent advances in lithium-sulfur batteries," *Journal of Power Sources*, vol. 267, pp. 770–783, 2014.

[75] F. Mizuno, S. Nakanishi, Y. Kotani, S. Yokoishi, and H. Iba, "Rechargeable Li-air batteries with carbonate-based liquid electrolytes," *Electrochemistry*, vol. 5, pp. 403–405, 2010.

[76] A. Kraytsberg and Y. Ein-Eli, "Review on Li-air batteries - Opportunities, limitations and perspective," *Journal of Power Sources*, vol. 196, no. 3, pp. 886–893, 2011.

[77] P. Johansson and R. Dominko, “EUROLIS - European lithium sulphur cells for automotive applications,” *EVS-27 Symposium*, pp. 1–3, 2013.

[78] X. Ji and L. F. Nazar, “Advances in LiS batteries,” *Journal of Materials Chemistry*, vol. 20, no. 44, p. 9821, 2010.

[79] H. Wang, Y. Yang, Y. Liang, J. T. Robinson, Y. Li, A. Jackson, Y. Cui, and H. Dai, “Graphene-wrapped sulfur particles as a rechargeable lithium-sulfur battery cathode material with high capacity and cycling stability,” *Nano Letters*, vol. 11, no. 7, pp. 2644–2647, 2011.

[80] S. Lu, Y. Cheng, X. Wu, and J. Liu, “Significantly improved long-cycle stability in high-rate Li-S batteries enabled by coaxial graphene wrapping over sulfur-coated carbon nanofibers,” *Nano Letters*, vol. 13, no. 6, pp. 2485–2489, 2013.

[81] R. Chen, T. Zhao, J. Lu, F. Wu, L. Li, J. Chen, G. Tan, Y. Ye, and K. Amine, “Graphene-based three-dimensional hierarchical sandwich-type architecture for high-performance Li/S batteries,” *Nano Letters*, vol. 13, no. 10, pp. 4642–4649, 2013.

[82] M. Zhao, X. Liu, Q. Zhang, G. Tian, J. Huang, W. Zhu, and F. Wei, “Graphene/single-walled carbon nanotube hybrids: One-step catalytic growth and applications for high-rate li-s batteries,” *ACS Nano*, vol. 6, no. 12, pp. 10 759 – 10 769, 2012.

[83] S. Evers and L. F. Nazar, “Graphene-enveloped sulfur in a one pot reaction: a cathode with good coulombic efficiency and high practical sulfur content,” *Chem Commun (Camb)*, vol. 48, no. 9, pp. 1233–1235, 2012.

[84] N. Liu, L. Hu, M. T. Mcdowell, A. Jackson, and Y. Cui, "Prelithiated Silicon Nanowires as an Anode for Lithium Ion Batteries," *Energy & Environmental Science*, no. 8, pp. 6487–6493, 2011.

[85] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A comparative study of observer design techniques for state of charge estimation in electric vehicles," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, pp. 102–107, Jul. 2012.

[86] D. S. Jung, T. H. Hwang, S. B. Park, and J. W. Choi, "Spray drying method for large-scale and high-performance silicon negative electrodes in Li-ion batteries," *Nano Letters*, vol. 13, no. 5, pp. 2092–2097, 2013.

[87] N. Liu, Z. Lu, J. Zhao, M. T. McDowell, H.-W. Lee, W. Zhao, and Y. Cui, "A pomegranate-inspired nanoscale design for large-volume-change lithium battery anodes." *Nature nanotechnology*, vol. 9, no. 3, pp. 187–92, 2014.

[88] A. Affanni, A. Bellini, G. Franceschini, P. Guglielmi, and C. Tassoni, "Battery choice and management for new-generation electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 52, pp. 1343–1349, 2005.

[89] S. M. Lukic, J. Cao, R. C. Bansal, F. Rodriguez, and A. Emadi, "Energy storage systems for automotive applications," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2258–2267, 2008.

[90] G. Berdichevsky, K. Kelty, J. B. Straubel, and E. Toomre, "The Tesla Roadster Battery System," Tesla Motors, Tech. Rep., 2006.

[91] D. H. Doughty and C. C. Crafts, “FreedomCAR Electrical Energy Storage System Abuse Test Manual for Electric and Hybrid Electric Vehicle Applications,” *United States Council for Automotive Research LLC (USCAR)*, pp. 3–48, 2005.

[92] Maxwell, “K2 ultracapacitors - 2.7v series,” 2014, document number: 1015370.5.

[93] J. E. Corporation, “Lithium ion capacitor ultimo,” 2014.

[94] J. Cao, N. Schofield, and A. Emadi, “Battery balancing methods: A comprehensive review,” in *Vehicle Power and Propulsion Conference (VPPC)*, 2008.

[95] N. H. Kutkut, H. L. N. Wiegman, D. M. Divan, and D. W. Novotny, “Charge equalization for an electric vehicle battery system,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 235–246, 1998.

[96] N. H. Kutkut, H. L. N. Wiegman, D. M. Divan, and D. W. Novotny, “Design considerations for charge equalization of an electric vehicle battery system,” *IEEE Transactions on Industry Applications*, vol. 35, pp. 28–35, 1999.

[97] A. J. Bard and L. R. Faulkner, *Electrochemical Methods: Fundementals and Applications*, 2nd ed. New York: Wiley, 2001.

[98] R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, and S. Habibi, “Reduced-Order Electrochemical Model Parameters Identification and State of Charge Estimation for Healthy and Aged Li-Ion Batteries - Part II: Aged Battery Model and State of Charge Estimation,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 678–690, 2014.

[99] S. E. Samadani, R. Fraser, and M. Fowler, "A Review Study of Methods for Lithium-ion Battery Health Monitoring and Remaining Life Estimation in Hybrid Electric Vehicles," *SAE International*, 2012.

[100] M. Chen and G. A. Rincón-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 504–511, 2006.

[101] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1. Background," *Journal of Power Sources*, vol. 134, pp. 252–261, 2004.

[102] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2. Modeling and identification," *Journal of Power Sources*, vol. 161, no. 2, pp. 1356–1368, 2004.

[103] E. Chemali, L. McCurlie, B. Howey, T. Stiene, M. M. Rahman, M. Preindl, R. Ahmed, and A. Emadi, "Minimizing battery wear in a hybrid energy storage system using a linear quadratic regulator," in *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2015, accepted.

[104] S. Buller, M. Thele, R. De Doncker, and E. Karden, "Impedance-based simulation models of supercapacitors and li-ion batteries for power electronic applications," in *Industry Applications Conference, 2003. 38th IAS Annual Meeting. Conference Record of the*, vol. 3, pp. 1596–1600 vol.3, Oct. 2003.

[105] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth, and D. Sauer, "Characterization of high-power lithium-ion batteries by electrochemical impedance

spectroscopy. ii: Modelling," *Journal of Power Sources*, vol. 196, no. 12, pp. 5349–5356, 2011.

[106] G. L. Plett, "Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 3. state and parameter estimation," *Journal of Power sources*, vol. 134, no. 2, pp. 277–292, 2004.

[107] K. A. Smith, "Electrochemical modeling, estimation and control of lithium ion batteries," 2006.

[108] K. A. Smith, C. D. Rahn, and C.-Y. Wang, "Control oriented 1d electrochemical model of lithium ion battery," *Energy Conversion and management*, vol. 48, no. 9, pp. 2565–2578, 2007.

[109] B. Kirby, *Micro- and Nanoscale Fluid Mechanics : Transport in Microfluidic Devices.* Cambridge: Cambridge University Press, 2010.

[110] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for advanced battery-management systems," *IEEE Control Systems*, vol. 30, no. 3, pp. 49–68, 2010.

[111] I. Snihir, W. Rey, E. Verbitskiy, A. Belfadhel-Ayeb, and P. Notten, "Battery open-circuit voltage estimation by a method of statistical analysis," *Journal of Power Sources*, vol. 159, no. 2, pp. 1484 – 7, 22 Sept. 2006.

[112] B. Bole, C. S. Kulkarni, and M. Daigle, "Randomized battery usage data set," *NASA Ames Prognostics Data Repository*, 2014.

[113] F. Codeca, S. Savaresi, and V. Manzoni, "The mix estimation algorithm for battery State-of-Charge estimator- Analysis of the sensitivity to measurement

errors." in *Proceedings of 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC)*, pp. 8083–8088, Dec. 2009.

[114] F. Codeca, S. Savaresi, and G. Rizzoni, "On battery state of charge estimation: A new mixed algorithm," in *Proceedings of 17th IEEE International Conference on Control Applications (CCA)*, pp. 102–107, Sep. 2008.

[115] M. A. Roscher, "Zustandserkennung von LiFePO4-Batterien fr Hybrid- und Elektrofahrzeuge," Ph.D. dissertation, RWTH Aachen University, 2010.

[116] F. Baronti, G. Fantechi, L. Fanucci, E. Leonardi, R. Roncella, R. Saletti, and S. Saponara, "State-of-charge estimation enhancing of lithium batteries through a temperature-dependent cell model," in *Proceedings of 2011 International Conference on Applied Electronics (AE)*, pp. 1–5, Sep. 2011.

[117] M. Roscher, O. Bohlen, and D. Sauer, "Reliable state estimation of multicell lithium-ion battery systems," *IEEE Transactions on Energy Conversion*, vol. 26, no. 3, pp. 737–743, Sep. 2011.

[118] T. Kim, W. Qiao, and L. Qu, "Real-time state of charge and electrical impedance estimation for lithium-ion batteries based on a hybrid battery model," in *Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE*, pp. 563–568, Mar. 2013.

[119] T. Kim, W. Qiao, and L. Qu, "Online state of charge and electrical impedance estimation for multicell lithium-ion batteries," in *Transportation Electrification Conference and Expo (ITEC), 2013 IEEE*, pp. 1–6, Jun. 2013.

[120] X. Hu, F. Sun, and Y. Zou, "Estimation of state of charge of a lithium-ion battery pack for electric vehicles using an adaptive luenberger observer," *Energies*, vol. 3, no. 9, pp. 1586 – 1603, 2010.

[121] H. Chaoui and P. Sicard, "Accurate state of charge (SOC) estimation for batteries using a reduced-order observer," in *2011 IEEE International Conference on Industrial Technology (ICIT)*, pp. 39–43, Mar. 2011.

[122] Y.-H. Chiang, W.-Y. Sean, and J.-C. Ke, "Online estimation of internal resistance and open-circuit voltage of lithium-ion batteries in electric vehicles," *Journal of Power Sources*, vol. 196, no. 8, pp. 3921 – 3932, 2011.

[123] I.-S. Kim, "The novel state of charge estimation method for lithium battery using sliding mode observer," *Journal of Power Sources*, vol. 163, no. 1, pp. 584 – 590, 2006.

[124] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A comparative study of observer design techniques for state of charge estimation in electric vehicles," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, pp. 102–107, Jul. 2012.

[125] F. Zhang, G. Liu, and L. Fang, "A battery state of charge estimation method using sliding mode observer," in *2008 7th World Congress on Intelligent Control and Automation*, pp. 989 – 94, 2008.

[126] X. Chen, W. Shen, Z. Cao, A. Kapoor, and I. Hijazin, "Adaptive gain sliding

mode observer for state of charge estimation based on combined battery equivalent circuit model in electric vehicles," in *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, pp. 601–606, Jun. 2013.

[127] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 3. State and parameter estimation," *Journal of Power Sources*, vol. 134, no. 2, pp. 277–292, 2004.

[128] W. Wang, D. Wang, X. Wang, T. Li, R. Ahmed, S. Habibi, and A. Emadi, "Comparison of kalman filter-based state of charge estimation strategies for li-ion batteries," in *Transportation Electrification Conference and Expo (ITEC), 2016 IEEE*, pp. 1–6. IEEE, 2016.

[129] M. Charkhgard and M. Farrokhi, "State-of-charge estimation for lithium-ion batteries using neural networks and ekf," *IEEE transactions on industrial electronics*, vol. 57, no. 12, pp. 4178–4187, 2010.

[130] J. Du, Z. Liu, and Y. Wang, "State of charge estimation for li-ion battery based on model from extreme learning machine," *Control Engineering Practice*, vol. 26, pp. 11–19, 2014.

[131] J. Meng, G. Luo, and F. Gao, "Lithium polymer battery state-of-charge estimation based on adaptive unscented kalman filter and support vector machine," *IEEE Transactions on Power Electronics*, vol. 31, no. 3, pp. 2226–2238, 2016.

[132] S. Tong, J. H. Lacap, and J. W. Park, "Battery state of charge estimation using a load-classifying neural network," *Journal of Energy Storage*, vol. 7, no. Supplement C, pp. 236 – 243, 2016.

[133] X. Dang, L. Yan, K. Xu, X. Wu, H. Jiang, and H. Sun, "Open-circuit voltage-based state of charge estimation of lithium-ion battery using dual neural network fusion battery model," *Electrochimica Acta*, vol. 188, pp. 356–366, 2016.

[134] J. Remmlinger, M. Buchholz, M. Meiler, P. Bernreuter, and K. Dietmayer, "State-of-health monitoring of lithium-ion batteries in electric vehicles by on-board internal resistance estimation," *Journal of Power Sources*, vol. 196, no. 12, pp. 5357–5363, 2011.

[135] Y.-H. Chiang, W.-Y. Sean, and J.-C. Ke, "Online estimation of internal resistance and open-circuit voltage of lithium-ion batteries in electric vehicles," *Journal of Power Sources*, vol. 196, no. 8, pp. 3921–3932, 2011.

[136] X. Feng, J. Li, M. Ouyang, L. Lu, J. Li, and X. He, "Using probability density function to evaluate the state of health of lithium-ion batteries," *Journal of Power Sources*, vol. 232, pp. 209–218, 2013.

[137] C. Weng, Y. Cui, J. Sun, and H. Peng, "On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression," *Journal of Power Sources*, vol. 235, pp. 36–44, 2013.

[138] Z. Guo, X. Qiu, G. Hou, B. Y. Liaw, and C. Zhang, "State of health estimation for lithium ion batteries based on charging curves," *Journal of Power Sources*, vol. 249, pp. 457–462, 2014.

[139] I. S. Kim, "A technique for estimating the state of health of lithium batteries through a dual-sliding-mode observer," *IEEE Transactions on Power Electronics*, vol. 25, no. 4, pp. 1013–1022, Apr. 2010.

[140] J. Remmlinger, M. Buchholz, T. Soczka-Guth, and K. Dietmayer, "On-board state-of-health monitoring of lithium-ion batteries using linear parameter-varying models," *Journal of Power Sources*, vol. 239, pp. 689 – 695, 2013.

[141] A. Eddahech, O. Briat, N. Bertrand, J.-Y. Delétage, and J.-M. Vinassa, "Behavior and state-of-health monitoring of li-ion batteries using impedance spectroscopy and recurrent neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 42, no. 1, pp. 487–494, 2012.

[142] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, and K. Dietmayer, "Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods," *Journal of Power Sources*, vol. 239, pp. 680–688, 2013.

[143] Y. Zhang, R. Xiong, H. He, and Z. Liu, "A lstm-rnn method for the lithuim-ion battery remaining useful life prediction," in *Prognostics and System Health Management Conference (PHM-Harbin), 2017*, pp. 1–4. IEEE, 2017.

[144] Z. Chen, C. C. Mi, Y. Fu, J. Xu, and X. Gong, "Online battery state of health estimation based on genetic algorithm for electric and hybrid vehicle applications," *Journal of Power Sources*, vol. 240, pp. 184–192, 2013.

[145] D. Andre, A. Nuhic, T. Soczka-Guth, and D. U. Sauer, "Comparative study of a structured neural network and an extended kalman filter for state of health determination of lithium-ion batteries in hybrid electricvehicles," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 3, pp. 951–961, 2013.

[146] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[147] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[148] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Arxiv.Org*, vol. 7, no. 3, pp. 171–180, 2015.

[149] L. Deng, "Deep Learning: Methods and Applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.

[150] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[151] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[152] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structureactivity relationships," *Journal of Chemical Information and Modeling*, vol. 55, no. 2, pp. 263–274, 2015.

[153] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep Learning for Identifying Metastatic Breast Cancer," *arXiv preprint*, pp. 1–6, 2016.

[154] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. url h ttp," *Software available from tensorflow.org*.

[155] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[156] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, no. 3, pp. 6645–6649, 2013.

[157] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," *ArXiv e-prints*, Dec. 2015.

[158] I. Sutskever, "Training Recurrent neural Networks," Ph.D. dissertation, University of Toronto, 2013.

[159] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." *30th International Conference on Machine Learning (ICML)*, vol. 28, pp. 1310–1318, 2013.

[160] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, 1998.

[161] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[162] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv preprint arXiv:1409.2329*, 2014.

[163] D. Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649, 2012.

[164] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-regulated splicing code,” *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.

[165] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes *et al.*, “The human splicing code reveals new insights into the genetic determinants of disease,” *Science*, vol. 347, no. 6218, p. 1254806, 2015.

[166] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning,” *Nature communications*, vol. 5, 2014.

[167] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[168] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Networks*, vol. 16, no. 10, pp. 1429–1451, 2003.

[169] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[170] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[171] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?" *Journal of Statistical Physics*, vol. 168, pp. 1223–1247, Sep. 2017.

[172] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[173] G. F. Montúfar and J. Morton, "When does a mixture of products contain a product of mixtures?" *SIAM Journal on Discrete Mathematics*, vol. 29, no. 1, pp. 321–347, 2015.

[174] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, pp. 2924–2932, 2014.

[175] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[176] A. Fotouhi, D. J. Auger, K. Propp, S. Longo, and M. Wild, "A review on electric vehicle battery modelling: From lithium-ion toward lithium–sulphur," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1008–1021, 2016.

[177] Y. Cao, R. C. Kroeze, and P. T. Krein, "Multi-timescale parametric electrical battery model for use in dynamic electric vehicle simulations," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 4, pp. 432–442, 2016.

[178] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[179] O. Delalleau and Y. Bengio, "Shallow vs. deep sum-product networks," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., pp. 666–674. Curran Associates, Inc., 2011.

[180] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, pp. 2924–2932, 2014.

[181] L. Holmstrom and P. Koistinen, "Using additive noise in back-propagation training," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 24–38, 1992.

[182] P. Koistinen and L. Holmström, "Kernel regression and backpropagation training with noise," in *Advances in Neural Information Processing Systems*, pp. 1033–1039, 1992.

[183] G. An, “The effects of adding noise during backpropagation training on a generalization performance,” *Neural computation*, vol. 8, no. 3, pp. 643–674, 1996.

[184] W. He, N. Williard, C. Chen, and M. Pecht, “State of charge estimation for li-ion batteries using neural network modeling and unscented kalman filter-based error cancellation,” *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 783–791, 2014.

[185] Panasonic, “Panasonic NCR18650PF Lithium-Ion Battery Datasheet (June 2016),” 2016.

[186] Panasonic, “Introduction of NCR18650PF, Panasonic,” 2013.

[187] P. J. Kollmeyer, “Development and Implementation of a Battery-Electric Light-Duty Class 2a Truck including Hybrid Energy Storage,” Ph.D. dissertation, The University of Wisconsin - Madison, 2015.

[188] P. J. Kollmeyer, W. Lamb, L. W. Juang, J. D. McFarland, T. M. Jahns, and B. Sarlioglu, “Design of an electric powertrain for a ford f150 crew cab truck utilizing a lithium battery pack and an interior pm synchronous machine drive,” in *2012 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–8, 2012.

[189] Y. S. Lee, W. Y. Wang, and T. Y. Kuo, “Soft computing for battery state-of-charge (bsoc) estimation in battery string systems,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 229–239, 2008.

[190] W.-Y. 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.