University of Hertfordshire UH

School of Physics,
Engineering and
Computer Science

MSc Data Science Project
7PAM2002-0901-2024
Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Simultaneous Forecasting & Anomaly Detection in Battery Performance of EVs Using Neural Networks

Student Name and SRN:

Muhammad Shoaib Manzoor 22038495

Supervisor: Dr Ashley Spindler

Date Submitted: 6th January 2025

Word Count: 5391

GitHub address: https://github.com/MShoaibManzoor/Advanced-Research

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6).

I have not used chatGPT, or any other generative AI tool, to write the report or code (other than where declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student SRN number: 22038495

Student Name printed: Muhammad Shoaib Manzoor

Student signature: Muhammad Shoaib Manzoor

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Acknowledgements

I would like to express my deepest gratitude to all those who have supported me throughout the process of completing this project. First & foremost, Dr Ashley Spindler, for her invaluable guidance, encouragement, and insightful feedback throughout the course of this project. Her expertise and support were crucial to the completion of this project. I would also like to extend my thanks to our module coordinator, Carolyn Devereux, for her detailed & engaging lectures, etching the pathway to a fruitful research and learning experience, A special appreciation goes profs Man-Tang Lai & Mykola Gordovskyy for their detailed lectures in Data Handling & Visualization and Machine Learning which paved the way for a comprehensive research.

Abstract

This study focuses on merging feature extraction from two different implementations of ANNs, namely LSTM & Autoencoders, to simultaneously forecast state of charge used and detect anomalies in charging behaviours of electric vehicles (EVs), respectively. The models have been trained separately to achieve the lowest value for the Huber loss function, found to be feasible for datasets with a large number of outliers, and a varying set of hyperparameters specific to the models, for instance, variable learning rates, layer orders, types of layers, nodes per layer, recurrent dropout rates in LSTM layers, size of the bottlenecks in coding layers of the auto encoders etc. The datasets employed are two pronged. Firstly, a vehicle's state of charge usage during a trip & secondly the accumulation of charge in the battery while plugged in for charging and features contributing to the charge usage & storage efficiency of the battery. The models were first trained independently on their respective datasets, LSTM on trip data and Autoencoder on charging data. The LSTM forecasting model was tuned to yield a loss function value of 0.038 & mean absolute error value of 0.0799 within the min and max range values of SOC being 0 & 1 scaled with the MinMaxScaler when trained independently. The Autoencoder anomaly detection model when trained independently yielded a loss value of 0.0235 with mean absolute error value of 0.1561 within range of 0 & 1 for SOC accumulated till a given time stamp. Finally, the two architectures are combined in a separate model using the shared dense layers to learn feature representations of both the problem contexts revealing a high level of convergence with significant increase in resource consumption. The resultant of this combination provides a robust loss value of 0.0140 using the MSE as loss function with its counterpart MAE being 0.0910.

Table of Contents

1. Introduction

With the minimization of space occupied by technology and every new gadget being simplified in terms of size and optimized for maximum storage and computational efficiency, focus turns to the operational efficiency of the product. For this purpose, the mechanisms driving these operations are expected to be compact in architecture and comparatively less demanding in terms of computational resources with respect to the solution, for clarity, the resource required by the processes will always be less when solved individually, the simultaneous approach must solve the problem to acceptable satisfaction without taking more resources compared to individual operations conducted in a series. The most efficient methodology is found to be multi-tasking or simultaneous operation, which is currently, however, far from perfect. This study is aimed at conducting an in-depth analysis of the said methodology. Artificial neural networks possess a very malleable architecture, the more their structure is refined, the better the results. Amongst a variety of architectures, the two chosen for this study propose a complex approach to the context. The primary objective is to produce a resultant model that can learn features from two different datasets that have a commonality unspecified to the models explicitly, which can be interpreted by the natural neural network, thus the need for simultaneous processing to identify learnable patterns in two datasets that are relevant to each other implicitly. The benchmark for this process is that the combined model produces results that are a close representative of the data produced by models individually, the computational resource and the complexity of the combined model justifies the need for the process and lastly the resultant values of the combined model possess a similar representation of the data being fed to the models individually. The two datasets employed for the purpose resonate through each other via human understandable concepts of Charging and discharging behaviours of EVs, the implementation of which requires transferable knowledge between the behaviours for the model to understand. While abnormal behaviours of batteries can be observed when there is an out-of-bound occurrence in charging patterns, but the human understanding model cannot predict these behaviours without having knowledge of prior battery usage or charging patterns or perhaps relevant domain knowledge.

1.1 Background

Multi-task ML models have been developed for a range of purposes, one of the observed methodologies is the introduction of errors generated by a Generative Adversarial Network (SMOTE-GAN) through oversampling structure for data balancing, into an Autoencoder model for data reconstruction and anomaly detection. [1] Which was applied to a analyse network traffic and detect normal and attack traffics, greatly reducing the shortcomings of the AE alone, to allow Intrusion Detection Systems to identify rare attacks and introduce robustness. The hybrid resampling process combining SMOTE & GAN increase proportion of rare attacks and improve class borders and a hybrid deep learning model consisting of two stages; AE-based anomaly detection which learns the feature representation and an MTL-based label classification with subnets for each type of network traffic. Another detailed study conducted upon multi-task convolutional DNNs for recommendation based on knowledge graphs [2], suggests the use of 5 components, responsible for Convolution Interaction (CI), User preference, Entity Relation Importance, Recommendation and Knowledge graph embedding. The self-explanatory architecture allows each component to deal with a separate aspect of the dataset and feeds into the relevant component of the model to generate well informed recommendations for users. Multi-task learning application on Electricity Load Forecasting conducted in 2023 [3] also shares architectural similarities with this study, the model features are shared between a main task, power load on, and an auxiliary task weather and time series features to predict short and medium term power loads, the architecture consists of an LSTM model and a convolutional neural network, consisting of hidden layers shared by all tasks to capture the intrinsic joint features through hard parameter sharing as well as where the models are isolated using distance regularization to constrain parameters between models and guarantee similarity of parameter space. Application of multi-task learning for gradient conflicts suggested by Chen J in 2025 [4], suggested a novel approach to employing multi-task models, the architecture was composed of an expert squad layer, a module that encapsulates the learning of both task-specific knowledge and shared features, with a novel backbone used at the foundation level, tailored specifically for multi-task learning applications. This approach involves partitioning of task-specific & shared channels, where the task specific channels are processed by dedicated network experts, responsible for distilling task specific knowledge. Meanwhile the shared features across all tasks are captured through pointwise soft aggregation mechanism drawing upon the output of all experts.

Study	Key Findings	Relevance
Multi-task learning for IoT traffic classification	AE & MTL-classification multi-task model with oversampling introduction through SMOTE & GAN to increase proportion of anomalies.[1]	Suggests a refined architecture for AE & MTL embedding. Relevant to the AE part of the proposed model by error reconstruction in network traffic.
Multi-task convolutional deep neural network for recommendation based on knowledge graphs	Implies component-based architecture with specific tasks for each. Combines resultant values via the interaction component.[2]	Linear structure that combines two features' representations out of four, relevant in terms of feature sharing but does not include complete context sharing like the proposed model.
A CNN and LSTM-based multi-task learning architecture for short and medium-term electricity load forecasting	Delegates primary and auxiliary tasks to two different architectures with different feature inputs.[3]	Shares architectural and input similarity to proposed model in terms of layering, feature sharing and output.
Mitigating gradient conflicts via expert squads in multi-task learning	Tasks delegated to separate experts with soft aggregation introducing feature sharing.[4]	Relevant in terms of inputs provided to specific layers with a different approach to feature sharing.

Table 1.1: Break down of included studies including their key findings and their relevance.

2. Dataset

The datasets employed for the study contain performance data from electric vehicles during their standard operations i.e. during a trip or plugged in for charging. It was collected by Calstart & U.S Department of energy to provide data for medium and heavy-duty battery electric vehicles in the United States. The dataset includes a total of 11 manufacturing companies out of which Proterra was selected due to a lesser number of missing values compared to the other companies. The data was collected by onboard sensors and data loggers (including HEM, ViriCiti & Geotab). Each row of the "Trip Data" contains the data collected during a distance travelled or idling time of the vehicle representing the variables leading to the charge being consumed from the battery. The sensors recorded a wide range of features involved in the vehicle's operation such as the average ambient temperature during the duration of trip in hours derived from the local trip start and end time stamps recorded in each trip. Energy efficiency is an engineered feature, derived from the distance travelled and energy consumed by the vehicle.

Name	Shape Before	Shape After	%age Retained
Vehicle Data	(24108, 24)	(8013, 9)	33.24%
Charging Data	(11859, 18)	(6296, 8)	53.09%

A major portion of the data was lost due to null values & irrelevance to the study, further explanation of the data characteristics is as below.

2.1 Column Analysis

2.1.1 Vehicle Data

For time information there were 3 available columns, namely Local Trip Start Time, Local Trip End Time & Date. The duration of the trip was acquired using start & end times of the trip after which the End Time was retained to signify the feature impacts on target by the end of the trip, to be used for time series analysis, the date column was removed due to large number of missing values. The following table specifies which columns were available against their relevance.

Feature Name	Relevance
🟩 **Initial SOC**	To observe SOC levels over time
🟩 **Final SOC**	To observe the drop in SOC at the end of Trip
🟩 **SOC Used**	The charge used by the end of trip
🟩 **Total Energy Consumption**	Energy consumed for insight into usage patterns
🟩 **Total Distance**	SOC usage w.r.t distance travelled
🟩 **Average Ambient Temperature**	Impact of temperature on SOC consumption
🟥 **Percent Idling Time**	% of total time vehicle was static with running engine
🟥 **Average Speed**	For speed fluctuations during trip
🟥 **Energy consumed while idling**	Charge consumed during static running
🟥 **Energy consumed while driving**	Charge consumed while in motion

Table 2.1: Shows the features available and their relevance to study, Green: retained, Red: dropped.

Percentage idling time would be beneficial to determine the low state of charge used during trips with longer durations, as Trip Duration would not have a direct effect on the consumption of charge this would benefit the model in learning patterns related to SOC Usage and Duration which can yet be retained using Total Distance & can be substituted for Energy Consumption while driving or idling, as the distance covered would reflect SOC usage instead. Average

Speed also has an impact on SOC usage but could not be retained due to an availability of ~25% available values. Other important features included Total Runtime of the vehicle which could give provide information of the vehicle's previous

2.1.2 Charging Data

Charging data followed the same pattern as trip data in terms of time values, Local Charge End Time was retained to observe SOC accumulations by the end of each charging session. The other features found in the dataset to be impactful in model training with respect to the topic relevance are as mentioned in the table.

Feature Name	Relevance
Starting SOC	To observe SOC levels charged over time
Ending SOC	To observe accumulated SOC at the end of charging
SOC Charged	Amount of SOC accumulated
Total Energy Delivered	Relevant to charging efficiency
Average Power	For power levels during charging session
Max Power	Max power used by charger during session

Table 2.2: The relevant features in charging dataset.

Other features included in the dataset were found to be derivatives of the above listed features, e.g. Local Connect and Disconnect times, which could be useful to acquire charging duration of the session, this was later engineered using the Charge Start & End times.

2.2 Exploratory Data Analysis

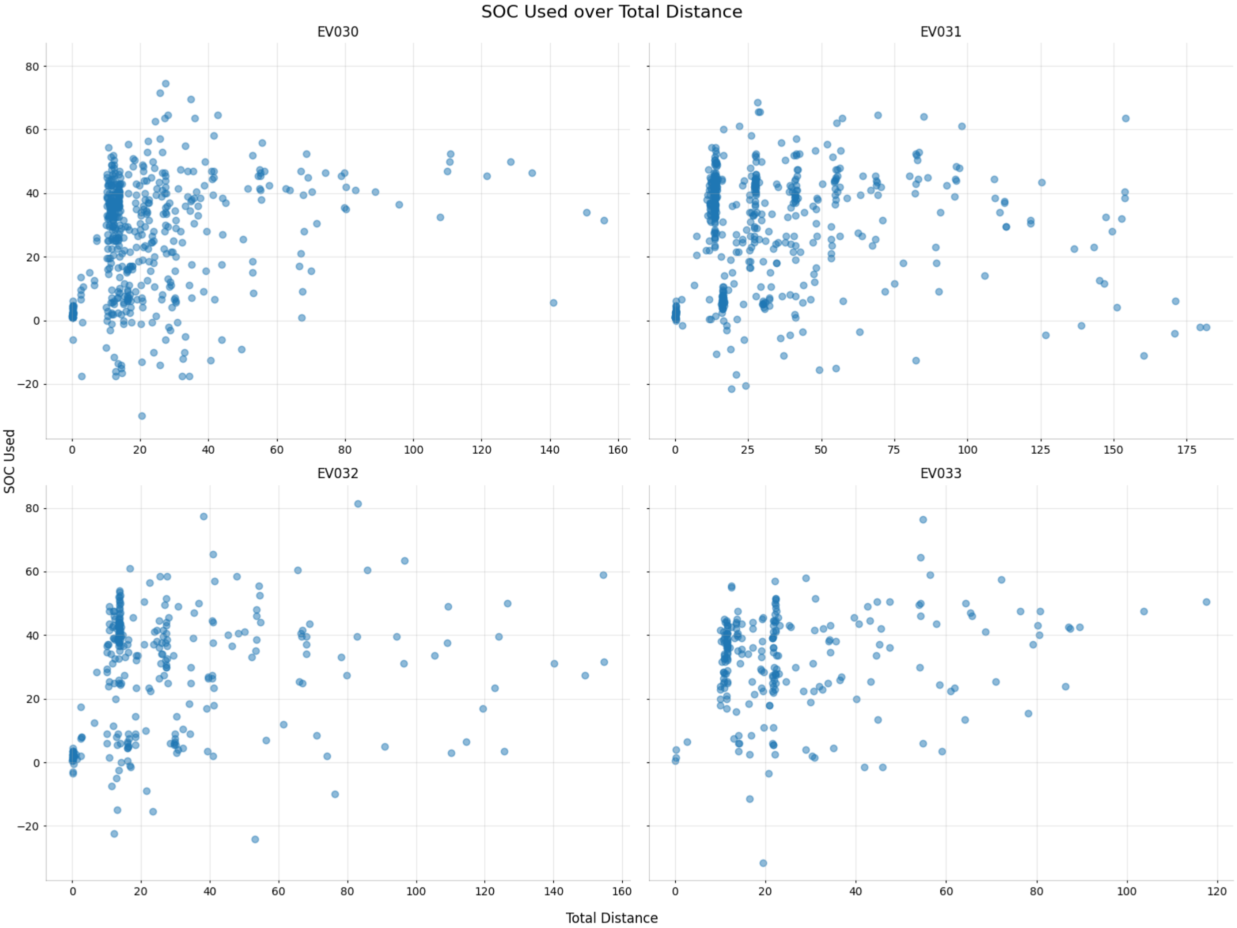


Figure 2.2.1. Shows the fluctuations in SOC w.r.t Total Distance revealing data skewness to short trips.

It was revealed during EDA that most records contain short distances travelled, this introduces a skewness in fitting towards shorter trips and making model application more robust for usage with abrupt changes in a shorter span of charge cycles, shown in figure 2.2.1.

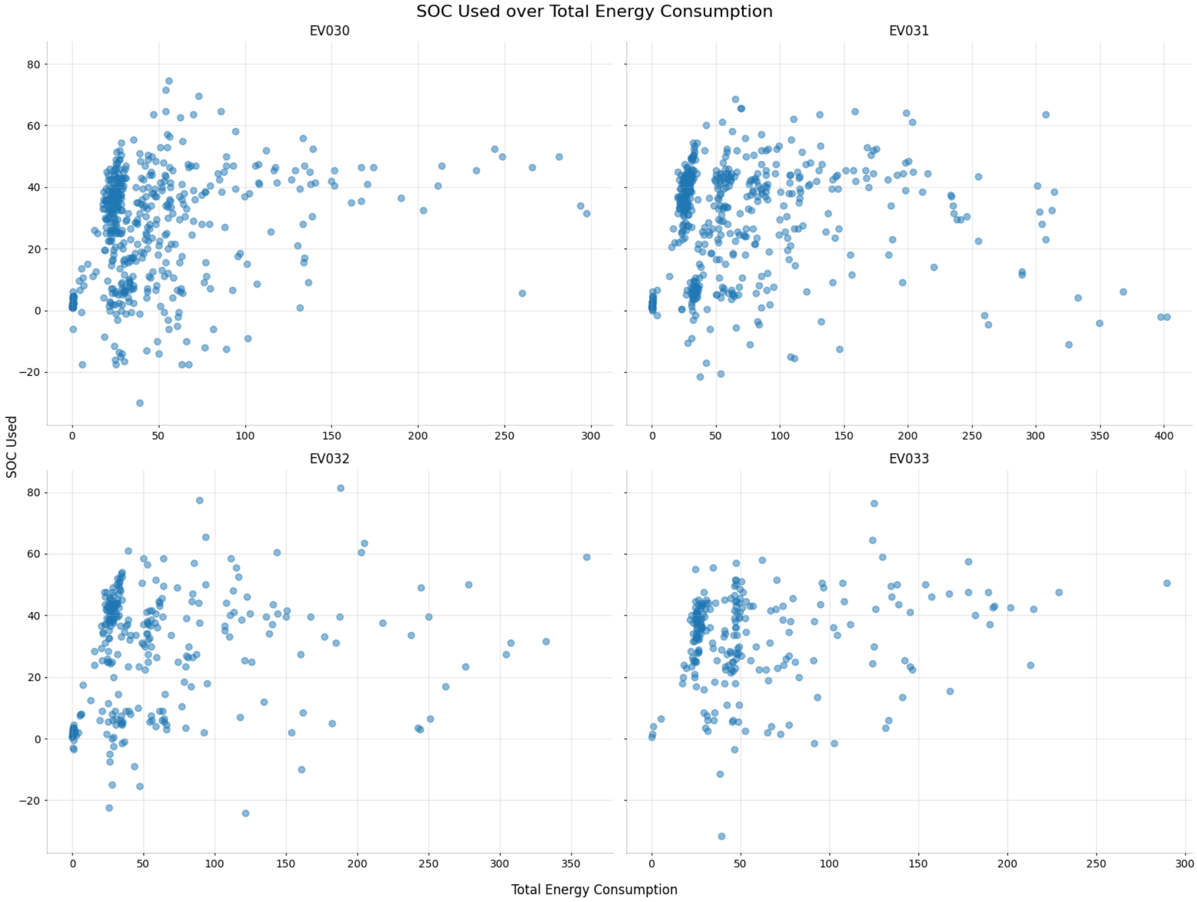


Figure 2.2.2: Effect of energy consumption on SOC Used.

#Due to the skewness of data points, it is difficult to observe the effect of energy consumption on high SOC usage, there however a directly proportional relation observable with a minute effect, as shown in figure 2.2.2.

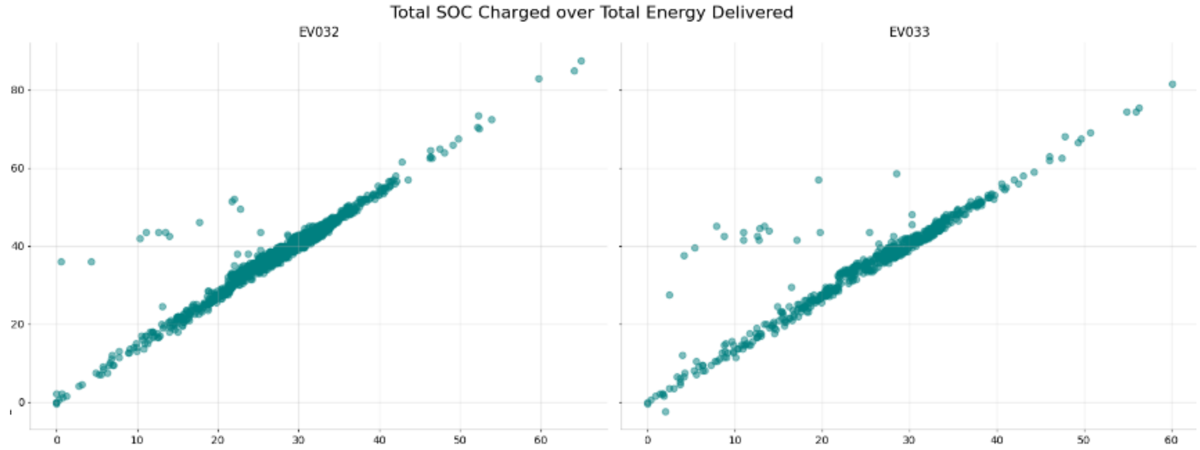


Figure 2.2.3: Directly proportional relation between SOC Charged & Energy Consumption.

For Charging Data, energy delivered has a strong linear relation to charge accumulated with a few of the sessions generating more charge upon less Energy delivered, which might be considered anomalous by the model.

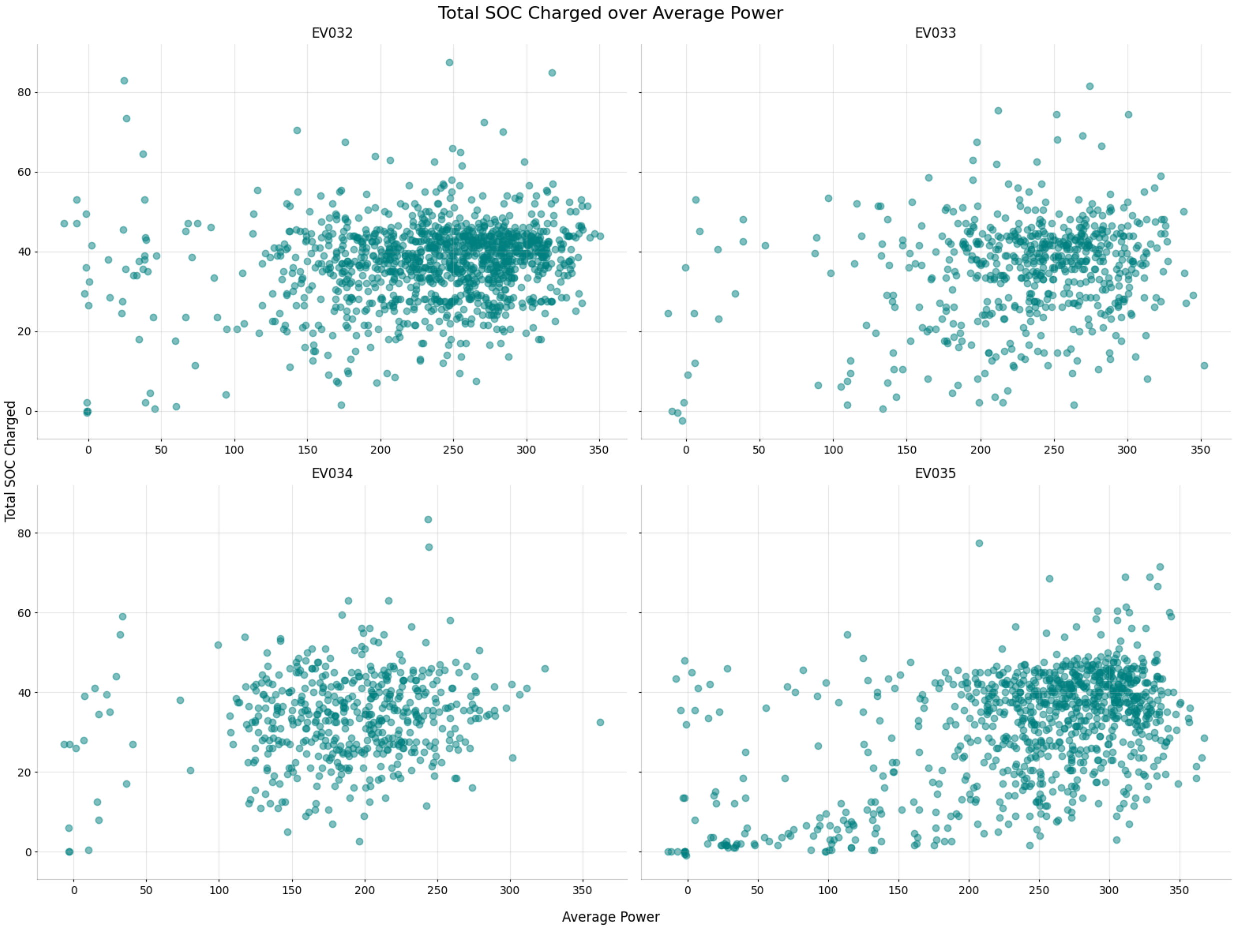


Figure 2.2.4: Average power related to State of Charge, no definitive relationship.

Furthermore, Average Power delivered by chargers does not seem to have a definitive relationship to charge accumulation, as evident from figure 2.2.3, as average power is responsible for the rate of energy delivered and should be directly proportional to the SOC accumulated the dataset does not provide that relation for the model, more power deliver can be a factor in battery degradation as well, neither relation is observable as in fig 2.2.4.

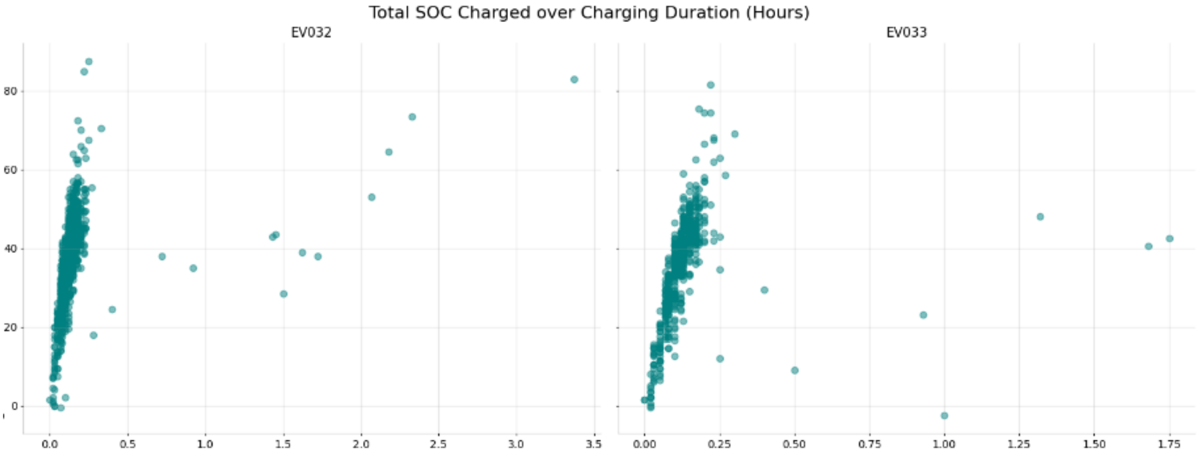


Figure 2.2.5: Charging Durations Vs. SOC Charged, representing quick charging employed.

The charging behaviours of observed in fig 2.2.5 indicate the usage of fast charging for most of the vehicles as short durations portray a greater charge accumulation with several anomalies suggesting use of slow charging or battery safeguards to protect against damages caused due to fast charging.

Link to the datasets:

https://livewire.energy.gov/ds/calstart/vehicle
https://livewire.energy.gov/ds/calstart/charging

2.3 Ethical Considerations

The datasets employed in this study have been anonymised during collection by introducing IDs for records directly identifying vehicles, no information is included in the dataset regarding direct identification of the subject vehicles & charging bays. There were, however, possibilities of the information being sensitive if there were locations provided for charging bays and Vehicle's GPS trackers, as other data was logged via onboard data loggers, which were not included in collection. No personal data was involved in the employed datasets and is publicly available on the links provided in the dataset section of this study. This is a government-based data collection project for the purpose of detailed research into providing affordable efficient, safe and accessible transportation future where mobility is decoupled from energy consumption.

More information regarding the collecting body and the platform employed can be found in the link: https://livewire.energy.gov/about

3. Methodology

The data contained trip & charging data from 11 companies all present in separate excel files, firstly, I fetched and combined the files into two Pandas data frames, one containing the trip data and the other the charging data. The number of records and attributes was reduced based on the number of missing rows, all rows containing missing values were dropped to retain a total of 8013 rows for Trip Data and 6296 total for Charging Data. As it would be feasible to merge the data frames to be combined on the common vehicle ID column, I first tried to merge the datasets with Vehicle ID and Local Times of Charging & trip, this resulted in a row explosion due to the multiplication of cardinality with common values in the mentioned attributes. So I decided to keep the datasets separate and feed them to the models individually and train the LSTM model on Trip data for forecasting and the Charging data on Autoencoder for Anomaly detection, combining the models later on would retain the vehicle ID feature from the two and columns and significance of other features to predict the target variable in Trip data “SOC Used” and “Total SOC Charged” in the Charging data.

In the preprocessing step, I engineered a few attributes from the available data as in the given table. After which the Local Start Times were dropped from both the datasets and the End Times were set as index to indicate the time series.

Attribute Name	Formula	Dataset
Energy Efficiency	Distance / Energy Delivered	Trip Data
Trip Duration (Hours)	(Local Trip End Time – Local Trip Start Time) / 3600	Trip Data
Total SOC Charged	Final SOC – Initial SOC	Charging Data
Charge Duration (Hours)	(Local Charge End Time – Local Charge Start Time) / 3600	Charging Data

Table 3.1: Shows the features engineered from available dataset.

Outliers were removed from the datasets using Inter Quartile Range (IQR) filtering, which yielded the following number of rows for each set.

Dataset	Rows	Rows After Filtering
Trip Data	8009	8000
Charge Data	6296	6033

Table 3.2: Shows the number of rows retained after IQR filtering.

The reason for selecting IQR filtering was its ability to tighten the bounds using the IQR multiplier which is set to Tukey’s value of $\pm 1.5 * IQR$ by default but can be reduced or increased to add or remove leniency for outliers. The number of abrupt outliers in the sets causes the loss functions to result in major fluctuations.

The numerical columns are then normalized using the MinMaxScaler. Which scales the dominant features to be treated equally, namely ‘Average Power’ with mean value of 234.63 and ‘Charging Duration’ with a mean value of 0.12. Similarly, the Trip Data contains ‘Total Energy Consumption’ (mean = 55.36, std = 84.82) & ‘Energy Efficiency’ (mean = 3.17, max = 425.81) that contain extreme value variations. After using Label Binarizer to convert the ‘Vehicle ID’ into binary columns, both the datasets are converted to sequences using the

CreateSequences() function. The length of sequences is the standard value of 5, which minimizes model underfitting due to uneven timestamps.

The study is focused on three different architectures of ANNs, the Long-Short Term Memory NNs, The Autoencoders and the Combined model that uses LSTM & Autoencoders simultaneously.

3.1 LSTM Recurrent Neural Network

Designed for the purpose of forecasting State of charge, The LSTM model architecture is designed to extract the maximum number of features in the first layer with 64 units, the recurrent dropout is added to preserve the dropout mask with a rate of 0.2, reducing reliance on any single connection by dropping 20% of recurrent connections within the recurrent state. BatchNormalization is introduced to normalize outputs of LSTM layer improving training stability & acceleration. Another dropout layer is added to reduce the number of connections for the next layer. A dense fully connected layer added after dropout introduces room for new connections with a reduced number of units (32), a kernel_regularizer with L2 normalization & ‘relu’ activation, to improve convergence for the following LSTM layer with 32 units for matching convergence with the previous layer, a recurrent drop out of 10% and a low standard drop out for outputs. The following layer consists of 16 dense units to filter the extracted feature weights further and provided to a linear output of 1 unit.

The model is compiled with the ‘Adam’ optimizer due to its robustness. Learning rate was selected in contrast to the decrease in loss values after each epoch a set of different learning rates were tried before the selection of 0.0001 which yielded optimal training time and convergence of model. The loss functions tested for the model were Mean Absolute Error, Mean Squared Error & Huber loss function, from which Huber loss was selected due to the coefficient of sensitivity for switching to either linear (MSE) to penalize large errors or quadratic (MAE) for robustness to outliers upon a desired threshold, which in this case was set to 0.5 to maintain a balance between the switch since the values have been scaled between 1 and 0.

3.2 Autoencoder Anomaly Detection

The Autoencoder architecture is streamlined with all dense layer with relu activation, starting & ending at fully connected layers of 64 units, that deconstruct & reconstruct the dimensionality respectively, the middle layers encode into 64 and 32, till the bottle neck of 16 units and form again the decoded the weights into 32 and 64 dense units, optimal for low dimensional data with a slight possibility of overfitting. The model first encodes latent representations of the data from 64, 32 and 16, also in terms of feature extraction, the decoding part reconstructs the input with the encoding information.

The model is compiled with Huber loss to combine the advantages of MAE & MSE with a threshold 0.1, both these values are tracked in the metrics. The model is also optimized with the Adam optimizer with the same learning rate to ensure weights are updated gradually and avoid jumping to optimal solution.

3.3 Multi-Task Model with LSTM Forecasting and AE Anomaly Detection

The final model is composed of a complex deep learning architecture, combining two architectures of neural networks to share features related to two similar inputs, sharing a common column ‘Vehicle ID’. The neural networks mentioned are LSTM & Autoencoder. The input layer consists of two separate inputs containing Trip Data & Charging Data as mentioned in the dataset section of this study, shaped according to the sequence length and number of features in the dataset representing sequential data from both sets. Two dense layers are added to create a moderate sized, 64-dimensional representation of both the trip and charge features independently, balancing capacity and computational cost with ‘relu’ activation functions. The shared layer concatenates the two representations and learn interactions between the TripDense & ChargeDense layers, passing it forward to a SharedDense layer of 128 units to expand the previous representation for further enhancement. The LSTM layer captures the shared temporal dependencies and learning patterns of the sequential data, shared between the inputs and divulge using the LSTMOutput layer for SOC Used over time. Encoders also employ LSTM layers with 64 units to encode the shared features into a fixed length vector, summarizing the input sequence, forwarding a compressed representation without sequential dependencies. A RepeatVector is added to the mix to provide bridging between the encoder and the decoder; by providing a sequence aligned representation, it enables the decoder to reconstruct the original sequence step by step and ensures temporal consistency and context propagation through duplication of encoded vector along the sequence length. The Decoder then reconstructs the original sequence from the latent vector and returns a time-distributed dense layer to reconstruct the original Charging features within the sequence. The output layers provide two outputs constructed on shared feature representations, namely SOCOutput provided by the LSTMOutput layer & the AnomalyOutput reconstructs the features from Charging Data to detect anomalies, both the layers use MSE as their loss function, which was selected to reduce interpretation complexity for the predicted values.

4. Results

The LSTM model was trained over 10 epochs with a batchsize of 64, coupled with a low learning rate that yielded optimal training step time. Loss metrics at the start of training were observed to be 0.0832 for the Huber loss function value, with a high mean absolute error of 0.2334 and mean squared error of 0.0856 within a scale of 0 and 1, signifying the loss function as penalizing more through quadratically rather than linearly. Validation loss was observed to be lower than the training loss and gradual decrease in metric values (fig 4.1). The model performance indicates overfitting, as the validation performance is much better than training even through the first epoch, the possible reason would be the uneven timestamps in the dataset which reduce generalization for an LSTM model.

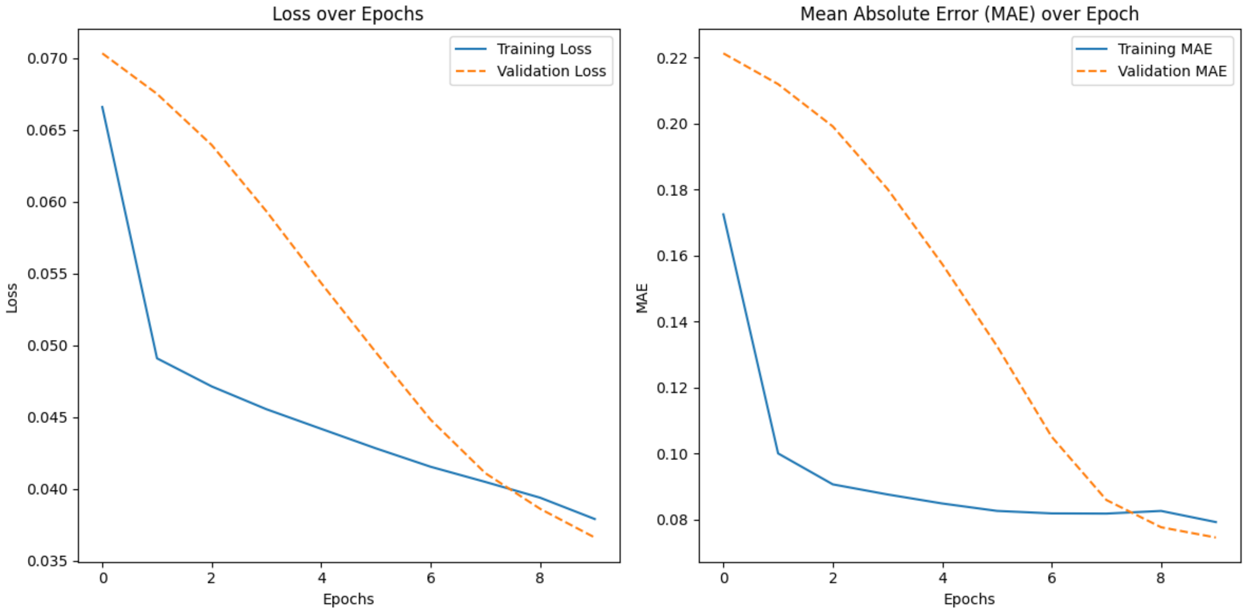


Figure 4.1: The LSTM model's performance over number of epochs.

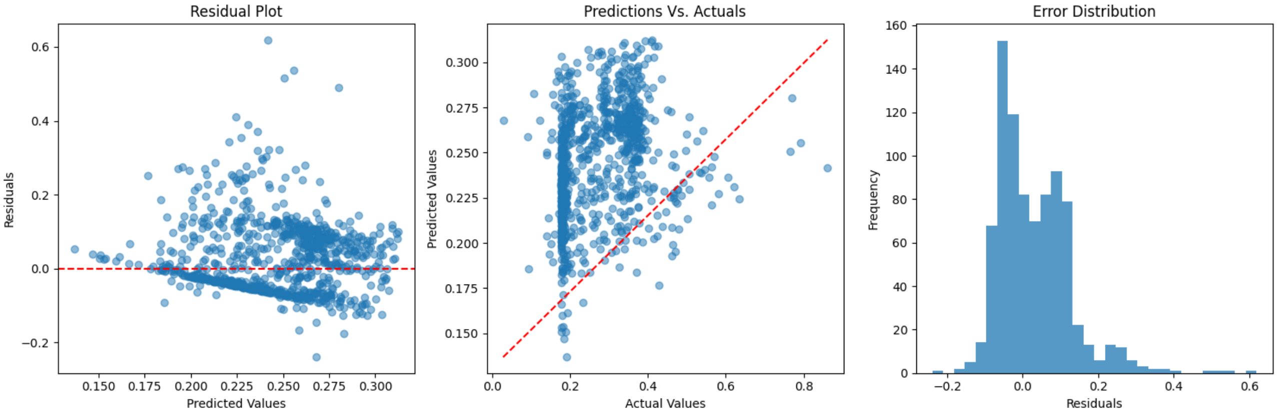


Figure 4.2: Evaluation Metrics LSTM Model.

Residuals for higher predicted values producing significant positive residuals portray the model's skewness to shorter values as mentioned in the dataset section. A higher variance of residuals indicates the model's error varying across different predicted ranges. Predictions Vs. Actual values mainly showing major deviations from the normal portray a data discrepancy towards shorter values. Slightly right skewed values for error distribution indicating larger positive residuals for some values.

General pattern of the data is predicted reasonably well but has issues with certain ranges. A few outliers suggest the model might struggle with capturing variability in specific regions of the input space.

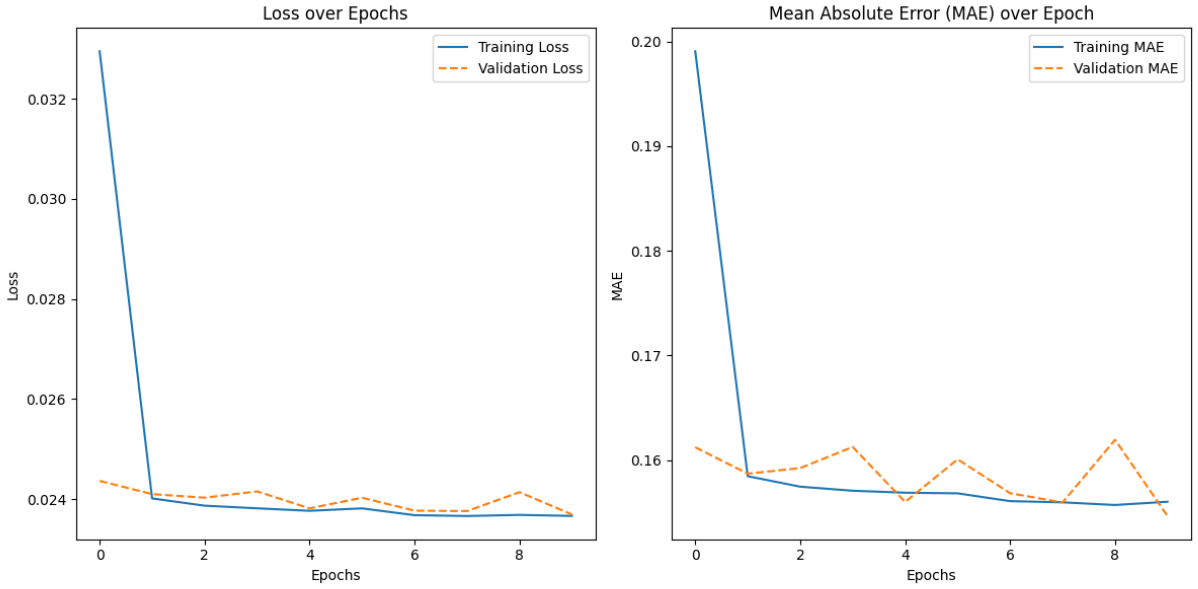
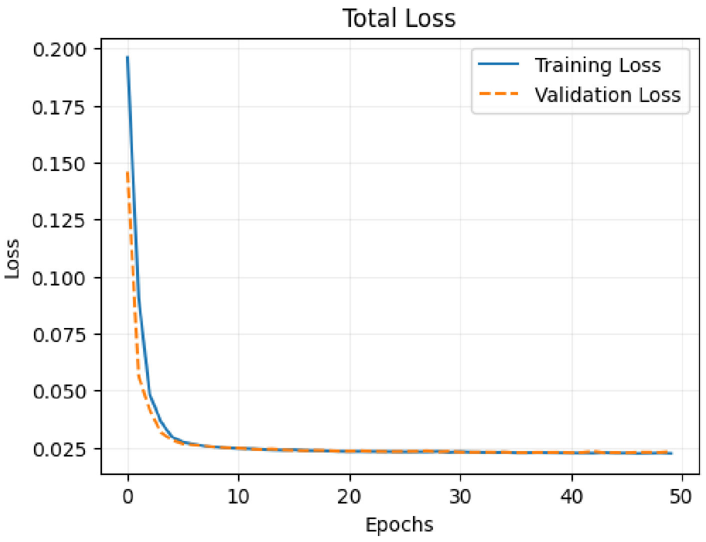


Figure 4.3: Loss metrics for Anomaly Detection.

The Anomaly detection model shows an abrupt decline in loss value after the first epoch (0.0469 to 0.0239) with the validation loss dropping to 0.024, portraying model convergence to features in the Charging data, the loss values stabilize after the first epoch and remain in the region of 0.024. MAE also shows a sharp decrease and then fluctuates between 0.1579 – 0.1558. The small and consistent differences between the training and validation metrics shows the model is not overfitting. There is indication of noise in the validation dataset due to the slight fluctuations observed in the plot.

The model loss values show that the model has learned effectively from training data and generalized well on the validation dataset. Further tuning could result in the model having better stabilization.



The combined model was compiled with MSE loss functions and MAE metrics for the two outputs, MSE is chosen for its effectiveness with penalizing large values & slower convergence to training data compared to other loss functions applied. MAE loss being the counterpart for MSE in terms of penalizing errors, it is selected to monitor how well the model is generalizing on the datasets.

The plot shows a steady decline in values after the first epoch, portraying the model convergence. The following plots show the MAE metric values for both the outputs, due to higher number of outliers and data noise in the Vehicle data, the validation metrics are turbulent.

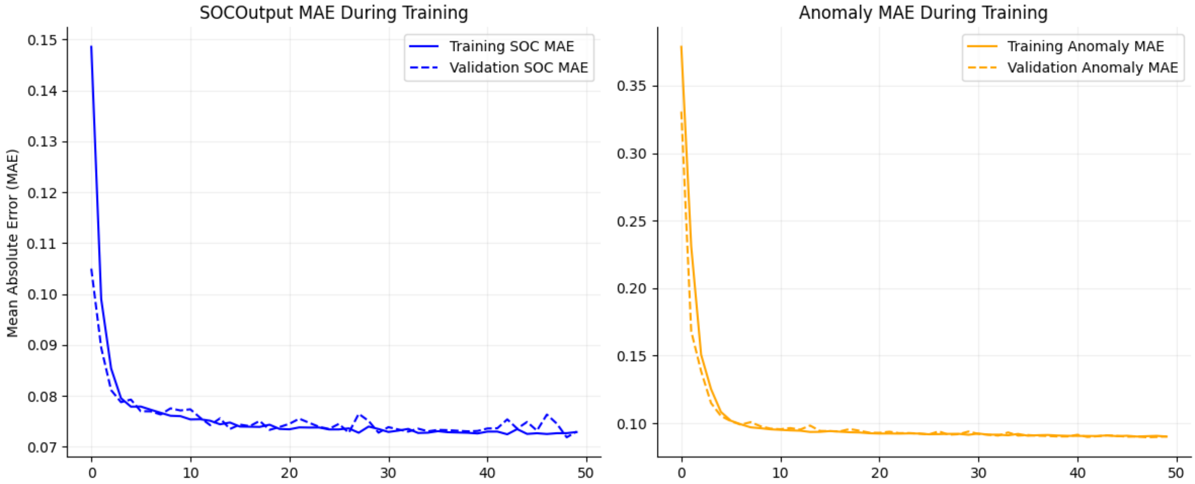


Figure 4.4: Evaluation metrics of Combined model.

The overall ability of the model to generalize upon the shared features of both layers and the addition of the repeat vector layer adds a feature representation of preserved time sequence. Model performance would greatly benefit from data with less noise.

5. Analysis & Discussion

This study aimed to explore employment of LSTM & Autoencoders in a combined architecture for a multi-task deep neural network with the ability to forecast State-of-Charge usage and detect anomalies in Charge Stored by EVs, the combined output would represent a vehicle's energy usage and charging anomalies. The main goal was to compare the model's effectiveness in a feature sharing environment compared to individually. The dataset was first scaled using the standard scaler that returned feature columns with a standard deviation of 1

LSTM Architecture which is widely used for sequence-based predictions of data was able to capture key trends over training epochs, with signs of over fitting as model validation loss was observed to be lower than training loss by the last epoch, the influential factors for this might be the uneven timestamps in the data, which even when sequenced through with the length of 5 sequences and with categorical representation by vehicle ID, encoded into the set using labelbinarizer, could hinder the model's ability to generalize effectively across all inputs[5]. Although reasonably good fit of the model was observed, residuals were skewed severely portraying a higher variance for certain prediction ranges. This indicates potential issues with the model's ability to capture variability specific to some regions of the input space, signifying the presence of outliers and fluctuations in the dataset. This is a common occurrence in time-series data where noise and data discrepancies can distort model predictions.[6]. Mitigations were applied to improve model performance such as changing sequence length, scaling input features which greatly improved model generalization. Optimizations applied to the model are as in Table 5.1.

Hyperparameter	Applied Values	Effect
Learning rate	0.005, 0.01, 0.001, 0.0001	The higher values result in decreased generalization & increased model convergence.
LSTM Units	64, 128, 256	Higher values result in model complexity and increase training time significantly but do not add to model convergence instead add overfitting.
L2 Normalization	0.001	No major effect in observation. Introduces complexity to the model by adding hyperparam sensitivity.
Recurrent Dropout	0.04, 0.05, 0.3, 0.4	Increases convergence time of the model adds fluctuations to loss function, useful to prevent overfitting.
Huber Loss Delta	0.05, 0.25, 0.4, 0.9	Improves model convergence by switching loss function to MSE or MAE upon delta, the higher the value the higher the sensitivity.

Table 5.1: LSTM params tuned and optimized to improve model convergence.

The Autoencoder model trained separately for anomaly detection on charging data, proved exceptional in terms of convergence. A sharp reduction in loss values after the first epoch indicated effectively the model's learning of feature representations in Charging Data. The validation loss also remained stable, which removed the possibility of overfitting. Regardless the model portrayed fluctuations in MAE metric between 0.1579 and 0.1558, indicating noise in the dataset, commonly observed in noisy real-world data. Although the model loss function and metrics were observed to be quite exceptional, the shape of the datasets could not be aligned to generate a residual plot as the datasets needed to be mouldable for all three. However, with complete confidence the model's ability to detect abnormal charging durations or inconsistencies showing malfunctioning equipment or inefficiencies in charging, was preserved with the defined architecture and parameters. More optimizations applied are as in Table 5.2.

Hyperparameters	Applied Values	Effect
Learning Rate	0.0001, 0.001, 0.005	Increases train speed with lower values at the cost of reduced convergence.
Huber Loss Delta	0.5, 0.1, 0.2, 0.3	As mentioned in table 5.1
Dense Units	256, 128, 64, 32, 16	Directly proportional to training speed, adds over fitting with high number of units.

Table 5.2: Autoencoder hyperparams tuned.

The combined model, integrating both the above-mentioned architectures for a multi-task DNN suggests an effective approach for leveraging the learning weights and feature representations from both the models with the shared and concatenated layer, Measures were taken to preserve model's temporal consistency by addition of the Repeat Vector layer, enabling the model to capture temporal dependencies from the LSTM model and detecting charging anomalies simultaneously. As with the other models, the model fit on training data quite exceptionally, but turbulent values were observed in validation loss due to noisy data, affecting the model's generalization capabilities [7].

Hyperparameters	Applied Values	Effect
Dense Units	128,64	Improves convergence at the cost of training time.
Batch Size	64, 32, 16, 8	Improves convergence time with minor effect to generalization due to complex architecture.
Epochs	50, 25, 10,	Higher values to improve convergence. Modified to save training time.

Table 5.3: Combined model hyperparams optimizations.

Future improvements involve addition of more potent preprocessing methodologies to quality dataset, metrics for the data to be re-evaluated and introduction of data imputation mechanisms for precise data retention. Additionally layer orders to maintain input and output shapes for better model evaluation and architecture refinement by rearranging of layers to enhance feature representations learned by the models.

Although similar behaviours were observed in all models due to data noise, the combined model was effective in terms of loss sensitivity. The architecture improvements to be considered would include several exchanges of features but also the preservation of some inherent features as the model's ability to solve its respective problem may be contaminated through concurrent operation. Furthermore, to enhance model fitting GridSearchCV can be utilized to monitor model fits across multiple folds, which was attempted but not employed due its high demand in computational and time resources.

6. Conclusion

With a refined set of tuning parameters and increased model complexity the combined model provides a robust & swift approach to a problem context. Although datasets were suboptimal for detailed exploration of the concept, the model convergence revealed potential in the approach to amplify deep neural network architectures with multi-task functionalities solving more than one or perhaps two problem contexts at once. This approach revealed a moderate demand of Time & computational resource with reasonable resultant values, which were affected mostly by the data noise and lack of consistency in the data time stamps. More robust and preprocessing techniques to retain usable data percentage such as use of imputation, feature engineering would mitigate data noise for future studies.

7. References

[1]. **Multi-task learning for IoT traffic classification: A comparative analysis of deep autoencoders** Dong H, Kotenko I, *Future Generation Computer Systems (2024) 158 242-254*

[2]. **Multi-task convolutional deep neural network for recommendation based on knowledge graphs** Jiang M, Li M, Zhou L, *Neurocomputing (2025) 619 129136*

[3]. **A CNN and LSTM-based multi-task learning architecture for short and medium-term electricity load forecasting** Zhang S, Chen R, Tan J, *Electric Power Systems Research (2023) 222 109507*

[4]. **Mitigating gradient conflicts via expert squads in multi-task learning** Chen J, Er M, *Neurocomputing (2025) 614 128832*

[5]. **Long Short-Term Memory** Hochreiter S, Schmidhuber J *Neural Computation (1997) 9(8) 1735-1780*

[6]. **System Identification** Ljung L *(1998), 163-173*

[7]. **4 - Data Warehousing and Online Analytical Processing** Han J, Kamber M, Pei J *Boston, Morgan Kaufmann, (2012), Third Edition, 125-185*