|  | Pimpri Chinchwad Education Trust’s  Pimpri Chinchwad College of Engineering  Sector No. 26, Pradhikaran, Nigdi, Pune – 411 044 |
| --- | --- |
| **NNFL Case Study : Prediction Model Report with Presentation** | |

Dept: Computer Engineering Academic Year 2024-25 Sem: III

**Subject: Neural Networks and Fuzzy Logic Class: SY BTech**

A CASE STUDY REPORT ON

**“Optimizing Electric Vehicle Performance:**

**Predicting Time, Range, and Efficiency”**

Under The Guidance of

**Dr.Nishi Gupta**

**Submitted By:**

| **Name** | **PRN** |
| --- | --- |
| Manali Dinesh Thorat | 123B1B037 |
| Pranali Rajendra Patil | 123B1B221 |
| Pranoti Santosh Musmade | 123B1B054 |
| Ishwari Balasaheb Walke | 123B1B068 |

**Optimizing Electric Vehicle Performance:**

**Predicting Time, Range, and Efficiency**

[1]

**Introduction**

Electric vehicles (EVs) are gaining momentum in the quest for sustainable transportation. In this case study, we focus on optimizing the performance of a customized EV designed by college students. This vehicle is powered by a specially built system comprising an accumulator (battery pack), a motor, and a motor controller. The accumulator, which consists of multiple cells, serves as the main energy storage unit, charged via a separate power supply. The motor controller bridges the electrical and mechanical components of the vehicle, transferring power from the battery to the motor for driving the wheels.

The customized EV system presents a unique opportunity for studying the interplay between electrical energy and mechanical output. Key parameters such as voltage, state of charge (SoC), motor temperature, speed, and power input/output are crucial for understanding the vehicle’s performance.[2] By capturing real-time data from these components, we can develop a neural network model to predict critical factors such as driving range, efficiency, and time of operation under various conditions.

The goal of this study is to leverage neural networks to process live EV data and predict performance metrics that aid in optimizing the vehicle’s operation. By analyzing data points such as voltage, torque, RPM, temperature, and power flow, we aim to forecast the vehicle’s efficiency and range, allowing for better energy management and extended vehicle lifespan.[3] ,[4]

This work has practical implications for improving the performance and reliability of customized EVs, helping engineers design more efficient systems for both academic and real-world applications.

**Methodology**

**Dataset Explanation**

This dataset appears to be related to electric vehicle(EV) data, containing information about vehicles' academic performance, work experience, and placement status after graduation. Here's a breakdown of the columns and their descriptions:

1. Micros: Microsecond timestamp or identifier for each recorded data point (integer).
2. Voltage: The voltage level of the vehicle's battery pack (float).
3. Pack SoC: State of Charge (SoC) of the battery pack, represented as a percentage (float).
4. DCL: Discharge Current Limit, indicating the maximum allowable current discharge (integer).
5. Failsafe: A failsafe indicator, likely representing whether a safety mechanism has been triggered (binary, 0 or 1).
6. DTC1: Diagnostic Trouble Code 1, indicating potential faults or issues detected in the system (integer).
7. DTC2: Diagnostic Trouble Code 2, additional fault detection indicator (integer).
8. Temperature High: Maximum temperature reading from a set of sensors, possibly monitoring battery or motor temperature (integer).
9. PC: Power Consumption or power control variable, potentially related to energy usage (float).
10. Torque: Torque produced by the motor, measured in Nm (float).
11. RPM: Revolutions Per Minute of the motor (integer).
12. Motor\_temp: Temperature of the motor (integer).
13. Controller\_temp: Temperature of the motor controller (integer).
14. time\_interval: Time interval between successive data points (float).
15. input\_power: Input power to the motor system, measured in kW or similar units (float).
16. output\_power: Output power generated by the motor system (float).
17. speed: Vehicle speed at the time of recording, likely in km/h (float).
18. E\_remaining: Energy remaining in the battery, likely in kWh (float).
19. E\_per\_km: Energy consumption per kilometer, indicating the efficiency of energy use (float).
20. motor\_efficiency: Efficiency of the motor, calculated as a ratio of output power to input power (float).

Based on further study, we identified the need for additional columns which were not present initially in our dataset, we created them manually. These new columns are as follows:

**Time\_interval**:

remaining\_time\_hours = remaining\_range\_km / average\_speed\_kmph

Remaining Time (minutes) = remaining\_time\_hours \* 60

**input\_power**:

Input Power =Voltage \* Peak Current

**output\_power:**

Output Power =Torque \* RPM

**Motor Efficiency:**

Motor Efficiency = (Output

Power / Input Power) \* 100%

**speed:**

speed=(rpm\*pi\*diameter of vehicle \*18)/ (60\*5\*driveratio),

where drive ratio=15.3, diameter of vehicle =0.586 m

**E\_remaining:**

E\_remaining​=(Pack SoC / 100​) ×Emax​(in kWh)

**E\_per\_km**:

E\_per\_km = speed (km/h) / input\_power(kW)

**Remaining Range (R\_km):**

R\_km ​= E\_per\_km​ / E\_remaining

​​

This dataset can be used to analyze various factors that may influence electric vehicle performance, such as motor efficiency, power consumption, and energy

usage. It can also be used to build predictive models for key performance indicators like range, power output, motor efficiency, and operational time, providing valuable insights into vehicle efficiency and areas for optimization.

**Data Preprocessing**

The project begins by importing the necessary libraries, including NumPy, Pandas, Matplotlib, Seaborn, and scikit-learn. The electric vehicle (EV) data is loaded from a CSV file using Pandas, and relevant features are selected and stored in the variable X, while the target variable (placement status) is stored in y. The data is then split into training and testing sets using the train\_test\_split function from scikit-learn, with a test set size of 30% and a random state of 30. The feature data (X\_train and X\_test) is standardized using the StandardScaler from scikit-learn to ensure that all features are on the same scale.

**Model Architecture**

The architecture described in your code is built for predicting key vehicle performance metrics, such as **motor efficiency**, **range**, and **remaining time**, using different machine learning and deep learning models. Here’s a detailed breakdown of the architecture for each section:

### **1. Range Prediction Model**

#### **Model Architecture:**

* **Input Features**:
  + **Pack SoC (State of Charge)**
  + **Voltage**
  + **PC (Power Control)**
  + **input\_power**
  + **speed**
* **Target Variable**: Range\_km(Vehicle Range in kilometers)
* **Model**: A Feed-Forward Neural Network (Sequential model) built with Keras/TensorFlow.

#### **Detailed Architecture:**

* **Input Layer**: 5 input features (Pack SoC, Voltage, PC, Input Power, Speed)
* **Hidden Layers**:
  + **Layer 1**: 64 neurons, ReLU activation function
  + **Layer 2**: 32 neurons, ReLU activation function
* **Output Layer**: A single neuron that predicts **Range** in kilometers.
* **Training**:
  + Optimizer: Adam optimizer
  + Loss function: Mean Squared Error (MSE)
  + Early stopping is not explicitly mentioned here, but could be included for optimization.
  + Batch size: 32
  + Epochs: 100

### **2. Remaining Time Prediction Model**

#### **Model Architecture:**

* **Input Features**:
  + **Pack SoC (State of Charge)**
  + **Voltage**
  + **Input Power**
* **Target Variable**: Remaining time (in minutes)
* **Model**: A Feed-Forward Neural Network (Sequential model) built with Keras/TensorFlow.

#### **Detailed Architecture:**

* **Input Layer**: 3 input features (Pack SoC, Voltage, Input Power)
* **Hidden Layers**:
  + **Layer 1**: 32 neurons, ReLU activation function
  + **Layer 2**: 16 neurons, ReLU activation function
* **Output Layer**: A single neuron that predicts the **Remaining Time** in minutes.
* **Training**:
  + Optimizer: Adam optimizer
  + Loss function: Mean Squared Error (MSE)
  + Early stopping:
    - Monitors the validation loss.
    - Stops training if there is no improvement in 10 consecutive epochs.
  + Epochs: Up to 200, with early stopping in place.
  + Validation split: 20% of the training data used for validation during training.

### **3. Motor Efficiency Prediction Model**

#### **Model Architecture:**

* **Input Features**:
  + **Voltage**
  + **PC (Power Control)**
  + **Torque**
  + **RPM (Revolutions per Minute)**
* **Target Variable**: Motor efficiency
* **Model**: Multi-layer Perceptron (MLP) using MLPRegressor from Scikit-learn.

#### **Detailed Architecture:**

* **Input Layer**: 4 input features (Voltage, PC, Torque, RPM)
* **Hidden Layers**:
  + **Layer 1**: 100 neurons, ReLU activation function
  + **Layer 2**: 100 neurons, ReLU activation function
* **Output Layer**: A single neuron that predicts **motor efficiency**
* **Training**:
  + Optimizer: Adam optimizer (default in MLPRegressor)
  + Loss: Mean squared error
  + Early stopping and learning rate adaptation are used to improve convergence.

### **Key Components of Each Model:**

#### **1. Data Preprocessing:**

* For each model, the input data is first split into training and testing sets using train\_test\_split().
* The input features are then standardized or normalized using **StandardScaler** or **MinMaxScaler**, which scales the input data to a common range, improving the model's performance.

#### **2. Model Training:**

* Each model is trained using **backpropagation** and **stochastic gradient descent (SGD)** or the **Adam optimizer**.
* The loss function is **mean squared error (MSE)**, which is suitable for regression tasks.
* Models are trained in a supervised manner, where the network adjusts its weights based on the error between the predicted values and the actual values.

#### **3. Evaluation and Metrics:**

* After training, models are evaluated on the test set using various regression metrics:
  + **Mean Absolute Error (MAE)**
  + **Mean Squared Error (MSE)**
  + **Root Mean Squared Error (RMSE)**
  + **R-squared (R²) score**

#### **4. Prediction Functions:**

* The model is designed to take user inputs (e.g., Pack SoC, Voltage, Input Power) to make real-time predictions for the vehicle’s remaining time, range, or motor efficiency.

### **Summary of Model Layers:**

* **Input Layer**:
  + The number of input features depends on the specific problem (e.g., 4 for motor efficiency, 5 for range prediction).
* **Hidden Layers**:
  + Typically 1-2 hidden layers.
  + The number of neurons varies (e.g., 32, 64, or 100 neurons per layer).
  + **Activation Function**: ReLU (Rectified Linear Unit) is used as the activation function for the hidden layers to introduce non-linearity.
* **Output Layer**:
  + A single neuron output for regression tasks (e.g., predicting range, time, or efficiency).
* **Optimizer**: Adam optimizer is used in all models.
* **Loss Function**: Mean squared error (MSE) is used to minimize prediction errors.
* **Training Process**:
  + The models use early stopping to prevent overfitting.
  + Batch size and the number of epochs are specified, with validation splits used for model evaluation.

This architecture is well-suited for solving regression problems related to vehicle performance prediction, with focus on motor efficiency, range, and remaining operational time.

**Relu Activation Function**

The **ReLU (Rectified Linear Unit)** function is employed in the three codes for predicting range, time, and efficiency due to its effectiveness in facilitating model training. It addresses the vanishing gradient problem commonly seen with traditional activation functions like sigmoid and tanh, allowing for faster learning in deeper networks.

In the range and time prediction models, ReLU promotes sparsity by outputting zero for negative inputs, which helps create a more efficient network where only active neurons contribute to the learning process. This nonlinearity is crucial for capturing complex relationships in the data.

In the efficiency prediction code, ReLU aids in improving the model's ability to learn intricate patterns from features such as voltage and RPM. Overall, its ability to enhance training speed and efficiency while effectively handling complex data makes ReLU a suitable choice for all three projects.

**Adam Optimizer**

The **Adam optimizer** is utilized in the three codes for predicting range, time, and efficiency due to its efficiency and adaptability. It combines the advantages of AdaGrad and RMSProp, adjusting learning rates based on the first and second moments of the gradients. This feature helps models converge faster, making it particularly effective for complex datasets.

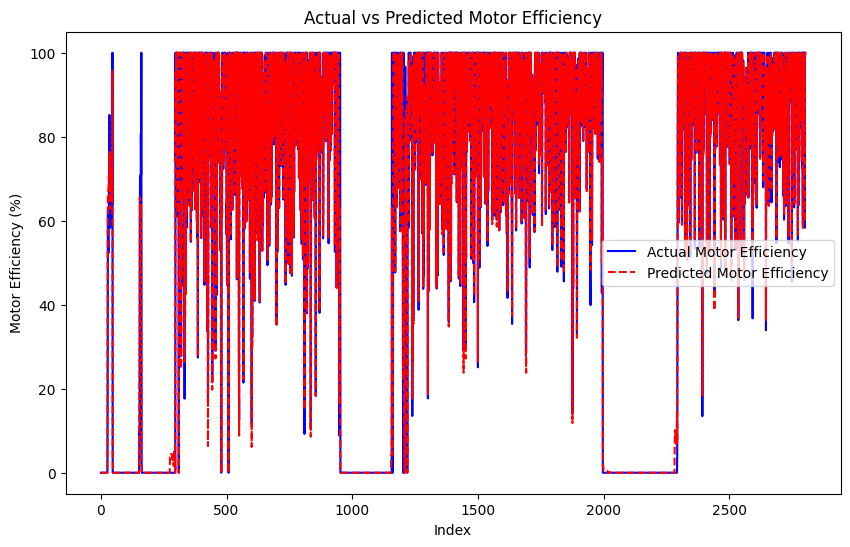
In the range and time prediction models, Adam optimizes learning from battery characteristics, such as voltage, power, and speed, which can vary widely during operation. By adjusting the learning rate dynamically, Adam helps the models learn more effectively from these varying inputs, resulting in faster convergence and potentially more accurate predictions.

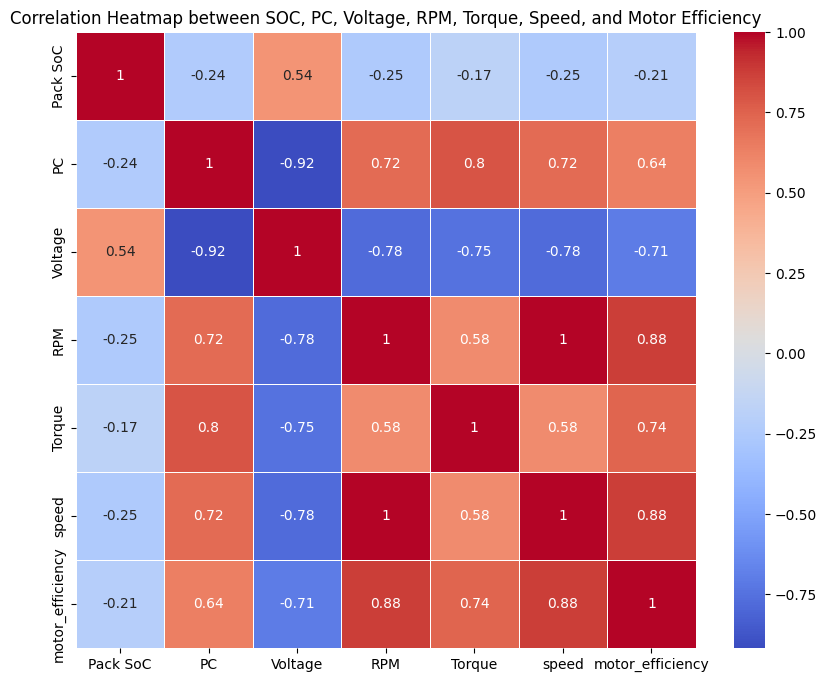
For the efficiency prediction model, Adam's adaptive learning capability ensures that the model can quickly respond to changes in the relationship between input features, such as power output and temperature. This adaptability is crucial because efficiency can be influenced by a range of factors that may not be linearly related.

In all three models, the use of Adam enhances training stability and ensures that the models do not get stuck in local minima, thereby improving the overall performance. The optimizer's ability to adjust the step size based on the gradient information from previous iterations allows it to maintain an appropriate balance between speed and accuracy in learning.

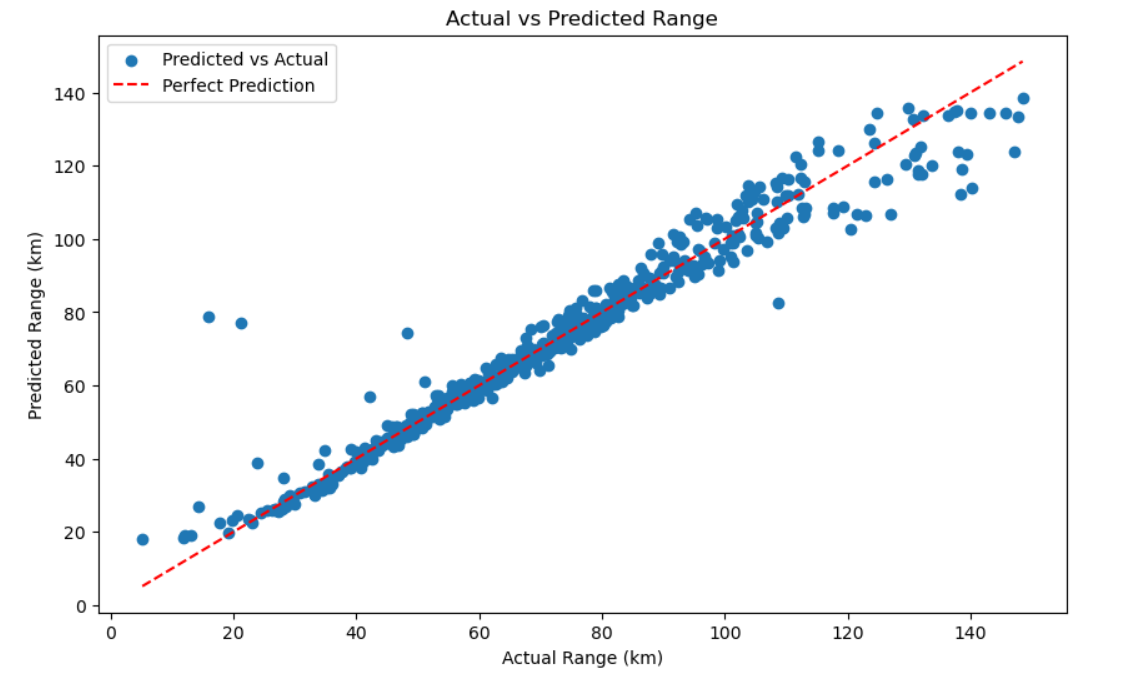
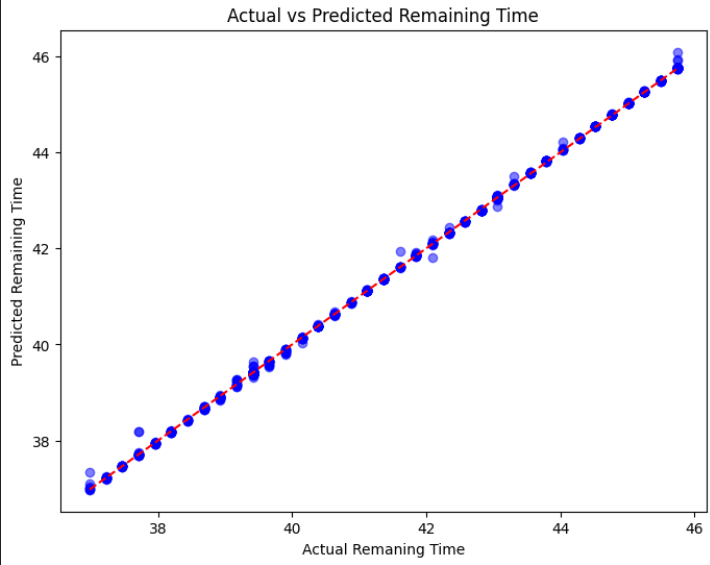
**Data Visualization**

The correlation matrix helps enhance performance and accuracy in predicting range, time, and efficiency by identifying relationships among features. In range and time prediction, it reveals how battery characteristics like voltage, current, and temperature interact, guiding feature selection and model refinement. For the efficiency prediction, its adaptive learning rates and bias-correction ensure robust training, especially in the early stages. Overall, Adam’s efficiency and robustness make it a preferred choice for these projects

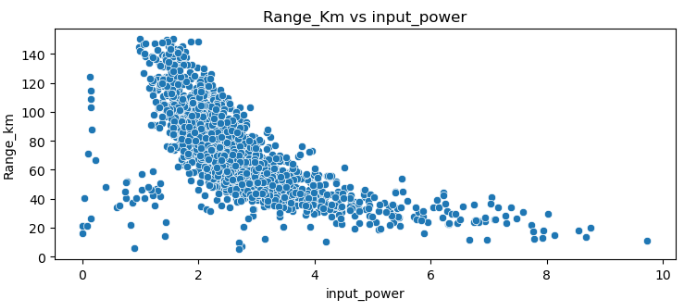
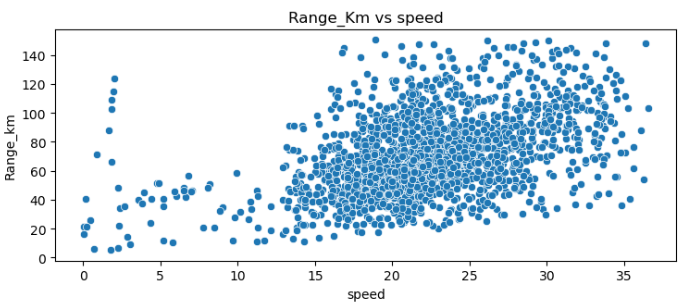


** Fig.2 Result for Motor Efficiency**

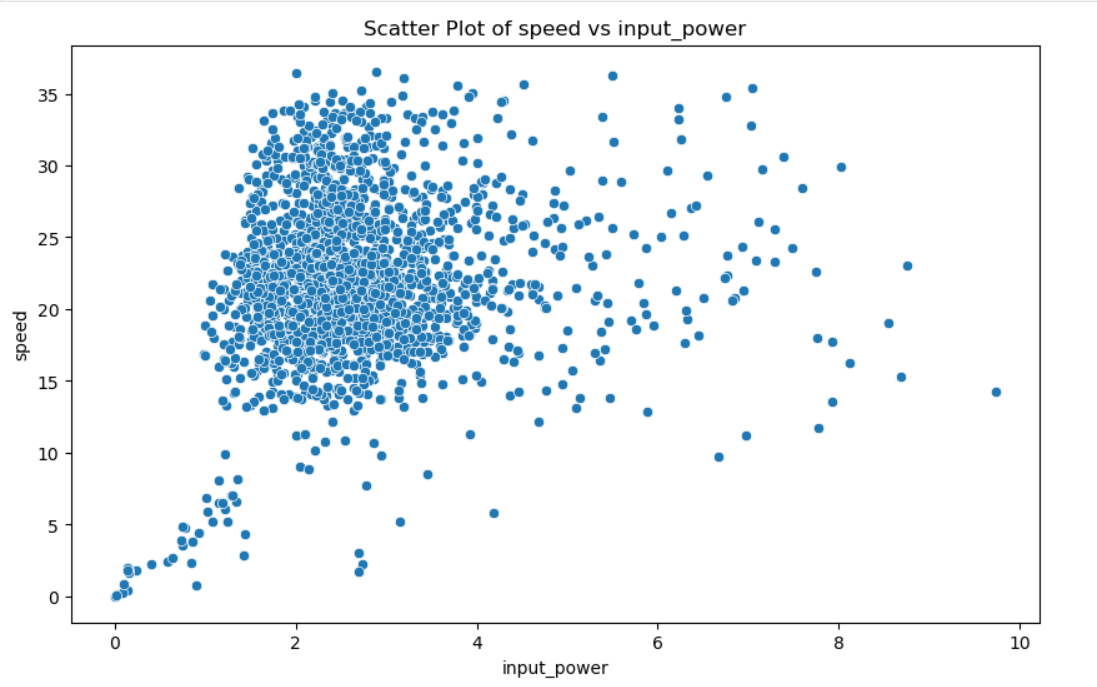
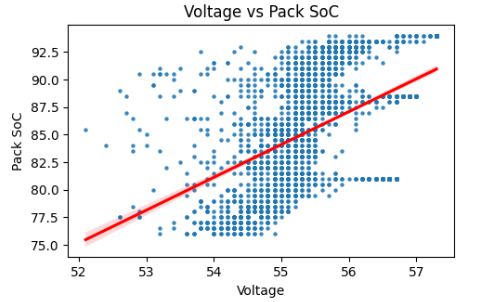
**Fig.1 Correlation Matrix**

**** 

**Fig.3 Result for Range Prediction Fig.4 Result for time prediction**

** **

**Fig.5 Range vs Power Fig.6 Range vs Speed**

**** 

**Fig.7 Speed vs Power Fig.8 Voltage vs Pack SoC**

**Results**

1. **Range Prediction**

Initially, the loss was high because the vehicle was at rest (i.e., speed = 0). As a result, whenever speed was used in any formula, it resulted in a multiplication value of 0 and caused empty boxes in the dataset when dividing by 0. However, as the vehicle began to move and the speed increased, the loss decreased, leading to more accurate results. And finally we achieved accuracy shown as following:

**Model Evaluation Metrics:**

Mean Absolute Error (MAE): 3.55

Mean Squared Error (MSE): 40.94

R-squared (R²) Score: 0.95

Accuracy = 0.95

1. **Time Prediction**

Initial SOC is a primary indicator of the total remaining charge. Higher SOC means more stored energy, allowing for a longer operational time. As the SOC is higher more voltage is demanded and more input power is given to the controller and then to the motor through the controller. As power consumption rate increases SOC starts decreasing which reduces remaining operational time. So that we have predicted remaining operational time with the following results:

**Model Evaluation Metrics:**

Mean Absolute Error (MAE): 0.0235

Mean Squared Error (MSE): 0.0036

Root Mean Squared Error (RMSE): 0.06015

R-squared (R²): 0.999

1. **Motor Efficiency**

Initially, when the EV was at rest, increasing torque demanded more current which caused heat loss and hence motor efficiency was 0 initially.

This increased heat loss caused higher loss value initially and further as vehicle gained the speed, need for current decreased as the torque decreased and hence due to no heat loss motor efficiency increased which made loss decrease as the EV gained speed.

**Model Evaluation Metrics:**

Mean Squared Error (MSE) : 21.1212

Root Mean Squared Error (RMSE) : 4.5958

Mean Absolute Error (MAE) : 1.2795

R -Squared Error (R2) : 0.9873

**Limitations/Discussion**

**1.Battery Life Prediction Limitation:**

One significant limitation of this model is it is not able to accurately predict the overall battery life of the electric vehicle. Battery life, in this context, refers to the number of charge-discharge cycles a battery can endure before its capacity significantly degrades. A **charging cycle** is defined as the process of fully charging the battery from a low State of Charge (SoC) to 100%, while a **discharging cycle** refers to the process of using the battery's energy until it reaches a low SoC, ready for recharging.

In real-world applications, predicting the long-term life of an EV battery requires data from a large number of full charging and discharging cycles—typically more than 2,000 cycles. However, for more accurate and reliable predictions, datasets with upwards of 100,000 cycles are necessary. Collecting such extensive data is incredibly challenging due to the long time periods involved, the cost of testing, and the variation in real-world usage patterns.

Without such large-scale, high-quality data, the model cannot fully capture how the battery will degrade over time, which limits its ability to provide accurate predictions for long-term battery health and lifespan.

**2.Limitation Due to Absence of Key Variables:**

Another limitation in this study is that the original dataset did not contain several critical variables that directly influence the accuracy of the predictions, such as **input power**, **output power**, **speed**, **time interval**, **energy remaining (E\_remaining)**, and **energy per kilometer (E\_per\_km)**. These variables had to be **calculated using formulas** based on other parameters in the dataset, rather than being provided as raw data.

While these calculated values provide useful approximations, relying on derived data can introduce inaccuracies in the prediction model. Directly measured data for these variables would offer greater precision.

**3. Assumption of Linear Relationships:**

The model assumes linear or simple non-linear relationships between variables like **SoC**, **Voltage**, and **Temperature**. However, real-world interactions between these factors are often highly **non-linear**. For example, small changes in temperature or SoC can cause disproportionate impacts on range and efficiency. This assumption may oversimplify complex relationships, limiting the accuracy of the predictions. More advanced models could better capture these nonlinear dynamics for improved performance forecasting.

**Conclusion**

**1. Range Prediction**

At higher speeds, the motor typically works harder to maintain or increase speed [fig.5], resulting in greater energy consumption. This increased power draw from the battery reduces the vehicle's range [fig.6], as more energy is required to sustain higher rotational speeds. Consequently, at high speeds, the motor may consume more power, which negatively impacts the range.

To address this issue, we need to determine a cruising speed for the electric vehicle where power consumption is minimized while maintaining a high speed. By stabilizing power demand at this cruising speed, we can achieve more efficient energy usage [fig.3]. This concept can be illustrated with the graphs in the Data Visualization Section.

Here’s a hypothetical conclusion based on this kind of analysis:

The maximum predicted range occurs when:

* Speed is between 17-28 km/h.
* Input power is between 2-4 kW.

**Then range will be maximum within the interval of 40-80 km**

Operating within these speed and power intervals allows the vehicle to achieve the best possible range according to the model's predictions.

**2.Time Prediction:**

In the analysis of remaining operational time for the electric vehicle (EV), several key relationships between **State of Charge (SOC)**, **voltage**, and **input power** play a critical role in determining how long the vehicle can run based on varying conditions:

1. **State of Charge (SOC)**: The SOC directly influences the remaining
2. operational time. Higher SOC values correlate with greater energy availability, allowing the vehicle to operate for longer durations. Conversely, as SOC decreases, the remaining time reduces because less energy is stored in the battery. This relationship highlights the importance of maintaining a higher SOC to maximize operational efficiency.
3. **Voltage**: Voltage levels are affected by the SOC. A higher SOC typically results in higher voltage(Fig. 8), which enables more efficient power delivery to the vehicle's systems. As SOC decreases, voltage may drop, potentially leading to increased energy loss and reduced efficiency. Consequently, a drop in voltage can diminish remaining operational time, particularly if input power demand remains high.
4. **Input Power**: The input power reflects the energy consumption of the vehicle's systems. When SOC and voltage are at optimal levels, the vehicle can sustain lower input power levels, maximizing remaining time. However, if the input power increases (due to higher speeds or additional load), it can accelerate battery depletion, thus shortening the remaining time. Efficient management of input power is crucial to optimizing operational duration.

In conclusion, the remaining operational time of the EV can vary significantly based on the interplay of SOC, voltage, and input power. By optimizing these factors, it is possible to enhance the vehicle's efficiency and prolong its usability across different driving conditions.

**3. Motor Efficiency :**

1. SoC is a fundamental parameter according to which current, and efficiency changes. Initially when EV is at rest, there is need for toque for which demand for current increases , if SoC is low high current withdrawal takes place as voltage doesn’t remain constant. This
2. causes a lot of heat loss and hence motor efficiency decreases.
3. As the torque increases, the need for current increases and hence current draw increases and so motor efficiency decreases.
4. As initially speed is zero, torque increases and so current demand increases, hence motor efficiency is low initially. Later on as EV gains speed, need for torque decreases hence current demand decreases and with constant voltage motor efficiency increases.
5. So as long as the SoC has not reached 25%, the motor efficiency will not decrease a lot as heat loss will not take place due to enough voltage. Once, SoC reaches 25%, the efficiency of motor will decrease as decreasing speed increases torque and torque increases current demand which causes heat loss.
6. When input power (power delivered from battery to the motor) is equal to the output power (mechanical power to draw the wheels), motor efficiency is 100%.

**Future Scope**

Predicting parameters like Range, Motor Efficiency, and Time in electric vehicles (EVs) offers several key benefits for the future of transportation:

1. **Extended Range and Optimal Trip Planning:**
   * Predicting the vehicle's range helps drivers plan routes more effectively, minimizing the risk of running out of charge and reducing range anxiety. It also allows charging stations to be better placed and utilized, enhancing the overall EV experience.
2. **Improved Motor Efficiency:**
   * By predicting motor efficiency, the vehicle can adjust driving modes (e.g., eco, sport) in real time to optimize energy use, reducing power loss and increasing overall efficiency. This results in lower energy consumption, leading to longer trips on a single charge.
3. **Time Optimization:**
   * Predicting how long a trip will take based on real-time factors like battery usage, speed allows for better time management. It helps optimize charging schedules and provides more accurate arrival times, improving convenience for EV users.

**References**

Dataset:

[1] <https://drive.google.com/file/d/1LEJ5RfJt_cquUBCE8Wgn1TqJwY2cNwrR/view?usp=sharing>

Codes and Graphs:

[2] <https://drive.google.com/file/d/1_fBYhO4gDsnYUFj7wYY0buACsg3V-mCL/view?usp=sharing>

[3] Prediction of Electric Vehicle driving range and Performance Characteristics:

A review on analytical modeling strategies with its influential factors and improvisation

techniques. (2023). IEEE Journals & Magazine | IEEE Xplore.

<https://ieeexplore.ieee.org/document/10323278>

[4] Jung, W., Ismail, A., Ariffin, M. F., & Noor, S. A. (2011). Study of Electric vehicle Battery

Reliability Improvement. *ResearchGate*.

[https://www.researchgate.net/publication/275214779\_Study\_Of\_Electric\_Vehicle\_Battery\_](https://www.researchgate.net/publication/275214779_Study_Of_Electric_Vehicle_Battery_Reliability_Improvement)

[Reliability\_Improvement](https://www.researchgate.net/publication/275214779_Study_Of_Electric_Vehicle_Battery_Reliability_Improvement)