Optimizing Battery Temperature for Electric Vehicles Using Quantum Annealing: Implications for Grid Load Management and Carbon Emission Reduction

Background and Introduction

1. Quantum AI and Climate :

Climate modeling and simulation are integral to understanding and predicting the impacts of climate change. These processes are complex and require significant computational resources, as they simulate intricate interactions within the Earth's climate system. As climate change accelerates, there is an urgent need for faster and more efficient modeling methods to better inform climate policy and adaptation strategies. The advent of artificial intelligence (AI) and quantum computing presents a promising solution to this challenge. AI has revolutionized data analysis and pattern recognition, while quantum computing offers exponential advantages for solving computationally intensive problems. Combining these technologies could significantly enhance our ability to model and manage climate impacts, including carbon emissions, smart charging of electric vehicles (EVs), and grid management.

2. Carbon Emissions, Smart Charging and Grid Management

Carbon emissions, particularly from fossil fuels, are a major driver of climate change. Transitioning to electric vehicles (EVs) is a key strategy for reducing these emissions, as EVs produce no tailpipe emissions. However, the electricity required to power EVs and the processes involved in battery manufacturing still contribute to carbon emissions. Optimizing battery usage is therefore critical to maximizing the environmental benefits of EVs. Smart charging strategies—such as determining the optimal time, current, voltage, and temperature for charging—can further reduce emissions and improve battery performance.

3. Impact of Temperature on SoC and SoH

Battery temperature significantly affects the State of Charge (SoC) and State of Health (SoH). Higher temperatures can boost SoC but can also degrade SoH and reduce overall efficiency. Therefore, managing battery temperature is crucial for optimizing battery life and performance.

4. How are estimates done - Classical vs AI vs Quantum

- Estimating battery performance involves various methods:
  - **Classical Methods**: The Extended Kalman Filter (EKF) is a traditional approach used for SoC and SoH estimation.
  - **AI Methods**: Techniques such as Long Short-Term Memory (LSTM) networks, Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Deep Learning models offer advanced prediction capabilities.
  - **Quantum Methods**: Recent developments include Variational Quantum Algorithms (VQA), Quantum Approximate Optimization Algorithm (QAOA), and Quantum Neural Network Regression, which leverage quantum computing to potentially outperform classical and AI methods in complex scenarios.

5. Discuss the Already implemented Quantum Methods

Quantum Methods in Battery Management:

- **Variational Quantum Algorithm for SOH Estimation**: This method improves SOH estimation accuracy by combining various model advantages and optimizing hyperparameters with a quantum circuit module. It has shown a 77.4% improvement in prediction accuracy compared to traditional methods.
- **Generalized Regression Neural Network and Quantum Genetic Algorithm**: This approach enhances SOH estimation accuracy using a combination of Particle Filter (PF), Quantum Genetic Algorithm (QGA), and Generalized Regression Neural Network (GRNN), resulting in high accuracy and low computational cost.
- **Quantum Neural Network Regression**: This method models battery capacity degradation using a classical-quantum hybrid approach, demonstrating potential advantages over conventional neural networks in handling large datasets.

6. Funnel into QUBO and Quantum Annealing

To address the challenges of optimizing battery temperature while balancing the State of Charge (SoC) and maximizing the State of Health (SoH), this project employs Quantum Annealing. This approach utilizes Quantum Unconstrained Binary Optimization (QUBO) to frame the problem as a combinatorial optimization task. Quantum Annealing is then used to solve this QUBO problem efficiently.

Quantum Annealing offers a different approach compared to classical and AI methods by directly tackling the optimization of complex, high-dimensional problems. By formulating the optimization of battery temperature, SoC, and SoH as a QUBO problem, Quantum Annealing can potentially find more effective solutions that balance these competing factors. This technique leverages quantum superposition and entanglement to explore a vast solution space, aiming to minimize temperature, optimize SoC, and maximize SoH simultaneously.

This innovative approach could lead to significant improvements in battery performance and efficiency, with implications for both individual battery management and broader grid load management.

7. Dataset Used:

The dataset used in this project is derived from extensive testing conducted by NASA. It includes detailed battery performance data under various operational profiles and temperatures, making it ideal for developing and validating prognostic algorithms. This data encompasses a range of conditions and aging effects, allowing for robust predictions of Remaining Useful Life (RUL) and accurate modeling of battery degradation.

Objectives/Problem Statement

The main objective of this project is to optimize the battery usage to maximize the life of the battery while keeping the carbon emission low and high efficiency. The focus was on optimizing the temperature, which is one of the major factors which can affect the SoC, SoH and efficiency of the battery at the same time. Attempts were made to minimize the temperature in order to maximize the SoH of the battery, constrained by the already existing SoC operating limits (20%-80%).

Methodology:

1. Dataset Preparation

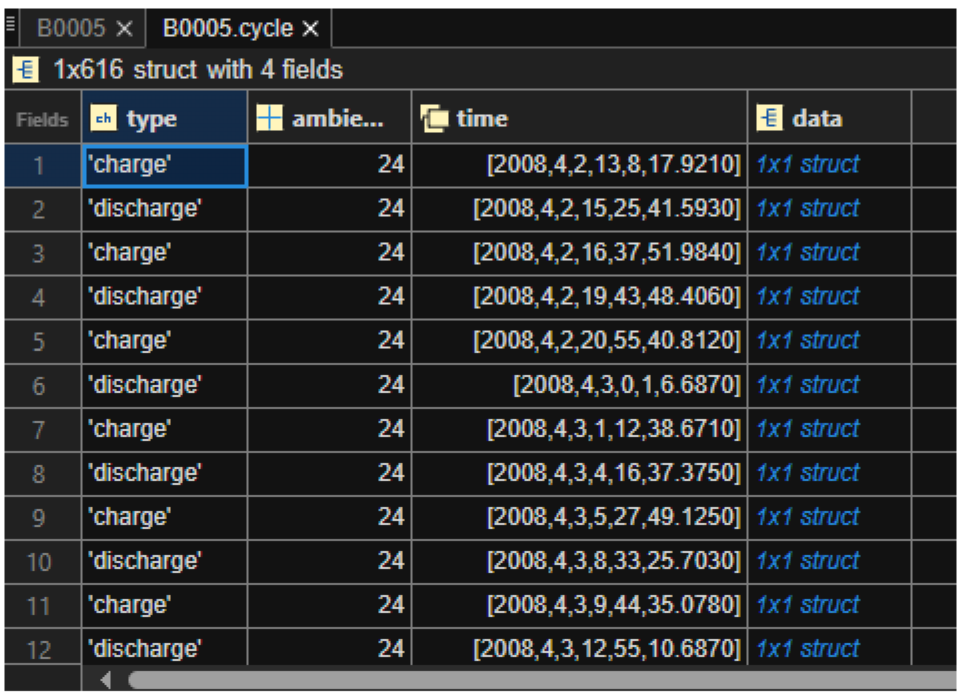
A. Files inside B0005.mat

B0005 ×

1x1 struct with 1 field

Field	Value	Size	Class
cycle	1x616 struct	1x616	struct
type	['charge','discharge','charge', ...]	-	-
ambient_temperature	[24,24,24, ...]	-	-
time	[[2008,4,2,13,8,17.9210],[2008,4,2,15,25,4...	-	-
data	[1x1 struct,1x1 struct,1x1 struct, ...]	-	-

B. Cycle has 616 rows containing data for the charging and discharging cycle:



B0005 × B0005.cycle ×

1x616 struct with 4 fields

Fields	type	ambie...	time	data
1	'charge'	24	[2008,4,2,13,8,17.9210]	1x1 struct
2	'discharge'	24	[2008,4,2,15,25,41.5930]	1x1 struct
3	'charge'	24	[2008,4,2,16,37,51.9840]	1x1 struct
4	'discharge'	24	[2008,4,2,19,43,48.4060]	1x1 struct
5	'charge'	24	[2008,4,2,20,55,40.8120]	1x1 struct
6	'discharge'	24	[2008,4,3,0,1,6.6870]	1x1 struct
7	'charge'	24	[2008,4,3,1,12,38.6710]	1x1 struct
8	'discharge'	24	[2008,4,3,4,16,37.3750]	1x1 struct
9	'charge'	24	[2008,4,3,5,27,49.1250]	1x1 struct
10	'discharge'	24	[2008,4,3,8,33,25.7030]	1x1 struct
11	'charge'	24	[2008,4,3,9,44,35.0780]	1x1 struct
12	'discharge'	24	[2008,4,3,12,55,10.6870]	1x1 struct

C. Data during charging cycle for charge the fields are:

Voltage measured:
Battery terminal voltage (Volts)
Current measured:
Battery output current (Amps) Temperature measured:
Battery temperature (degree C) Current charge:
Current measured at charger (Amps) Voltage charge:
Voltage measured at charger (Volts) Time:
Time vector for the cycle (secs)

B0005 × | B0005.cycle × | B0005.cycle(1).data ×

1x1 struct with 6 fields

Field	Value	Size	Class
Voltage_measured	1x789 double	1x789	double
Current_measured	1x789 double	1x789	double
Temperature_measured	1x789 double	1x789	double
Current_charge	1x789 double	1x789	double
Voltage_charge	1x789 double	1x789	double
Time	1x789 double	1x789	double

D. Data during discharging cycle : for discharge the fields are:
Voltage measured:
Battery terminal voltage (Volts) Current measured:
Battery output current (Amps) Temperature measured:
Battery temperature (degree C) Current charge:
Current measured at load (Amps) Voltage charge:
Voltage measured at load (Volts) Time:
Time vector for the cycle (secs) Capacity:
Battery capacity (Ahr) for discharge till 2.7V

B0005 × | B0005.cycle × | B0005.cycle(1).data × | B0005.cycle(2).data ×

1x1 struct with 7 fields

Field	Value	Size	Class
Voltage_measured	1x197 double	1x197	double
Current_measured	1x197 double	1x197	double
Temperature_measured	1x197 double	1x197	double
Current_load	1x197 double	1x197	double
Voltage_load	1x197 double	1x197	double
Time	1x197 double	1x197	double
Capacity	1.8565	1x1	double

Organizing the discharge cycles: Discharge cycle program file

2. Objective Function

To obtain an equation for Temperature as a function of Voltage, Current, and Tim, a multivariate regression analysis was performed. This process gives an approximate mathematical model that describes the relationship between these variables.

The simplest approach was to use a linear model, but depending on the data, one might need a more complex non-linear model.

Linear model: $Temp(V, I, t) = a0 + a1V + a2I + a3t + a4VI + a5Vt + a6It + a7VIt$

Where V is Voltage, I is Current, t is Time, and a0 through a7 are coefficients to be determined.

Ipython notebook with libraries like scikit-learn was used to perform this analysis.

Relation obtained :

$$\mathrm{Temp}(V, I, t) = -16.3967 + 39.4782 \cdot V + 7.3247 \cdot I + 0.0056 \cdot t - 7.0293 \cdot V^2 + 0.5705 \cdot V \cdot I - 0.0011 \cdot V \cdot t + 3.3260 \cdot I^2 - 0.0004 \cdot I \cdot t - 0.0000 \cdot t^2$$

The R-squared value of **0.96** indicates that the model fits well into the data.

We tried the below approaches to further our understanding.

Approach 1: Expanded Temperature function

Approach 2: BQM Solving

Approach 3: Separate relations of Temp vs Each variable

Different regression models were tried to obtain the best fitting curve:

a. Linear Regression
b. Polynomial Regression
c. Decision Tree
d. Random Forest
e. Neural Network

Graphs were plotted and R^2 values were compared to analyze which one fits the best.

All these approaches were implemented on the D-Wave Quantum Machine using the Ocean SDK.

Results:

Based on the above approaches, we obtained various kinds of results, ranging from relations between temperature and other parameters to binary values (Quantum Annealing). The important ones have been highlighted below:

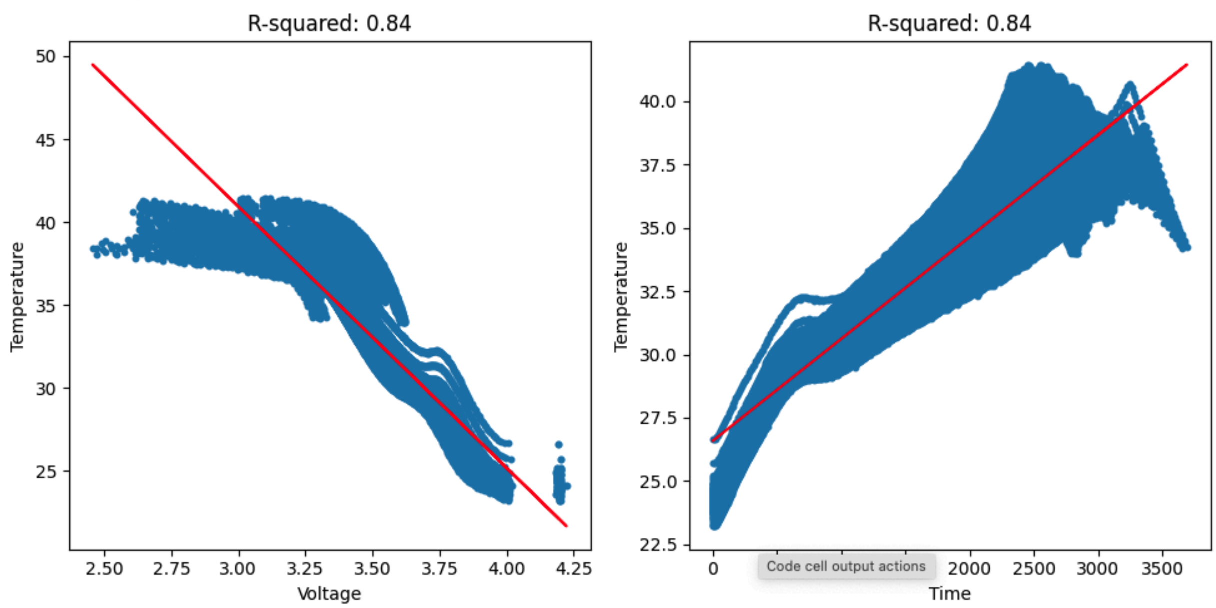
Approach 2: BQM Solving

```
   x1 x2 x3    energy num_oc.
0   0  0  0 -16.3967       1
7   0  0  1 -16.3911       1
3   0  1  0   -5.746       1
4   0  1  1  -5.7408       1
1   1  0  0  16.0522       1
6   1  0  1  16.0567       1
2   1  1  0  27.2734       1
5   1  1  1  27.2775       1
['BINARY', 8 rows, 8 samples, 3 variables]
```

Approach 3: Separate relations of Temp vs Each variable

a. Linear Regression

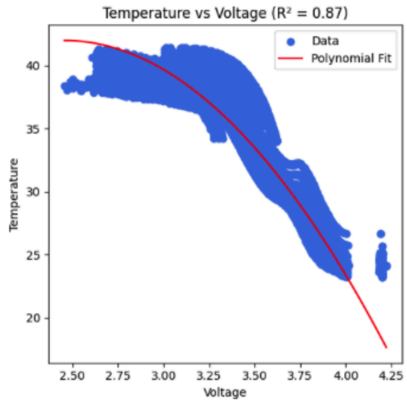
```
Temperature vs Voltage:
Equation: Temp = -15.7345 * V + 88.1279

Temperature vs Time:
Equation: Temp = 0.0040 * t + 26.5910
```

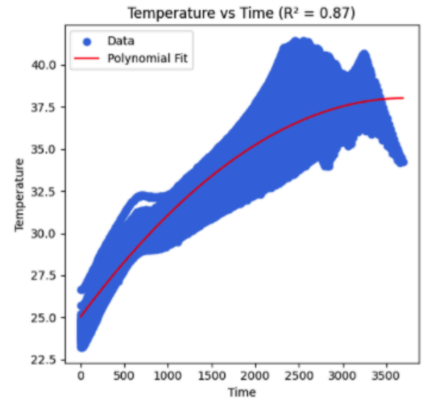



```
Predictions:
Temperature at Voltage = 4.5: 17.32
Temperature at Time = 2500: 36.66
```

b. Polynomial Regression

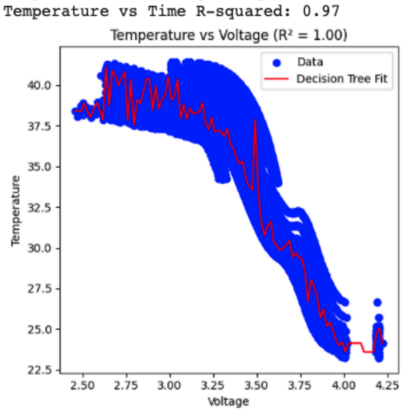
```
Temperature vs Voltage R-squared: 0.87
Temperature vs Time R-squared: 0.87
```

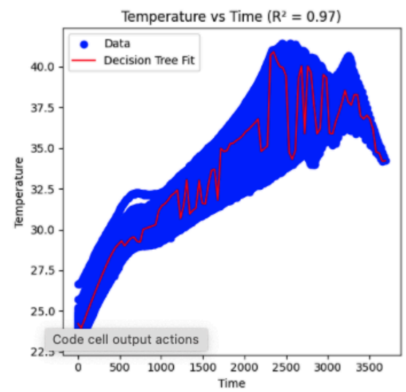




c. Decision Tree

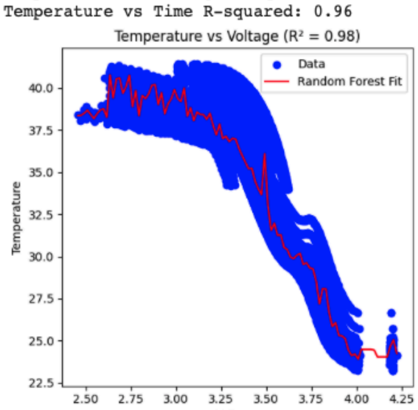
```
Temperature vs Voltage R-squared: 1.00
Temperature vs Time R-squared: 0.97
```

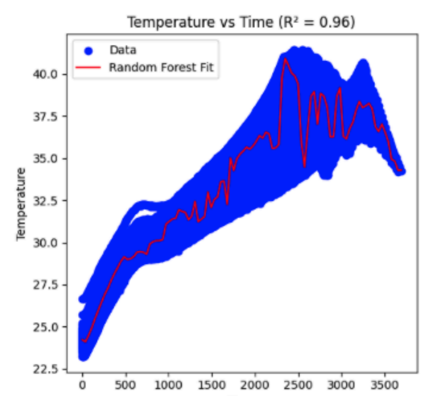




d. Random Forest

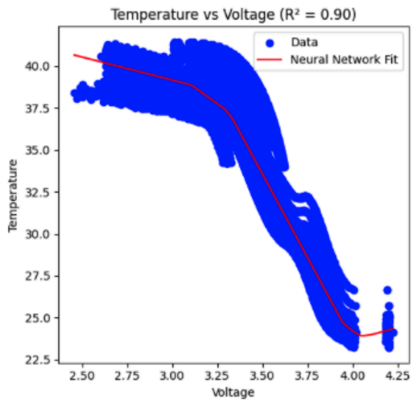
```
Temperature vs Voltage R-squared: 0.98
Temperature vs Time R-squared: 0.96
```

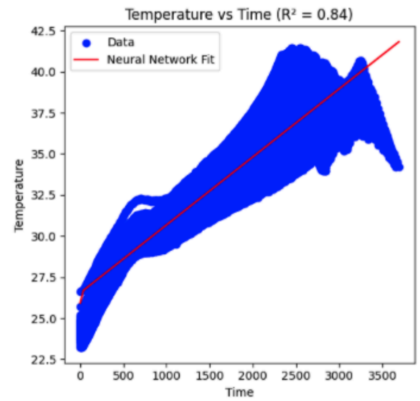




e. Neural Network

```
Temperature vs Voltage R-squared: 0.90
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:698: UserWarning: Training interrupt
  warnings.warn("Training interrupted by user.")
Temperature vs Time R-squared: 0.84
```





Discussion

We encountered several challenges while forming the QUBO for our problem. Key issues included:

1. **Complexity of Encoding Variables**: Encoding variables into binary proved more complex than anticipated, particularly when incorporating SoC penalties and various constraints. The lack of existing references made this process more challenging and highlighted gaps in our approach.
2. **Evaluating Quantum Annealing**:We are questioning whether quantum annealing is the best choice for this problem. Given the difficulties we faced, we are considering alternative quantum computing methods or other optimization techniques that may better address our needs.
3. **Dataset Complexity**: Our initial lack of thorough analysis of the dataset led to multiple pivots in our approach. Understanding the full complexity of the dataset is crucial, and future efforts will focus on a more comprehensive analysis before proceeding.

This iterative process underscores the nature of research—where repeated trials and adjustments are often required before reaching a viable solution. Though we are still working through these issues, the lessons learned are integral to refining our approach and ultimately achieving success in our project.

Future Work

Key Areas for Future Development:

1. **Refinement of QUBO Formulation**: We will enhance the QUBO formulation by incorporating additional constraints and refining penalty terms to better capture the complexities of battery temperature and SoC management.

2. **Optimization of Quantum Annealing Setup**: Further optimization of the quantum annealing process will be conducted to improve the accuracy and efficiency of the solutions obtained. This will involve fine-tuning the embedding process and exploring advanced quantum algorithms.

Conclusion:

This project aimed to optimize battery temperature management for electric vehicles using quantum annealing, focusing on enhancing battery performance, grid load management, and carbon emission reduction. We converted datasets, formulated a QUBO problem, and applied quantum annealing with D-Wave. Despite facing challenges in variable encoding, QUBO formulation, and constraint application, we gained valuable insights.

Our results are preliminary, revealing the need for further refinement and exploration. Future work will address these challenges, improve dataset analysis, and explore alternative optimization methods to advance battery management systems and support sustainable energy solutions.

References:

1. Effect of Temperature on Aging of battery
2. Keeping SoC within optimal ranges for battery operation
3. Quantum Optimization in Climate Science: Quantum Optimization
4. Smart Charging Strategies: [Smart charging strategy for electric vehicles]
5. Battery State Estimation Methods:
    a. Classical Model : Implementing the Extended Kalman Filter for SoH and SoC estimation.
    b. Quantum Models:
        i. quantum algorithm optimized stacking :
    c. AI Models: SoH estimation using machine learning and AI models
    - CNN and Random Forest
    - Support Vector Machine
    - Deep Learning
    - LSTM
6. QUBO and Quantum Annealing Approach:
    a. QUBO and Quantum Annealing
    b. A Practical Guide to Quantum Machine Learning and Quantum Optimization
    c. Quantum bridge analytics I: a tutorial on formulating and using QUBO models
    d. Quantum Annealing Problem with Constraints and Penalties Reference

7. Datasets used:
    a. Matlab dataset to csv : reference
    b. Experiments on Li-Ion batteries. "Battery Data Set". by: B. Saha and K. Goebel (2007)., NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA