Li-Ion Battery Temperature Forecasting Method: Case-Study

ARTI KHAPARDE, VAIDEHI DESHMUKH, VIDUSHI SHARMA, UTKARSH SINGH
Department of Electrical and Electronics Engineering,
Dr. Vishwanath Karad MIT World Peace University,
Kothrud, Pune,
INDIA

Abstract: - Monitoring and managing battery health is crucial for enhancing performance and lowering running expenses for electronic devices. This paper covers the Deep-learning-enabled temperature forecasting for Li-ion batteries, where they are tested independently. This research presents time series forecasting approaches to predict the temperature of the battery packs. In the proposed model, a Long Short-Term Memory (LSTM) and autoregressive integrated moving average (ARIMA) for predicting the battery temperature and beware of probable future temperatures beforehand to minimize the chances of overcharging and prevent the battery from crossing the threshold value above which battery's health characteristics might get hampered. The growing popularity of data-driven battery prognostics methods shows that ARIMA and LSTM are even when there aren't many prior details available about the batteries. Have a unique dataset of 34 lithium-ion battery packs for this challenge. In one way, the results imply that the existing ARIMA techniques offer interpreting data at various batteries. Having said that, LSTM model outcome recommend that the developed Univariate and Multivariate LSTM model provides finer prediction accuracy in the existence of greater diversification in data for one battery. Thus, try to generalize one forecasting model for each battery type depending on the model's performance.

Key-Words: - Lithium-ion batteries, ARIMA, Deep Learning, LSTM, Time Series, and battery temperature.

Received: March 11, 2023. Revised: October 8, 2023. Accepted: November 25, 2023. Published: December 31, 2023.

1 Introduction

Lithium-ion batteries are utilized in many different applications because of their high energy density, high power density, low pollution, and prolonged lifespan, [1]. While the current life test will take a while, the battery life will inevitably be evaluated thoroughly and frequently during development. Long battery life means that performance feedback is sometimes delayed by many months to years, as is the case with many chemical, mechanical, and electronic systems. Additionally, a battery's electrical performance will alter as its remaining useful life (RUL) increases over time. The temperature and electrochemical characteristics of batteries are influenced by their thermal effect during operation, which has a significant impact on their longevity and safety. Additionally, the heat accumulation and growing temperature inside the batteries could cause thermal runaway, which could burn or explode devices, [2]. To use batteries practically, it is crucial to forecast the temperature change of the batteries in electronic devices and electric vehicles and to study the thermal effect on the batteries.

Thus, irreversible heat and reversible heat are the two different types of heat. The relative contributions of irreversible heat and reversible heat to temperature change are significant for battery thermal management. Batteries have significant temperature fluctuations when in use, and high temperatures can compromise the stability of electronic devices, [3]. If we can anticipate the temperature, early warning can be given to prevent the hidden dangers of extreme temperatures. Battery temperature can fluctuate due to both physical and chemical processes, which are affected by the battery's size, composition, packing, and load circumstances. It is challenging to forecast the temperature change of batteries because of the intricacy of heat generation and the unpredictability of environmental factors. Researchers are concentrating on model-based methods for in-situ measurements employing internal sensors to ascertain the core temperature inside Li-ion cells, [4]. A variety of modeling techniques are applied, most frequently using the coupling or co-simulation of thermal and electrical phenomena.

Recently, data-driven techniques are slowly becoming more popular. Data-driven solutions based on statistical theories or artificial intelligence algorithms can deal directly with recorded data as a result of the sensor's outstanding precision in real-time battery temperature monitoring, [5]. Researchers provide various techniques, such as artificial neural network (ANN) approaches, multi-node equivalent circuit models, physic-chemical thermal models, and electro-thermal models, with a focus on large prismatic cells. However, they are not suitable for usage in a BTMS because of the enormous computing time and complexity of parameterization; instead, straightforward and workable solutions are required, [6]. The time and expense associated with gathering and producing data, as well as the decrease in prediction accuracy due to extrapolation throughout the prediction process due to a lack of training data, are other drawbacks of conventional models. For this reason, deep learning offers an appropriate way to simulate the complex and non-linear relationships among input and output values in the context of Li-ion cells. For predicting battery temperature, an LSTM is recommended in the suggested study due to these advantages. It can analyze lengthy input sequences without growing the network size. The proposed work aims to generalize a single forecasting model for every kind of battery. Observations of temperature, current, voltage, and time of the battery type at different start times. The proposed model performs better for most of the files, i.e., it gives a lower RMSE and can be concluded as the best-fit forecasting model for that battery type. ARIMA models may, however, account for a variety of patterns, including non-seasonal or seasonal fluctuations, non-linear or linear trends, and constant or changing volatility. Since ARIMA models, at most require fewer variables and assumptions, it becomes easy to put in the application and understand. Non-stationary time series are modeled. It meets expectations for short-term forecasts. To generalize the prediction, necessity is not more than the time series historical data. The key contribution of the proposed model is discussed below:

- The temperature of a Li-ion battery is predicted using an ARIMA-based LSTM under a variety of circumstances.
- A real-time dataset is gathered and pre-processed to improve the data quality for effective prediction performance.
- The pre-processed data are further proceeding for the prediction process using LSTM, which can remember information for extended periods.
- The performance of LSTM is improved through the use of ARIMA, which uses the series data to provide a better understanding and prediction process.

2 Problem Formulation

Several approaches were developed to manage the power flows without affecting the battery lifecycle. A few of them were briefly discussed in the following subsection.

2.1 Related Work

The authors in [7], developed a machine-learning model to address the state-of-health (SoH) prediction for Li-ion batteries, which are employed as power sources in electric trucks. The authors propose the use of supervised learning for estimating the battery's SoH to improve the battery's accessibility at the forklift process, demonstrating the capabilities of ARIMA in scenarios with very little prior knowledge about the batteries and the utilization of data-directed methodologies for increased forecasting process. However, the model has an impact of poor prediction performance. The authors in [8], developed Lithium Ion battery temperature variations that can be tracked using a model called the convolutional transformer (Convtrans), which yields pleasing results because the battery temperature can be generally represented as a time series. On the other hand, Convtrans forecasts 24 times more temperature data than single-step time series forecasting while maintaining a high level of accuracy, although it takes six times longer to operate.

The authors in [9], suggested a data-driven strategy for lithium-ion battery health management presented in this removes the need for battery physical models to estimate SOH. The authors continuously track the tension, current, charge, cell temperature, and ambient temperature while using iron phosphate lithium-ion batteries and subjecting them to charge-discharge cycles based on conventional IEC and ISO profiles. However, the model has more time consumption and improper prediction. The authors in [10], suggested direct current (DC) resistance, hundreds of capacity, and electrochemical impedance spectroscopy measurements taken under various conditions of health, temperature, and state of charge (SOC) are used to predict capacity from EIS using a variety of machine learning models, including linear, random forest, Gaussian process, and artificial neural network regression. However, in practical applications, putting cells through equilibration cycles is not feasible. The authors in [11], developed a collection of rudimentary models to take the place of the extreme learning machine's active functions to improve generalization performance. Because the model parameters and initial SOC in these "rough models" are randomly selected within a predetermined range, little battery-specific knowledge is required. However, it is not necessary, and the range can be discovered by studying a datasheet or drawing on experience.

2.1.1 Sub-subsection

When including a sub-subsection you must use, for its heading, small letters, 11pt, left justified, bold, Times New Roman as here.

3 Problem Solution

3.1 Battery Data Description and Pre-Processing

The dataset contains time series data of 34 unique Lithium-ion battery types. The data has been measured at equally spaced intervals of time that are approximately 6 seconds. To comprehend the entire data, there is a metdata.csv file that contains information about all the battery types and the files associated with each battery type. Fundamentally, the metadata file contains operational profiles (charge, discharge, and impedance) of 34 unique battery types. Within each battery type, there are at least 62 files. These files are nothing but readings obtained from the battery at different instances of time for charge, discharge, and impedance. But the battery temperature performance during charging is something we are worried about. Therefore, we shall extract every file related to the particular battery type that has been charged. For this article, we will examine a collection of lithium-ion batteries and evaluate how well the forecasting model performs on them. This is because the readings of the different batteries were taken in sets of four batteries each. So, if we can generalize a forecasting model individually for a set of four batteries that have been tested together, the same process can be replicated on other sets of lithium-ion batteries. Eventually, each battery type can be concluded to have one forecasting model that is best fit for that data.

Four Li-ion batteries (# 45–48) were grouped and subjected to three distinct operating descriptions (charge, discharge, and impedance) in an environment with a temperature of four degrees Celsius. At a predefined load current level of 1A, the discharge was forced to halt at 2V, 2.5V, 2.2V, and 2.7V for each of the batteries 45, 46, 47, and 48. Charging was done in a constant current (CC) mode at 1.5A until the battery voltage reached 4.2V and then continued in a constant voltage (CV) mode until the charge current dropped to 20mA. Electrochemical impedance spectroscopy (EIS) frequency sweeps in the 0.1Hz–5 kHz range were used to estimate impedance. The tests continued until the capacity dropped to 1.4 amps (30 percent fading). Bear in mind that the capacity was small for numerous discharge runs, [12]. The causes that lead to this behavior are yet to be studied properly. The fields for charge:

Table 1. The instrument used for Measurements

Voltage_measured	Battery terminal voltage (Volts)
Temperature_measured	Battery temperature (degree C)
Current_measured	Battery output current (Amps)
Voltage_charge	Voltage measured at charger (Volts)
Current_charge	Current measured at charger (Amps)
Time	Time vector for the cycle (sec)

Table 1 shows the Instrument Used For Measurements. It is good practice to deal with null values before commencing model implementation. This step is to ensure we do not get any faulty outputs or errors during model implementation. The null values in the dataset might be because of faulty battery readings. For handling the null values: If the ratio of the sum of the features (in our case: Temperature measured) to the total number of observations is found to be less than 10%, then we drop those null values. Hence, if the kurtosis of the feature is between -1 and 1 that is platykurtic, then we take the mean of the observation and replace it in place of missing values. If the kurtosis of the feature is not between -1 and 1 that is leptokurtic, then we take the median of the observations to put in place of missing values.

Examine the relationships between the various features and our target variable, the measured temperature, to determine whether features have a strong link with the temperature. We find that there is a large positive correlation between measured current and temperature and a substantial negative correlation between time and measured temperature and current charge. This has been verified for every cycle of charging the battery in concern. This confirms the battery to be having similar characteristics during its operational profile for different cycles. Owing to this Pearson's Correlation obtained, we consider the current charge as the other independent variable for Multivariate LSTM forecasting. To train our model, we split the dataset in an 80-20% ratio. We keep aside 20% of the data for validation. We resample the data frame to contain only the features that concern, i.e., Time, Temperature measured, and Current Charge. Figure 1 shows Pearson's Correlation Matrix.

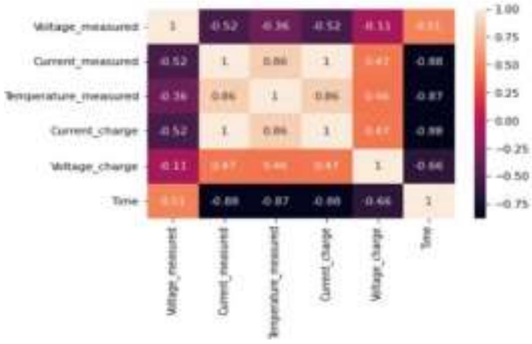


Fig. 1: Pearson's Correlation Matrix

3.2 ARIMA

To create a forecast model using time series analysis, the fitting ARIMA model is discovered from an input y to the criterion Þ. This process is known as integrated (I), autoregressive (AR), and modeling average (MA).

By transforming the data, the time series can be rendered stationary if its statistical characteristics, such as its variance and mean are not constant. Differentiating is a straightforward transformation that is accomplished using the following equation; nevertheless, it is important to emphasize that the appropriate data modification method depends on the data:

$$yt = Yt - Yt\text{–}1 \tag{1}$$

Where differenced, stationary time series is denoted by y. The stationary estimator values, ŷt, are computed from this stationary time series. The model that is given implies $ARIMA$ (p, d, q), where p represents the AR-term, q denotes MA-term, and d denotes the number of differencing operations performed. An organized method for defining these variables ought to be the interpretation of autocorrelations and partial autocorrelations of y. The correlation between yt and $yt\text{–}k$ is an autocorrelation of lag k at y. The amount of correlation among yt and $yt\text{–}1$ cannot be explained by the fact that yt is associated with $yt\text{–}1$ which is correlated with $yt\text{–}2$, and the remaining chained correlation steps up to the step $corr(yt\text{—i—}1, yt\text{– i})$ comprise the partial autocorrelation of y at lag i. For a detailed examination of the subject, several lecture notes and reference materials are accessible.

3.3 LSTM

LSTM is short for long short-term memory networks utilized in Deep Learning. Constituting a variation of recurrent neural networks (RNNs), which can understand long-term dependencies, mainly in sequence forecast issues, LSTM is designed according to the created dataset that is the input of LSTM is stated as voltage, current, and time as well, and the output is termed as maximum voltage. The key purpose behind LSTM is the foundation of memory cells, accountable for accumulating and obtaining details with time. An input gate, a forget gate, and an output gate make up the three primary parts of these memory cells. Figure 2 shows the LSTM cell.

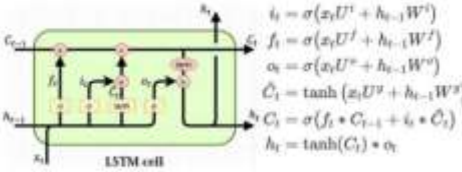

C_{t-1}
h_{t-1}
x_t
h_t
C_t
f_t
i_t
o_t
$\tilde{C}_t$
tanh
LSTM cell
$i_t = \sigma(x_t U^i + h_{t-1} W^i)$
$f_t = \sigma(x_t U^f + h_{t-1} W^f)$
$o_t = \sigma(x_t U^o + h_{t-1} W^o)$
$\tilde{C}_t = \tanh(x_t U^g + h_{t-1} W^g)$
$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t)$
$h_t = \tanh(C_t) * o_t$

Fig. 2: LSTM cell

An LSTM network that is specifically created to operate with univariate time series data is known as a "Univariate LSTM." A single variable is monitored over time in univariate time series data, such as stock prices, temperature readings, or daily sales numbers. A sequence of historical data points from a single variable serves as the input to a univariate LSTM network, and the objective at hand is typically to predict the value that will come next in the sequence. The input sequence is processed by the LSTM network, which then discovers patterns and dependencies in the data to produce a forecast for the following time step.

Multivariate time series data consists of numerous variables seen simultaneously at each time step, in contrast to univariate time series, which only include a single variable that is monitored across time. With multivariate LSTM, the objective is often to predict one or more variables at the following time step using a sequence of historical data points from several variables as the network's input. One input sequence per variable is often used in the design of a multivariate LSTM, which is then processed by various LSTM branches or shared LSTM layers.

4 Evaluation Methods

Root-mean-square error (RMSE) of temperature prediction is the metric used to evaluate our approaches, which takes into account their applicability to compute the precision in the practical implementation. RMSE is chosen as the loss function because it is appropriate for the evaluation of regression models and, consequently, for the evaluation of time series prediction models.

To determine the value of $RMSE$, all temperature readings up to the most recent observation and estimator pair are considered. Where y is the desired (correct) output and ŷ is the estimator. The RMSE number is in the identical unit as the forecasted value, which is the superiority of this method. This statistic is always positive, with lower values implying higher performance. In contrast to MSE, this makes it simpler to apprehend.

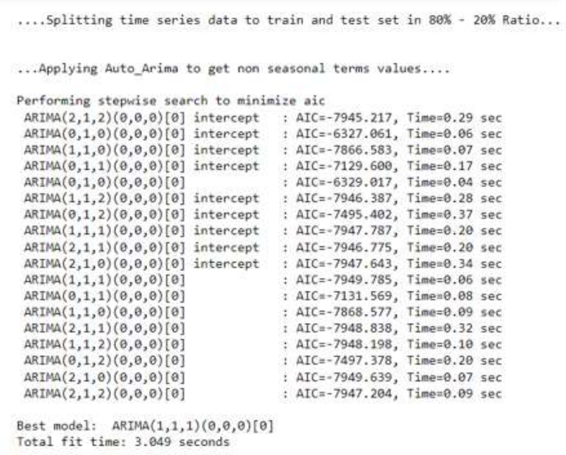
$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\tilde{y}_i - y_i)^2}{n}} \qquad (2)$$

4.1 Results and Discussion

To verify the implementation method efficacy, we selected four sets of Lithium-ion batteries' (#B0045, B0046, B0047, and B0048) 34 batteries worth of time series.

4.1.1 ARIMA Model -Results

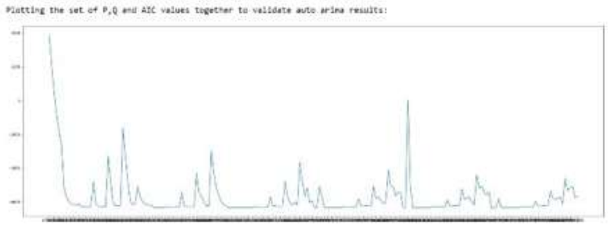
Although there was a reason for including the seasonal component, as the temperature time series graph revealed, we used the fundamental non-seasonal ARIMA. However, due to the length of observations not being long enough to ensure seasonality or some clear pattern, we picked fundamental ARIMA for model buildout. Mainly, the seasonality is presently unclear for the temperature measured. For implementing the ARIMA model, we require optimal values of (p, d, q). Instead of doing the laborious task of manually checking the time series for stationarity (d) and representing the PACF and ACF plots to get the optimal p and q values. We make use of a pre-defined Auto-Arima library that automatically checks the Akaike information criterion (AIC) against all the possible integration of (p, d, q) values and returns the best-suited (p, d, q) values for the time series data given as input. AIC approximates models comparatively. It is a solitary number score that can be utilized to find out among several models which is, in all likelihood, to be the finest model for a particular data set. A lower AIC score is better. Figure 3 shows the Auto Arima implementation e.g.



```
....Splitting time series data to train and test set in 80% - 20% Ratio...

...Applying Auto_Arima to get non seasonal terms values....

Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-7945.217, Time=0.29 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-6327.061, Time=0.06 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-7866.583, Time=0.07 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-7129.600, Time=0.17 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-6329.017, Time=0.04 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=-7946.387, Time=0.28 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=-7495.402, Time=0.37 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-7947.787, Time=0.20 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-7946.775, Time=0.20 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=-7947.643, Time=0.34 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=-7949.785, Time=0.06 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=-7131.569, Time=0.08 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=-7868.577, Time=0.09 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=-7948.838, Time=0.32 sec
 ARIMA(1,1,2)(0,0,0)[0]             : AIC=-7948.198, Time=0.10 sec
 ARIMA(0,1,2)(0,0,0)[0]             : AIC=-7497.378, Time=0.20 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=-7949.639, Time=0.07 sec
 ARIMA(2,1,2)(0,0,0)[0]             : AIC=-7947.204, Time=0.09 sec

Best model:  ARIMA(1,1,1)(0,0,0)[0]
Total fit time: 3.049 seconds
```

Fig. 3: Auto Arima implementation e.g

However, as a precautionary measure to validate the (p, d, q) values obtained from the Auto-Arima library, we made a function that generates every possible combination of (p, d, q) values and fits the Arima model to obtain the respective AIC values. We take the (p, d, q) value from here that gave the minimum AIC value and cross-check it with the Auto-Arima results to be sure that we have the best (p, d, q) values. We also plotted a graph of AIC scores against the (p, d, q) combination values to ensure that, ultimately the graph is decreasing and the lowest point in the graph is the optimal point we are looking for. Figure 4 shows the Validation plot of AIC vs (p,d,q) values.



Plotting the set of P,Q and AIC values together to validate auto arima results:

Fig. 4: Validation plot of AIC vs (p,d,q) values

4.2 LSTM Model -Results

In the LSTM-based approach, we need to consider a look back window size period for the model to consider to make forecasts. By rule of thumb, this windowsize should be greater than or equal to one seasonalityperiod. However, because the time series index data that we have available is in seconds and not in date time format, we cannot plot seasonality graphs to check the length of the seasonal period (m) from the decomposition graphs. Instead of this, we have gained insights from the PACF graph. This graph shows the correlation of today's value with the lag values. The lag value that gives the highest correlation is the optimal value to be considered. By trying different window size values for each battery type, we chose the value of window size that, on average gave the best performance. When we observed the PACF graph, there were high autocorrelations with many time steps in the past. We found out the best value forwindow size only by trial and error. After individuallyrunning Univariate LSTM and Multivariate LSTM on a set of window size values, on average we found a window size of 1 to work best for Univariate LSTM and a window size of 2 to work best for Multivariate LSTM. Three LSTM layers comprise the sequential model; each LSTM layer is succeeded by a dropout layer and, in the end, by a dense layer. The Adam Optimization algorithm is employed to determine the ideal set of parameters to reduce the cost function. We are training the Univariate LSTM model on 30 epochs and the Multivariate LSTM model on 80 epochs to learn the fine tunes of the data and provide accurate results. Figure 5 shows the PACF Plot.

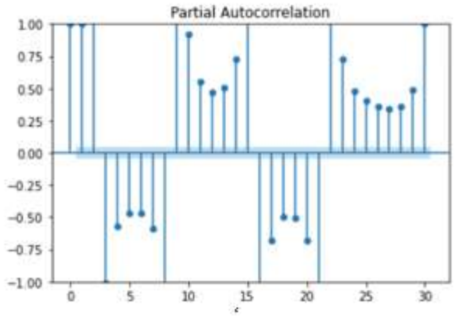


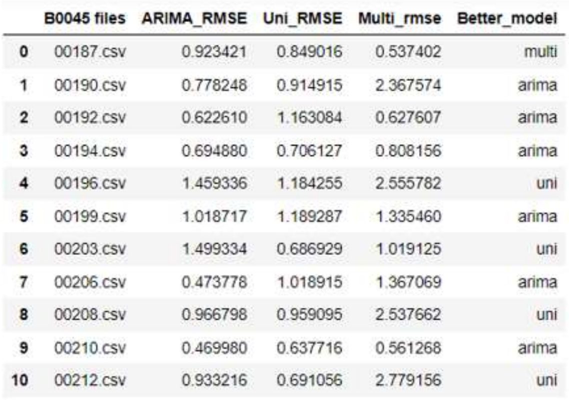
Fig. 5: PACF Plot

From this PACF plot, e.g., of a particular battery, we can observe the possible window size values that can be considered.

4.3 Forecasting Models Implementation

We have implemented the discussed forecasting models individually on each battery type. We ran the models individually on all the charging files associated with a particular battery type and found the model's performance using RMSE. We now have an entire table comprising all the charging files associated with a battery type and the respective RMSE values of all the models on all files separately. After that, we combine the data to determine which model was selected the most frequently for a specific kind of battery. When it comes to predicting future temperature values, the model that was selected the most times can be considered the most suitable model for that particular type of battery.

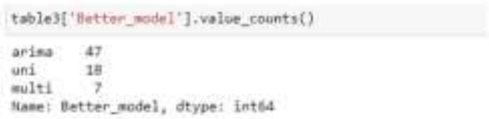
4.4 BATTERY TYPE B0045 Analysis



	B0045 files	ARIMA_RMSE	Uni_RMSE	Multi_rmse	Better_model
0	00187.csv	0.923421	0.849016	0.537402	multi
1	00190.csv	0.778248	0.914915	2.367574	arima
2	00192.csv	0.622610	1.163084	0.627607	arima
3	00194.csv	0.694880	0.706127	0.808156	arima
4	00196.csv	1.459336	1.184255	2.555782	uni
5	00199.csv	1.018717	1.189287	1.335460	arima
6	00203.csv	1.499334	0.686929	1.019125	uni
7	00206.csv	0.473778	1.018915	1.367069	arima
8	00208.csv	0.966798	0.959095	2.537662	uni
9	00210.csv	0.469980	0.637716	0.561268	arima
10	00212.csv	0.933216	0.691056	2.779156	uni

Fig. 6: Forecasting Models comparison table

Figure 6 is a view of the top 10 cells of the B0045 Battery Type forecasting model's performance. You can see the RMSE values of the different models on each charging file and the model preferred for each file.



```
table3['Better_model'].value_counts()

arima    47
uni      18
multi     7
Name: Better_model, dtype: int64
```

Fig. 7: Value Count for B0045 Best Forecasting Model

We can observe from Figure 7 the Arima model performed better most of the time compared to the other two models. We can conclude from this that ARIMA is the best-fit forecasting model for B0045. It is important to note that these models chosen **are** based on the current scenario of historical data it has. As these are Time series forecasting models, as and when it gets new data and the data horizon expands, the model which is performing best currently for a particular battery type doesn't need to continue to do so in the future. It may or may not change. Figure 8 shows the Visualizing RMSE values of B0045 charging files.

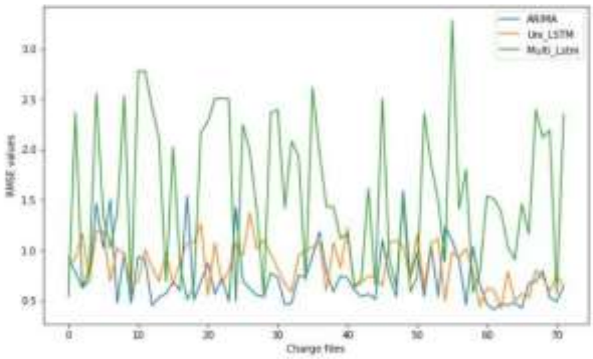
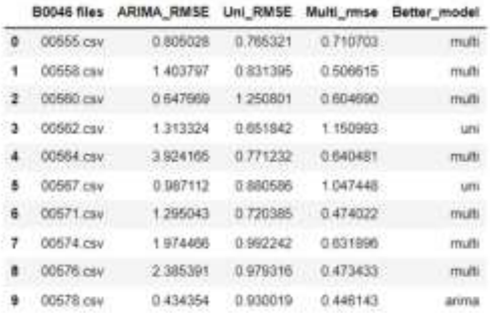


Fig. 8: Visualizing RMSE values of B0045 charging files

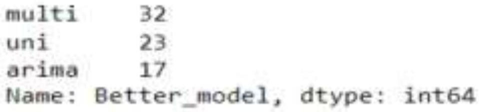
4.5 BATTERY TYPE B0046 Analysis



	B0046 files	ARIMA_RMSE	Uni_RMSE	Multi_rmse	Better_model
0	00555.csv	0.805028	0.765321	0.710703	multi
1	00558.csv	1.403797	0.831395	0.506615	multi
2	00560.csv	0.647669	1.250801	0.604690	multi
3	00562.csv	1.313324	0.651842	1.150993	uni
4	00564.csv	3.924165	0.771232	0.640481	multi
5	00567.csv	0.987112	0.880586	1.047448	uni
6	00571.csv	1.295043	0.720385	0.474022	multi
7	00574.csv	1.974466	0.992242	0.631896	multi
8	00576.csv	2.385391	0.979316	0.473433	multi
9	00578.csv	0.434354	0.930019	0.446143	arima

Fig. 9: B0046 Forecasting Models Comparison Table

Figure 9 is a view of the top 10 cells of the B0046 Battery Type forecasting model's performance. You can see the RMSE values of the different models on each charging file and the model preferred for each file.



```
multi     32
uni       23
arima     17
Name: Better_model, dtype: int64
```

Fig. 10: Count of B0046 Forecasting models performance

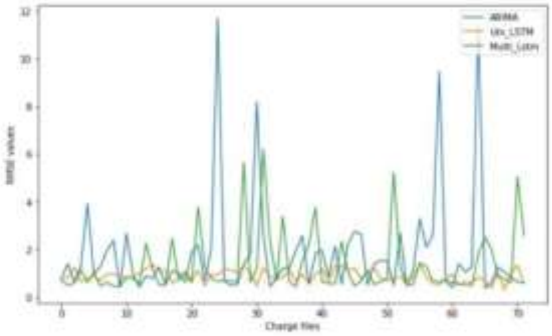
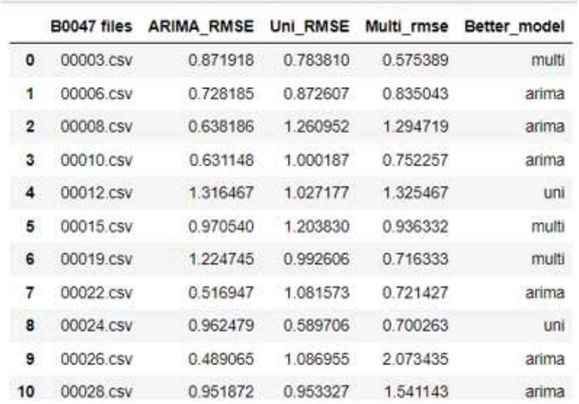


Fig. 11: RMSE plot of B0046 charging files

Figure 10 shows the Count of the B0046 Forecasting model's performance. This graph plots the RMSE values of all the files in the B0046 battery type. Overall, we can observe that Multivariate LSTM gives the lowest RMSE for most of the models. Implies multivariate LSTM is best for B0046 Battery Type. Figure 11 shows the RMSE plot of B0046 charging files.

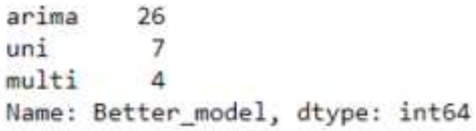
4.6 BATTERY TYPE B0047 Analysis



	B0047 files	ARIMA_RMSE	Uni_RMSE	Multi_rmse	Better_model
0	00003.csv	0.871918	0.783810	0.575389	multi
1	00006.csv	0.728185	0.872607	0.835043	arima
2	00008.csv	0.638186	1.260952	1.294719	arima
3	00010.csv	0.631148	1.000187	0.752257	arima
4	00012.csv	1.316467	1.027177	1.325467	uni
5	00015.csv	0.970540	1.203830	0.936332	multi
6	00019.csv	1.224745	0.992606	0.716333	multi
7	00022.csv	0.516947	1.081573	0.721427	arima
8	00024.csv	0.962479	0.589706	0.700263	uni
9	00026.csv	0.489065	1.086955	2.073435	arima
10	00028.csv	0.951872	0.953327	1.541143	arima

Fig. 12: B0047 Forecasting Models Comparison Table

Figure 12 shows the B0047 Forecasting Models Comparison Table. This table shows the topmost cells of the B0047 comparison table for understanding. Figure 13 shows the Value Count for B0047 Best Forecasting models.



```
arima    26
uni       7
multi     4
Name: Better_model, dtype: int64
```

Fig. 13: Value Count for B0047 Best Forecasting Models

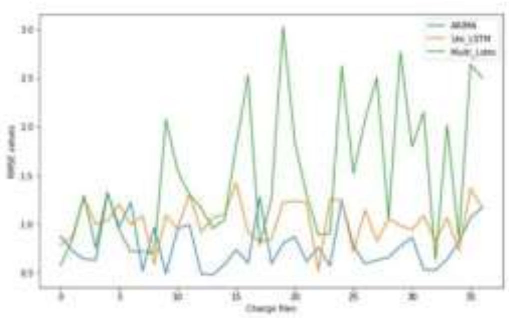
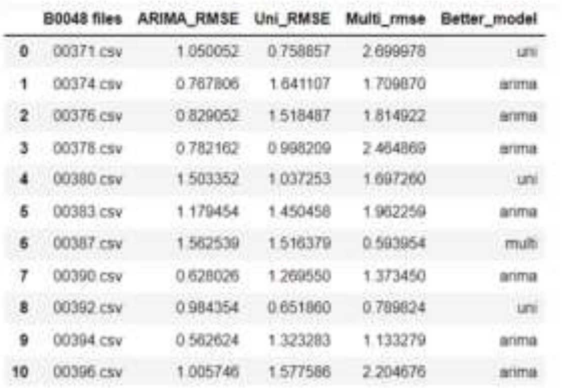


Fig. 14: RMSE Visualization plot of B0047 charging files

Figure 14 shows the RMSE Visualization plot of B0047 charging files. This graph plots the RMSE values of all the files in the B0047 battery type. Overall, we can observe that ARIMA gives the lowest RMSE for most of the models. Implies ARIMA is best for the B0047 Battery Type.

4.7 BATTERY TYPE B0048 Analysis

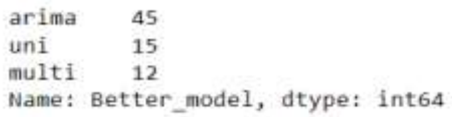
This graph plots the RMSE values of all the files in the B0047 battery type. Overall, we can observe that ARIMAgives the lowest RMSE for most of the models. Implies ARIMA is best for the B0047 Battery Type. Figure 15 shows the B0048 Forecasting models' comparison table.



	B0048 files	ARIMA_RMSE	Uni_RMSE	Multi_rmse	Better_model
0	00371.csv	1.050052	0.758857	2.699978	uni
1	00374.csv	0.767806	1.641107	1.709870	arima
2	00376.csv	0.829052	1.518487	1.814922	arima
3	00378.csv	0.782162	0.998209	2.464869	arima
4	00380.csv	1.503352	1.037253	1.697260	uni
5	00383.csv	1.179454	1.450458	1.962259	arima
6	00387.csv	1.562539	1.516379	0.593954	multi
7	00390.csv	0.628026	1.269550	1.373450	arima
8	00392.csv	0.984354	0.651860	0.789824	uni
9	00394.csv	0.562624	1.323283	1.133279	arima
10	00396.csv	1.005746	1.577586	2.204676	arima

Fig. 15: B0048 Forecasting models' comparison table

This table shows the topmost cells of the B0048 comparison table for understanding. Figure 16 shows the Vale count for the best B0048 forecasting models.



```
arima    45
uni      15
multi    12
Name: Better_model, dtype: int64
```

Fig. 16: Count for best B0048 forecasting models

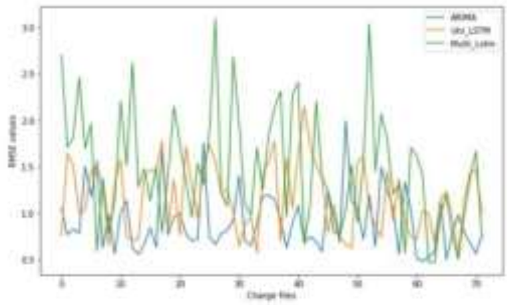


Fig. 17: RMSE plot for B0048 charging files

Figure 17 shows the RMSE plot for B0048 charging files. This graph plots the RMSE values of all the files in the B0048 battery type. Overall, we can observe that ARIMA gives the lowest RMSE for most of the models. Implies ARIMA is best for the B0048 Battery Type. Then, the performance of LSTM is compared to some other traditional approaches like ANN and Deep Neural Network (DNN).

Table 2. Performance metrics comparison

Metrics	Proposed	DNN	ANN
Accuracy	0.96	0.85	0.76
Precision	0.94	0.84	0.75
Recall	0.95	0.83	0.74
Specificity	0.97	0.86	0.77
NPV	0.97	0.85	0.76
FPR	0.02	0.11	0.17

Table 2 demonstrates the proposed and traditional model comparisons. This proves that the proposed model provides an effective prediction performance by the use of ARIMA.

4.8 Discussion

In the first section, the relevance of the non-seasonal ARIMA model to the batteries in temperature prediction. Nonetheless, the ensued loss function RMSE value denoted the potential to improve the model and perhaps an opportunity to test LSTM-based learning methods given the nature of batteries to somewhat emulate patterns based on its past values. We resampled the dataset to consider only the features that relate strongly to our target variable and implemented Univariate and Multivariate LSTM separately on all the four Battery types (B0045, B0046, B0047, B0048) that we have considered for analysis. Based on a comparison of the performance values of three predicting approaches for overall charging data for that battery type, we attempted to generalize one forecasting model for each battery type.

Due to the time series' short length and the fact that observations are collected in seconds, they may eventually exhibit seasonality, which points to a flaw in our models. It will be important to make more datasets with reasonably long time series available to address forecasting accuracy problems, which are well-known in the field of batteries. The drawback of Multivariate LSTM forecasting is that we will need future values of the independent variable as well to forecast the subsequent value of our desired variable. Unless we have values for the upcoming independent variable, we will only be able to forecast a one-time step in the future of the target variable. The RMSE of B0045 is 1.5, B0046 is 2, B0047 is 1.3, and B0048 is 2. The proposed model provides 0.96 accuracy and 0.02 false positive rate.

5 Conclusion

This study has demonstrated the importance of LSTM and ARIMA for forecasting battery temperature, given that adequate battery data is available. A way to generalize a forecasting model for each battery type has been discussed based on the performance of the models on all the charging files associated with a particular battery type. The method's primary flaw is the very short time series data, which may reveal seasonality or other variations ultimately related to the battery chemistry.

The use cases of this application are many in inaccessible areas where a person cannot repeatedly intervene to record the temperature. This application is able to give caution warnings in case the threshold temperature is going to be crossed in x time steps in the future. Business value is that provided there is some way to collect the temperature values of batteries, this application can be implemented to caution the user before threshold breach. This can save time and money for the consumer to prevent any system malfunctioning take care of such a situation well in advance and ensure smooth functioning. This model has an impact on more complex design and over fitting issues in the prediction process, thus causing harder prediction performance. In future work, a novel technique will be integrated into the deep learning model to avoid over fitting issues with improve its performance.

References:

[1] R. Mo, X. Tan, F. Li, R. Tao, J. Xu, D. Kong, Z. Wang, B. Xu, X. Wang, C. Wang, and J. Li, “Tin-graphene tubes as anodes for lithium-ion batteries with high volumetric and gravimetric energy densities,” *Nature communications*, vol. 11, no. 1, pp. 1374, 2020.

[2] M. Elmahallawy, T. Elfouly, A. Alouani, and A. Massoud, “A Comprehensive Review of Lithium-Ion Batteries Modeling, and State of Health and Remaining Useful Lifetime Prediction,” *IEEE Access*, 2022

[3] H. Zhang, L. Wang, and X. He, “Trends in a study on thermal runaway mechanism of lithium-ion battery with LiNixMnyCo1-x-yO2 cathode materials,” *Battery Energy*, vol. 1, no. 1, pp. 20210011, 2022.

[4] A. Elgammal, and T.Ramla, “Optimal model predictive frequency control management of grid integration PV/wind/FC/storage battery based smart grid using multi objective particle swarm optimization MOPSO,” *WSEAS Transactions on Electronics*, vol. 12, pp. 46-54, 2021 https://doi.org/10.37394/232017.2021.12.7.

[5] T. Sathapornbumrungpao, D. Moonjud, N. Donjaroennon, U. Leetond, S. Nuchkum, and T. Chaisirithungnaklang, “Battery Management System Using Relay Contactor by Arduino Controller for Lithium-Ion Battery,” In *International Conference on Clean Energy and Electrical Systems*, pp. 153-162, 2023, Singapore: Springer Nature Singapore.

[6] M. Fethi, B. Messaoud, and B. Mohammed, "Cube Satellite Battery Charger Regulator Design," WSEAS Transactions on Electronics, vol. 13, pp. 142-146, 2022 https://doi.org/10.37394/232017.2022.13.19.

[7] M. Huotari, S. Arora, A. Malhi, and K. Främling, “A dynamic battery state-of-health forecasting model for electric trucks: li-ion batteries case-study,” In *ASME International Mechanical Engineering Congress and Exposition*, vol. 84560, pp. V008T08A021, 2020. American Society of Mechanical Engineers.

[8] L. Yao, S. Xu, A. Tang, F. Zhou, J. Hou, Y. Xiao, and Z. Fu, “A review of lithium-ion battery state of health estimation and prediction methods,” *World Electric Vehicle Journal*, vol. 12, no. 3, pp. 113, 2021.

[9] P. A. Christensen, P. A. Anderson, G. D. Harper, S. M. Lambert, W. Mrozik, M. A. Rajaeifar, M. S. Wise, and O. Heidrich, “Risk management over the life cycle of lithium-ion batteries in electric vehicles,” *Renewable and Sustainable Energy Reviews*, vol. 148, pp. 111240, 2021.

[10] P. Gasper, A. Schiek, K. Smith, Y. Shimonishi, and S. Yoshida, “Predicting battery capacity from impedance at varying temperature and state of charge using machine learning,” *Cell Reports Physical Science*, vol. 3, no. 12, 2022.

[11] X. Tang, K. Yao, B. Liu, W. Hu, and F. Gao, “Long-term battery voltage, power, and surface temperature prediction using a model-based extreme learning machine,” *Energies*, vol. 11, no. 1, pp. 86, 2018.

[12] NASA Battery Dataset, [Onlin]. https://www.kaggle.com/datasets/patrickfleith/nasa-battery-dataset (Accessed Date: November 4, 2023).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.