Time Series Forecasting on Individual Household Electricity Consumption Data

https://github.com/skamble2/CS-584-Final-Project

Soham Kamble (A20517098)
Illinois Institute of Technology
Chicago, Illinois
skamble2@hawk.iit.edu

Pranit Kotkar (A20512027)
Illinois Institute of Technology
Chicago, Illinois
pkotkar1@hawk.iit.edu

Bhavya Chawla (A20516957)
Illinois Institute of Technology
Chicago, Illinois
bchawla@hawk.iit.edu

***Abstract*—This study explores multi-step time series forecasting for individual household electricity consumption, employing models like Encoder-Decoder LSTMs, CNN-LSTM Encoder-Decoder, and Conv-LSTM Encoder-Decoder. Using a multivariate time series dataset spanning four years, the research aims to predict total active power usage over the next seven days. Results show promising efficiency: Encoder-Decoder LSTM displays accuracy on specific days, CNN-LSTM captures trends with fine-tuning needs, and Conv-LSTM adeptly handles data complexities. While the models show promise, limitations necessitate refinement, calibration, and consideration of outlier events. This study contributes insights for advancing time series forecasting for household power consumption, laying the foundation for innovative predictive models.**

Keywords—LSTM, time series forecasting, power consumption, Encoder-Decoder LSTM, Conv-LSTM

I. Introduction

This Energy and resource consumption are at an all-time high, making it possible and necessary to have accurate forecasting of not just the power generation but also the power consumption. The rising usage, along with the availability of data from the widespread adoption of smart grids and meters for power and electricity, is a blessing. It allows the implementation of multiple existing predictive models for forecasting.

The focus was on multi-step forecasting, as it involves predicting multiple future time steps, particularly Long Short-Term memory. The survey talks about the models studied and implemented, like Encoder-Decoder LSTMs, CNN-LSTM Encoder-Decoder, and Conv-LSTM Encoder-Decoder, to find the best one that suits the goal of predicting total active power usage for each day over the next seven days. A comprehensive approach by taking into account various factors and multiple input variables - multivariate. As per the requirements and goals, the models worked on an individual household electric power consumption dataset. It is a multivariate time series dataset that describes the electricity consumption for a single household over four years (Dec 2006 to Nov 2010). The current data was being recorded per minute, but for this model, it was more practical to down-sample it to daily totals.

The model's results display efficiency and effectiveness. The Encoder-Decoder LSTM demonstrates accuracy on select days; the CNN-LSTM catches visible trends but requires more fine-tuning, and the Conv-LSTM deals well with the complexities of high variability in observed data. The need for fine-tuning and testing these models is very important for improving model accuracy and making them more generalized.

II. Literature Survey

This paper explores time series forecasting on individual household electricity consumption data amid rising demands and with implementing models such as Encoder-Decoder LSTM, conv-LSTM, and CNN-LSTM. The dataset, consisting of data collected over four years and preprocessed for missing values and split into training and testing sets, was introduced with new features to represent the power consumption of the entire circuit.

The Encoder-Decoder LSTM model is designed with layers for input computation, feature vector duplication, and time-distributed dense layers. The performance analysis reveals accurate predictions on certain days but inconsistencies on others, indicating the need for additional calibration. The root mean squared error (RMSE) plot highlights variation in prediction errors, suggesting the model struggles to capture underlying patterns, possibly leading to overfitting.

The CNN-LSTM model incorporates convolutional neural networks and LSTM layers for feature extraction and sequence handling. The actual vs. predicted values exhibit deviations, emphasizing the model's potential for refinement. Weekly seasonality is observed in the cyclic pattern, and the training vs. validation loss graph indicates effective generalization.

The Conv-LSTM model utilizes the Adam optimizer and reshapes input data into subsequences. The comparison between actual and predicted values reveals the model's struggle to capture high variability, suggesting limitations in pattern recognition. The RMSE plot emphasizes higher prediction errors on specific days, indicating oversight of outlier events.

In conclusion, the study leverages advanced deep learning models for time series forecasting in household electricity consumption. While achieving promising results, the models exhibit limitations in capturing variability and underlying patterns. The need for additional refinement, calibration, and consideration of outlier events is highlighted. The work contributes valuable insights into the application of LSTM, forecasting, and power consumption analysis in the context of individual households.

III. Dataset

A. Data Collection

For the project, the Individual Household Electric Power Consumption provided by the UCI Machine Learning Repository was chosen. The dataset has fields such as Date, Time, Global Active Power, Global Reactive Power, Voltage,

Global Intensity, sub_metering_1 which corresponds to the kitchen, sub_metering_2 which corresponds to the Laundry Room, and sub_metering_3 which corresponds to an electric-water heater and an air-conditioner.

The dataset comprises of 2075259 measurements gathered in a house located in Sceaux (7km of Paris, France) between December 2006 and November 2010 (47 months).

B. Data Exploration

The dataset contains 25979 values, which are either marked by '?' or NA values. Dropping of the rows with these values in the dataset was done to implement time series forecasting on the data. Further, to implement the models, split time series data into train and test sets. The first 328 days' worth of data was allocated to training data and the rest to testing data. Divided the training dataset in batches of 7 to restructure into weekly windows.

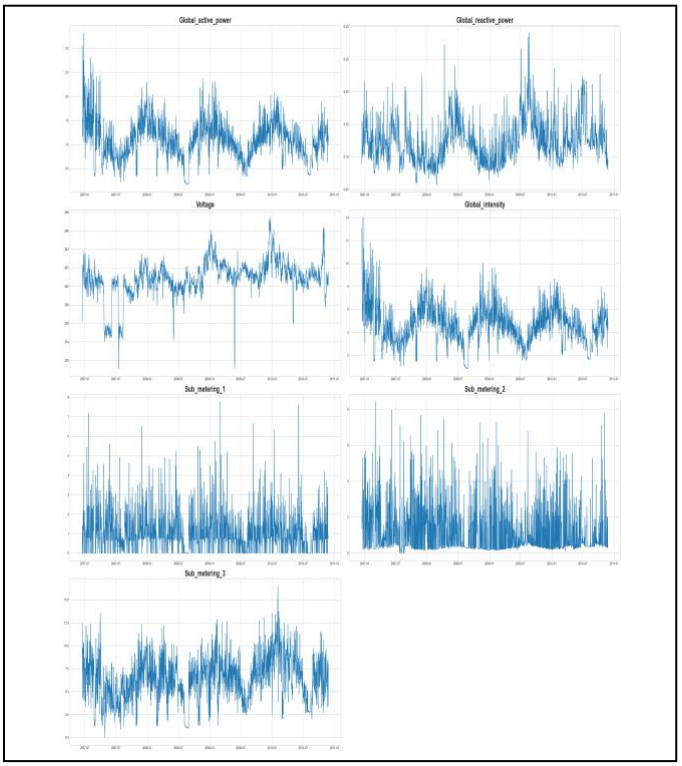

Fig. 1. Average consumption for each hour of the day

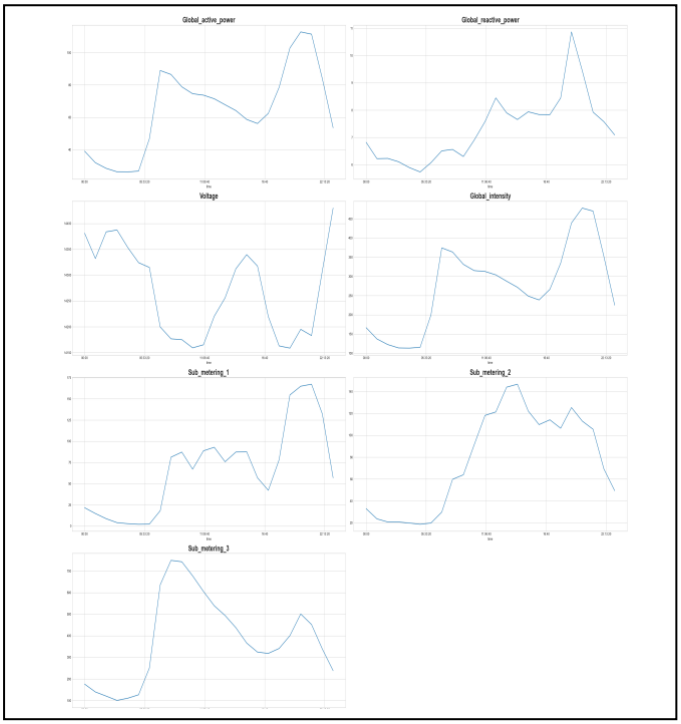

Fig. 2. Four years' worth of average daily consumption

IV. Model Design

Time Series Forecasting on data has been instrumental in giving the ability to estimate future events or trends. This emerged as an important factor for designing the models to predict power consumption for the week ahead based on the recent power consumption.

A. Data pre-processing

Found a high correlation between Global Active Power and Global intensity. Time series models are designed to recognize and learn from these dependencies in time series data. Hence, they are capable of handling high correlation in data, not requiring any further processing.

Created a new column called sub_metering_4 to calculate the power consumption of the entire circuit which does not include the power consumption by either sub_metering_1, sub_metering_2, or sub_metering_3.

$$sub\_metering = sub\_metering\_1 + sub\_metering\_2 + sub\_metering\_2$$

$$sub\_metering\_4 = \left(global\_active\_power * \frac{1000}{60}\right) - sub\_metering$$

The models were then trained using three different models: Encoder-Decoder LSTM, Conv-LSTM, and CNN-LSTM.

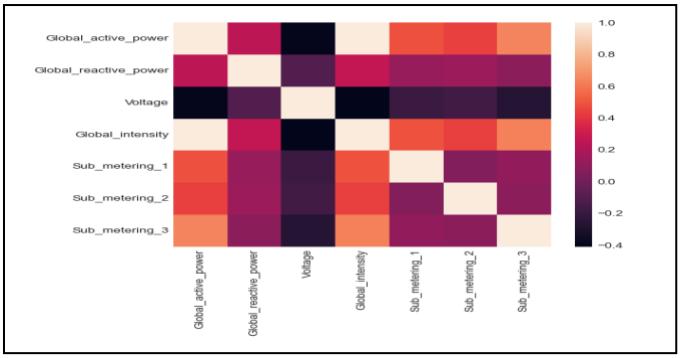

Fig. 3. Correlation Matrix of Electricity Consumption Data

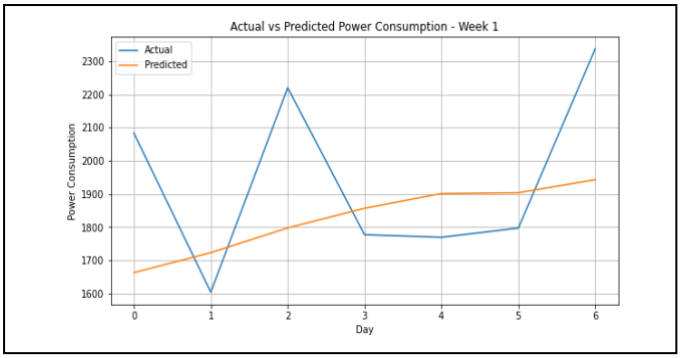

Fig. 4. Actual vs. Predicted Power Consumption – Week1

V. EXPERIMENTS AND RESULTS

A. *Encoder-Decoder LSTM*

Firstly, we have developed an LSTM Encoder-Decoder model to predict weekly power consumption using historical data. Division of the data was done into weeks for the purpose of splitting the data into train and test data, then leveraged TensorFlow to develop an LSTM model.

The model design involved four layers. First, to compute the input sequences, an LSTM layer with 200 units. This layer used the ReLU activation function. Next, for duplication of the feature vector for the output sequence, a RepeatVector layer is added. To return the sequences, the addition of an LSTM layer is the same as the previous, with 200 units, and a ReLU activation layer. Finally added, TimeDistributedDense layers with the ReLU activation function. The final layer will give a single output unit for prediction results.

Compilation of model was done using Adam optimizer and Mean Squared Error loss function. Training was done using a default epoch number of 50 and the batch size of 16. During, the training of the model, validation is also involved.

Sketched the model's performance spanning a week. It was done by analyzing the power consumption data. Assessing the real and expected usage of energy there was a clear exhibition of the accuracy of the model on certain days. The actual usage of the electricity curve on Days 0 and 4 is nearly similar to that of the predicted consumption curve, this was obsequious using the plot. Although, many disparities can be seen for Day 2 as well as for Day 6. It showcases that the model's accuracy is not stable as it varies. To make it consistent, extra calibration must be done.

Further analysis was done using the RMSE plot. It shows the LSTM's daily undertaking. Through, the pot it was clearly visible that at the start of the week, the prediction errors were lower. It also makes us understand using the visualization that it peaks on Friday. Although, from the variation we get to know that it is not performing as expected. From this, we can concur that there are underlying patterns in the data that the model is unable to capture.

To analyze the learning dynamics of the model, the training versus validation loss map across 50 epochs was studied. From that plot, it was seen that training data was getting plateaued. Concurring with this fact there might be rapid acquisition of patterns from training data. Also, a hushed fall after the plateauing is seen because of the validity loss. This difference suggests that there may have been overfitting of the training set. This overfitting will limit its generalization capacity.

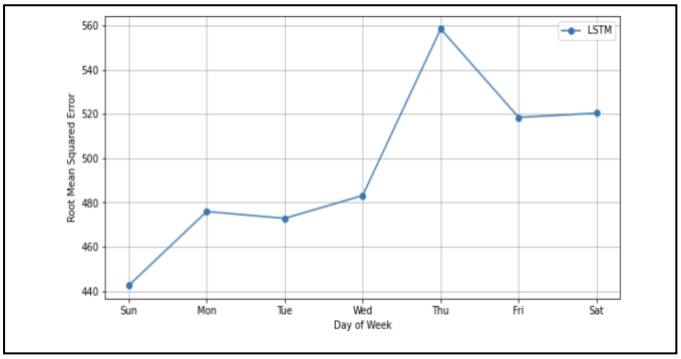

Fig. 5. Root Mean Squared Error by Day

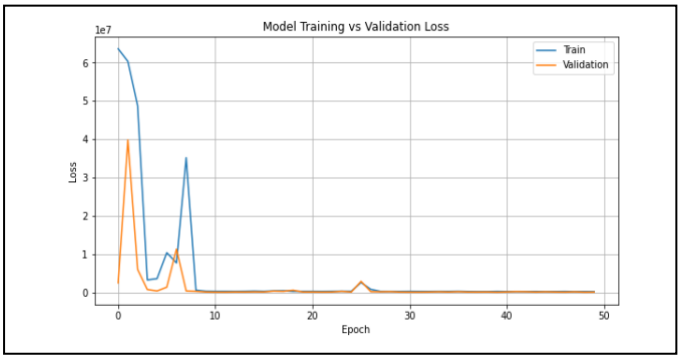

Fig. 6. Model Training vs Validation Loss

B. *CNN-LSTM*

Initially, univariate datasets were into training and testing sets. These two sets that is training set and the testing set were made discrete sets. These were split every week.

A function called an evaluate_forecasts function was implemented to calculate RMSE for each day. This was done so that every day's prediction performance could be assessed on a molecular basis. Along with this, implemented aggregate RMSE score too for predictive accuracy for the entire testing period.

Reconfigured data into a series of inputs and outputs. It was harmonious with the model's multi-step prediction paradigm. The model was built using a proper amalgamation of the CNN or Convolutional Neural Networks with LSTM or Long Short-Term Memory layers. These two form the encoder-decoder structure of our model. This was done so that CNN's feature extraction capability and LSTM's sequence handling prowess could be used.

Adam optimizer was used for efficient convergence during training. This is very helpful when the data is high-

dimensional. The training duration was chosen to be 20 epochs, which will help in balancing the tradeoff between overfitting and underfitting. This optimizer also considers other extensions i.e., AdaGrad and RMSProp.

Firstly, two layers of Conv1D layers are put in. These layers have a filter where each filter has a kernel size of 3. In total, there are 64 filters. Leveraging the ReLU function, these two layers are successfully able to extract features from the input time series data. After these layers to reduce the spatial dimension of output, a MaxPooling1D layer is used. It has a pool size of 2. This will help in summarizing the layers. After this for sequential processing, a RepeatVector is used. This is done using the LSTM layer which has 200 units. At last, TimeDistributedDense layers will ReLU activation function, and 100 units will be used. This is done while the final output layer is added to give one single output. This is implemented per time step.

Fig. 6. depicts the comparison between actual vs predicted values over one week. There are obsequious deviations in the plot. Although the predictions can apprehend the trends, underestimation of peaks is done. There is also an overestimation of troughs is done. From this, we get to know that the model can be improved by more refinement. On the final day, there is a major disparity between actual and predicted values. It showcases how a potential aspect is not well captured.

From Fig. 7. it is evident that there is a cyclic pattern. The variability with the mean values advancing towards the weekend suggests weekly seasonality. On weekends it can be seen the standard deviation is larger. Also from the linear trends, we see it is fairly flat with a major increase only on the final day. This will require further analysis of causal factors.

The last graph i.e., the Fig. 8. showcases training vs validation loss. From the plot, it is seen that at the start the training loss decreases rapidly but as epochs increase, it stabilizes, we can concur with effective generalization by seeing that validation loss is decreasing at a more gradual rate. In the end, the close nearness of the two curves shows that the model has achieved a balance between learning and generalization.

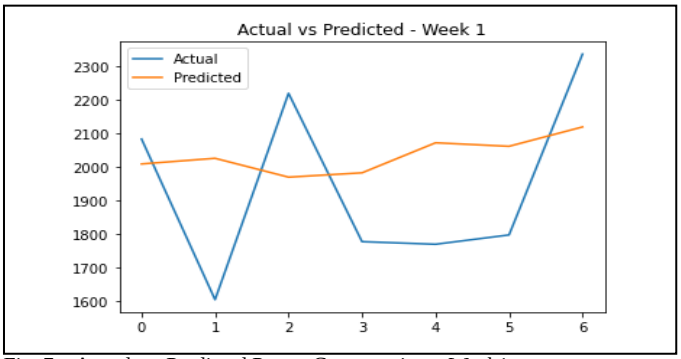

Fig. 7. Actual vs. Predicted Power Consumption – Week1

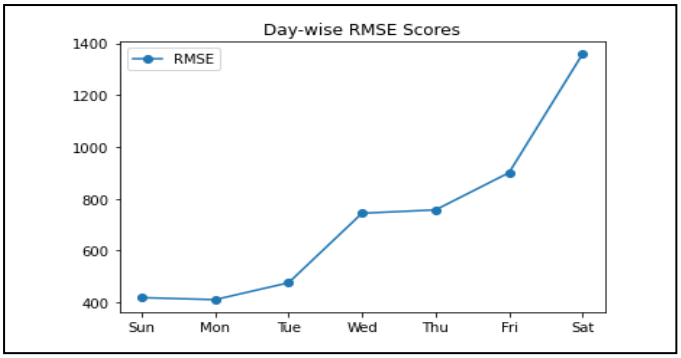

Fig. 8. Root Mean Squared Error by Day

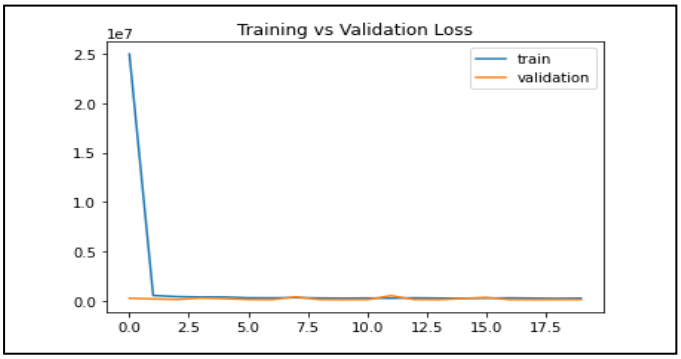

Fig. 9. Model Training vs Validation Loss

C. *Conv-LSTM*

Used Adam optimizer as it is highly useful when working on time series data. It also helps in rapid convergence. It plays a crucial role in the Conv-LSTM model by including complexities of the data.

The training duration was chosen to be 20 epochs, which will help in balancing the tradeoff between overfitting and underfitting. The batch size of 16 is chosen. 16 is chosen as batch size to optimize memory utilization and processing speed. These are quite mandatory and useful when developing a high-performing model.

Tailored the model to handle time series forecasting properly. First, reshaped input data into subsequences. ConvLSTM2D layer is used in designing the model. It involves filters each having a kernel size of (1,3). In total, there are 64 filters. The layers leverage the ReLU activation layer. It helps in seizing spatial and temporal relationships.

Then, a Flatten layer is used. This helps to transform the output. After this layer comes the RepeatVector layer which helps the input to become the same length as that of output sequence. This is done in LSTM layers.

The LSTM layer is used to make predictions at each step. It involves 200 units and leveraged ReLU activation function too. For the output TimeDistributedDense layers are used. These TimeDistributedDense layers have 100 units. They too use ReLU as an activation function. The final layer is a TimeDistributedDense layer. It only has a single unit to output the prediction for each step.

Fig. 10. shows the comparison of actual vs predicted over one week. It is clearly seen that the actual line fluctuates showing the variability in observed data. While the predicted line is smoother. This shows that the model's predictions were unable to capture high variability. This disparity suggests that

the model may not be able to capture underlying patterns. when developing a high-performing model.

RMSE can be seen in Fig. 11. On Friday there is a significant peak, showcasing that there might be a higher error in predictions for that particular day. This suggests that the model is not taking into consideration the outlier event that occurred on Friday.

Training vs Validation Loss is shown in Fig. 12. The steepness and then flattening of training loss showcases that the model learns quickly helping in fitting the training data. There might be overfitting as the validation loss is relatively flat.

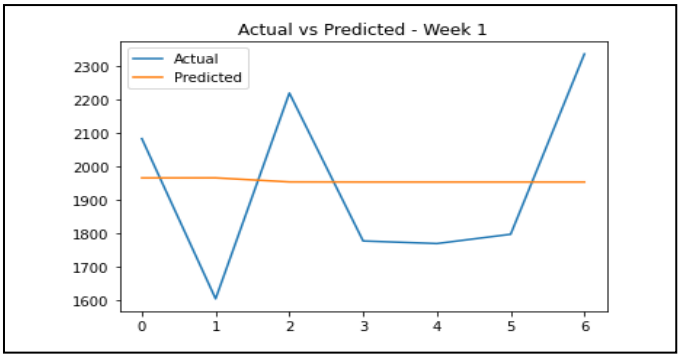

Fig. 10. Actual vs. Predicted Power Consumption – Week1

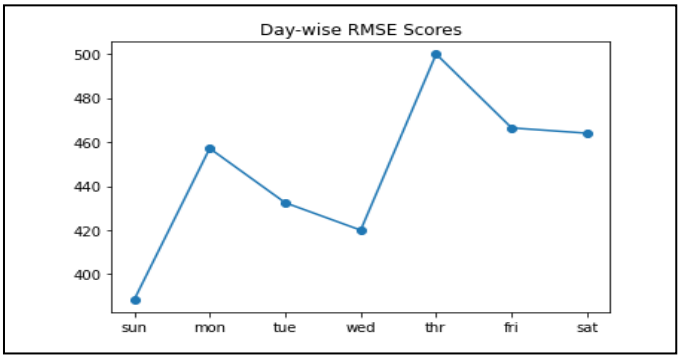

Fig. 11. Root Mean Squared Error by Day

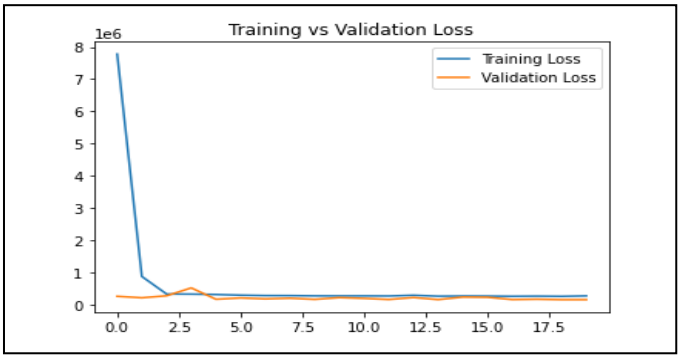

Fig. 12. Model Training vs Validation Loss

CONCLUSION

Through Table I. LSTM model might be unable to capture the complexities in the data as seen using MSE and RMSE. Also, from MAE we can say here are large deviations in the predictions. But from MAPE and SMAPE we can say that model is favorable if implemented on less complex data.

From the Table I. it is also visible that CNN-LSTM model performs better than the LSTM model. As it has lower MSE and RMSE scores as compared to the LSTM model it is better able to capture the complexities in the data. Although, a point to be noted as there is no consistency in this data. Hence, confirming that there might be a better model able to work superiorly on these complex data.

From Table I. we can say that Conv-LSTM model performs the best. When it is compared to the other models i.e. LSTM and CNN-LSTM. This is very visible by seeing the MSE score of the Conv-LSTM model. It has low scores throughout all the days of the week. This tells is that there is less deviation. Also, from low RMSE score we can say that standard deviation on prediction errors is also very low. MAE too shows that predictions are like actual data as it is low too. This fact is strengthened buy thew fact that MAPE as well as SMAPE are also low.

In conclusion, outperforms the other models. We can say that the Conv-LSTM model is superior with the consistency in its metrics. It is not only consistent on metrics but also on different days making it the preferable choice. This consistency shows that it is able to capture the complexities of the data ion a better manner than the other two model. Complexities for both linear as well as non-linear patterns in time series data. This implementation of the model can lead to better electricity consumption management leading to reduced electricity costs.

TABLE I. EVALUATION METRICS

MODEL	METRIC	DAY						
		SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
LSTM	MSE	195715.7	226479.8	223556.3	233426.3	311899.8	268800.4	270824.6
	RMSE	442.4	475.9	472.8	483.1	558.5	518.5	520.4
	MAE	342.8	369.0	380.3	387.2	465.8	439.9	429.4
	MAPE	30.1	33.2	35.5	34.8	43.9	39.4	38.4
	SMAPE	23.2	25.5	26.3	26.0	31.2	29.0	27.9
CNN-LSTM	MSE	175089.21	168637.54	226433.37	554078.58	573283.41	811377.95	1846376.61
	RMSE	418.44	410.66	475.85	744.36	757.15	900.77	1358.81
	MAE	321.26	314.48	411.84	634.87	674.22	844.61	1278.74
	MAPE	22.85	21.17	26.78	37.10	43.92	56.63	77.96
	SMAPE	21.92	22.41	30.04	47.97	57.62	81.59	127.72
CONV-LSTM	MSE	150894.78	208959.55	187012.27	176399.61	249926.99	217535.48	215302.57
	RMSE	388.45	457.12	432.45	420.00	499.93	466.41	464.01
	MAE	291.37	373.47	332.02	338.66	397.30	394.92	378.42
	MAPE	23.91	31.91	30.72	29.34	37.21	35.63	33.30
	SMAPE	19.77	25.43	23.24	23.19	27.29	26.79	25.45

WORK DISTRIBUTION

Soham Kamble: CNN-LSTM Model with Multivariate Input, Literature Survey, EDA, Introduction, Data Preparation.

Pranit Kotkar: Conv-LSTM Model with Multivariate Input, Literature Survey, Introduction, Evaluation Metrics, Conclusion.

Bhavya Chawla: LSTM-Encoder Decoder Model with Multivariate Input, Introduction, Literature Survey.

REFERENCES

[1] Hebrail,Georges and Berard,Alice. (2012). Individual household electric power consumption. UCI Machine Learning Repository. https://doi.org/10.24432/C58K54.

[2] Gers, F.A., Eck, D., Schmidhuber, J. (2001). Applying LSTM to Time Series Predictable through Time-Window Approaches. In: Dorffner, G., Bischof, H., Hornik, K. (eds) Artificial Neural Networks — ICANN 2001. ICANN 2001. Lecture Notes in Computer Science, vol 2130. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44668-0_93.

[3] Hebrail,Georges and Berard,Alice. (2012). Individual household electric power consumption. UCI Machine Learning Repository. https://doi.org/10.24432/C58K54.

[4] Shi et. al., Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, arXiv:1506.04214.

[5] Sutskever et. al., Sequence to Sequence Learning with Neural Networks, arXiv:1409.3215

[6] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735.

[7] Kingma et. al., Adam: A Method for Stochastic Optimization, arXiv:1412.6980.

[8] Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASS)