Sequence Prediction of Drone Power Consumption with Hybrid TCN-LSTM Architectures

Author 1 **Author 2** **Author 3**
University of Michigan

Abstract

This study addresses the complexities of multivariate time series (MTS) analysis and forecasting in real-world applications, particularly focusing on power consumption and energy range predictions for package delivery Uncrewed Aerial Vehicle (UAVs). Traditional methods often fall short in capturing intricate dependencies across time steps and among multiple series. To overcome these limitations, we introduce a novel sequence-to-sequence model that synergizes a Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM). Our TCN-LSTM model uniquely combines a TCN-based encoder to effectively grasp temporal dependencies within the input data, with an LSTM-based decoder for generating future sequences from encoded features. This architecture is further enhanced by integrating future covariates at each forecasted time step emphasizing future conditions, thereby optimizing prediction accuracy for future energy consumption. Unlike most existing studies focused on single-step predictions, our model excels in multi-step forecasting, as demonstrated through extensive experiments on real-world MTS data. The results show that our TCN-LSTM model outperforms previous DeepTCN and LSTM-LSTM models, marking a significant advancement in sequence-to-sequence prediction accuracy for complex MTS challenges.

The codebase is available at GitHub Code

1 Introduction

1.1 Motivation

The surge in electric mobility, characterized by the growing prevalence of electric vehicles (EVs) and drones, brings forth the challenge of accurately forecasting power consumption and operational range. Traditional models often fall short in capturing the dynamic interplay of factors affecting power usage, leading to uncertainties in range predictions and operational planning. To tackle this, our project embarks on an exploratory journey, starting with the development of baseline LSTM-LSTM models (with and without covariates) and progressing to more sophisticated architectures. We delve into the realms of DeepTCN and TCN-LSTM models, examining their efficacy both with and without the incorporation of covariates. This stepwise approach allows us to not only benchmark but also uncover insights into the strengths and limitations of each model in the context of multivariate time series forecasting.

Our research transcends technical boundaries, impacting the user experience and operational efficiencies in the electric mobility sector. Improved forecasting models can significantly alleviate range anxiety for EV and drone users, and offer industries critical data for resource optimization and strategic decision-making. By grounding our work in real-world drone datasets, we not only validate these models in realistic scenarios but also pave the way for their application across diverse domains. This progression of models from simple to complex, and the integration of real-world data, demonstrates our commitment to advancing the field of time series forecasting and supporting the global shift towards sustainable transportation solutions.

1.2 Problem Statement and Comparative Analysis

The domain of multivariate time series (MTS) forecasting, especially in the context of electric mobility, poses a distinct challenge: predicting power consumption and operational range based on historical multivariate observations. This task demands a model capable of understanding complex dependencies both within and across these time series. Mathematically, this involves modeling a function f such that given a historical series of observations X (comprising multiple features over time), the model can accurately predict a future series of observations Y.

Let $X = [x_1, x_2, \ldots, x_n]$ denote the input sequence, with each x_i being a vector of features at time step i. The objective is to predict $Y = [y_{n+1}, y_{n+2}, \ldots, y_{n+m}]$, where m is the prediction horizon. This complexity is heightened by the necessity to include covariates – external variables that potentially influence predictions but are not part of the main time series.

Formally, we define the problem as:

$$Y = f(X, C) \tag{1}$$

where C represents a set of covariates. The crux lies in crafting f to effectively capture both temporal and cross-sectional dependencies within X, and the influence of C on future values in Y.

In the context of electric mobility, this equates to precisely forecasting the power consumption of electric vehicles (EVs) and drones under diverse operational conditions. The model must consider variables like payload, altitude, speed, and environmental conditions, represented within X and C.

Our approach to this problem is comprehensive. We commence with Long Short-Term Memory (LSTM) models as our baseline, acknowledging their prowess in sequence learning for time-dependent data. Following this, we introduce the architecture of DeepTCN, an advanced version of Temporal Convolutional Networks (TCN), which integrates TCNs with residual blocks, therefore being able to take future covariates into model predictions. Subsequently, we delve into the incorporation of TCN for their aptitude in capturing longer dependencies and scalability benefits. In comparison to other sequential models such as standard RNNs and GRUs, our chosen models (LSTM, DeepTCN, TCN-LSTM) offer distinct advantages in handling long-term dependencies and high-dimensional data, essential in the context of electric mobility. LSTM's ability to remember information over long periods makes it ideal as a baseline, while TCN's unique architecture provides superior scalability and parallel computation capabilities. DeepTCN further enhances it by integrating TCNs with deeper networks and covariates, making it capable of handling more complex and longer sequences.

2 Related Work

LSTM networks are foundational in time series forecasting, with Hochreiter and Schmidhuber's pioneering work establishing their capacity for learning long-term dependencies [4]. Building on this, sequence-to-sequence models were introduced by Sutskever et al. for complex learning tasks, such as NLP, and have since been adapted for time series forecasting [8]. The evolution of these methodologies saw the successful application of deep convolutional neural networks, including TCNs, which have been shown by Bai et al. to outperform recurrent models in sequence modeling tasks [1][9]. Our research harnesses TCNs alongside LSTMs in a novel sequence-to-sequence architecture, focusing on multivariate time series (MTS) forecasting for electric mobility, specifically targeting the prediction of range and energy consumption.

Our novel TCN-LSTM architecture is inspired by Lai et al.'s integration of convolutional and recurrent layers in LSTNet [6] and by recent advancements in EV range prediction by D. Kim et al. [5]. We differentiate our work by applying this integrated model to drone power consumption datasets, as explored by Rodrigues et al. [3], and by extending prediction horizons and enhancing forecast accuracy. This approach uniquely positions our project to provide more precise and actionable predictions for electric mobility, leveraging the strengths of both TCN and LSTM models.

3 Method

Multivariate time series (MTS) forecasting necessitates models capable of grasping both the intricate temporal dependencies and potential influence of external factors. Our study embarked on a journey beginning with the LSTM model, revered for its proficiency in capturing long-term dependencies in sequential data, which served as the baseline for our investigation. The LSTM's ability to remember and utilize past information over extended sequences made it an ideal starting point.

However, despite the LSTM's capabilities, we recognized the potential for enhancing prediction accuracy by considering additional temporal patterns and external covariates. This led us to explore DeepTCN, a deeper version of the Temporal Convolutional Network that incorporates future covariates. DeepTCN's architecture is adept at handling longer sequences and provides a mechanism to include future covariates, albeit in a non-autoregressive fashion. The incorporation of covariates in DeepTCN, though significant, treats each covariate independently rather than sequentially, which could limit the model's foresight in certain predictive scenarios.

To address this, we advanced towards a hybrid approach of TCN-LSTM. By employing TCN as the encoder and LSTM as the decoder, we aim to amalgamate TCN's strength in capturing complex spatial-temporal patterns with LSTM's prowess in autoregressive forecasting. This integration allows our model to consider the historical data processed through the TCN and sequentially generate future predictions with LSTM, accounting for the influence of covariates at each step. Such an approach is expected to yield a more nuanced and accurate forecasting model, adept at predicting longer sequences and generating superior results.

This section of the paper will outline the fundamental aspects of LSTM-LSTM architecture, delve into the advanced DeepTCN structure, and culminate with our proposed TCN-LSTM model. Each of these models contributes uniquely to our understanding of MTS forecasting, leading us to a more refined approach that harnesses the combined strengths of TCN and LSTM.

3.1 LSTM-LSTM

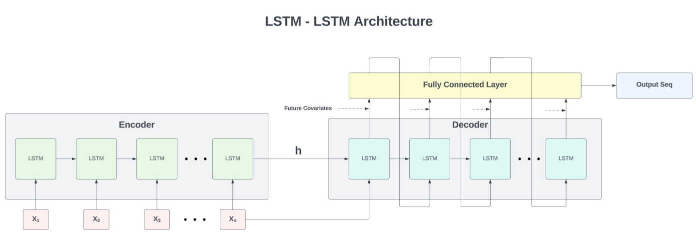


Figure 1: LSTM-LSTM model architecture.

The LSTM-LSTM shown in figure 1 model encompasses two primary components, Encoder and Decoder. The encoder is an LSTM layer tasked with processing the input sequence and capturing temporal information. It transforms the input sequence into a set of hidden states, termed h and c, that encapsulate the historical data's features and are regarded as short-term and long-term memories, respectively. The decoder is an LSTM layer that takes the final hidden state from the encoder to generate the output sequence. It predicts the next value in the sequence in an autoregressive manner, utilizing each step's output as input for the subsequent step. The decoder's output is typically processed through a fully connected layer to map the LSTM features to the desired output size.

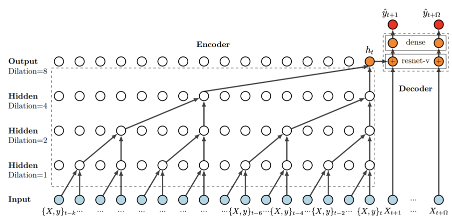
3.1.1 LSTM-LSTM Without Covariates

This model variant operates solely on a sequence of historical data points to forecast future values, excluding any external or additional information (covariates). The model leverages the last input sequence time step to initiate the decoding process. It is suited for scenarios where the prediction is predicated exclusively on the target sequence's historical values.

3.1.2 LSTM-LSTM With Covariates

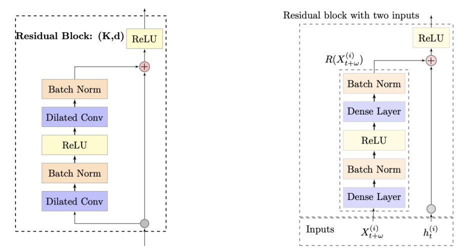
This version integrates external information, termed covariates, potentially affecting the predictions. Covariates can span a range of factors, from environmental conditions to economic indicators, or any pertinent data anticipated to influence future values of the target sequence. Here, while the encoder processes the historical sequence, the decoder can utilize future covariates to construct the output sequence. Each decoding step involves merging the corresponding future time step's covariate with the encoder's hidden state to forge a prediction. This model is advantageous when the future sequence's outcome is dictated not only by its historical values but also by other influencing factors.

3.2 Transition to DeepTCN



(a) Architecture of **DeepTCN**

Figure 2: Architecture of DeepTCN



(b) Encoder module

(c) Decoder module

Figure 3: Encoder and Decoder of DeepTCN

Deep Temporal Convolutional Networks (DeepTCN) [2] represent a novel architecture designed for sequence modeling, particularly effective for multivariate time series forecasting. The DeepTCN structure is an advanced variant of Temporal Convolutional Networks (TCN) that leverages deep learning principles to enhance performance, especially for long-term predictions. As shown in Figure 2 and 3, DeepTCN comprises an encoder-decoder architecture. For encoder part, stacked dilated causal convolutional nets are constructed to capture the long-term temporal dependencies. For decoder part, it included a variant of residual block and an output dense layer. The module resnet-v is designed to integrate output of stochastic process of historical observations and future covariates. Then the output dense layer is adopted to map the output of resnet-v into final forecasts.

3.2.1 Encoder and Decoder

The encoder, as depicted in Figure 3(b), is constructed with a series of residual blocks, each containing two layers of dilated causal convolutions. These convolutions are purposefully designed to ensure that the output at time t is only dependent on inputs from time t and earlier, thereby preserving the causal nature of the forecast. The first convolution layer in each residual block is followed by batch normalization and a Rectified Linear Unit (ReLU) activation function to introduce non-linearity. The second convolution layer is followed by another batch normalization. The output of the second batch normalization is then added to the block's input, followed by an additional ReLU activation function. This design allows the network to learn an identity function, ensuring that the higher layers of the network can perform as well as the lower layers without losing information.

The decoder of DeepTCN, shown in Figure 3(c), synthesizes the encoder's outputs, denoted as $h_t^{(i)}$, with future covariates $X_{t+w}^{(i)}$ through a specialized module named resnet-v. This component employs a sequence of dense layers and batch normalizations, interjected with ReLU activations, to form a nonlinear function $R(.)$ that assimilates the future covariates into the predictive process. The decoder's architecture allows it to incorporate imminent external influences, making it particularly adept for forecasting applications sensitive to environmental changes, like power consumption.

3.3 Integration of Historical Features and Future Covariates

DeepTCN's design facilitates the seamless integration of historical data with future covariates. In the forward pass, the model's encoder captures temporal dependencies within the historical data, and global average pooling distills this into a comprehensive representation. At the decoder stage, the model combines this representation with future covariates at each forecasted time step. By doing so, the model conjures predictions that reflect both past patterns and future conditions, resulting in a robust multi-step forecast. This dual integration accentuates the model's proficiency in delivering detailed forecasts, leveraging an extensive dataset's full potential.

3.4 TCN-LSTM: A Hybrid Approach

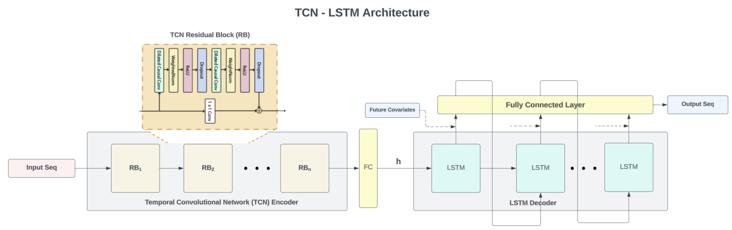


Figure 4: TCN-LSTM model architecture.

The TCN-LSTM model shown in figure 4 consists of two primary components: the TCN-based encoder and the LSTM-based decoder. The encoder is responsible for capturing the spatiotemporal features and long-term dependencies in the input data, while the decoder focuses on generating the future sequence based on the encoded features.

The encoder employs the TCN, as described in the previous section. The TCN encoder's output is then passed through a fully connected (FC) layer in temporal and feature dimensions, in order to map the features to a size compatible with the LSTM's hidden states. By doing that, the TCN encoder processes each input sequence x across its entire length, transforming it into a higher-level representation that captures both local and global dependencies. This feature-rich representation is then condensed into a final set of hidden states h and cell states c, which encapsulate the essential information required for the decoding phase. More specifically, h as short-term memory for the LSTM is obtained by passing the TCN outputs at the last time step through a FC layer, while c as long-term memory for LSTM is generated by passing entire sequence through FC in both temporal and feature dimension,

The LSTM-based decoder takes outputs from the encoder as its initial hidden and cell states, and uses the inputs from the current time step. It then proceeds to generate the output sequence in an autoregressive manner, using its previous outputs as inputs for subsequent steps. This process allows the LSTM to model the temporal aspect of the sequence based on previously learned patterns.

3.4.1 TCN-LSTM Without Covariates

The TCN-LSTM model without covariates focuses exclusively on the historical sequence data, without additional external inputs. This setup is predicated on the assumption that the past values of the sequence contain sufficient information to predict future outcomes. The model learns to map the sequence of historical observations $x_{1:n}$ to a predicted future sequence $y_{n+1:n+m}$, where n is the length of the input sequence and m is the prediction horizon.

In our application of drone power consumption prediction, the TCN-LSTM without covariates model aims to predict future power use based purely on historical flight data. This model is particularly useful in scenarios where external factors are either unavailable or deemed to have a negligible impact on the prediction task. The choice to initially exclude covariates is a deliberate simplification that allows us to establish a baseline performance, upon which we can later assess the incremental benefit of incorporating additional contextual information.

3.4.2 TCN-LSTM With Covariates

For a more comprehensive predictive model that incorporates external factors influencing the target variables, we introduce covariates into our TCN-LSTM architecture. In this version of the TCN-LSTM model, the covariates are integrated directly into the LSTM decoder. The Encoder module remains unchanged, leveraging the capabilities of TCN to capture complex patterns in the input sequence. The Decoder module, however, is adapted to include additional information from the covariates at each time step of the output sequence. This fusion of historical data and relevant external factors aims to yield more accurate and context-aware predictions.

The Decoder is therefore designed to handle two streams of input at each time step: the last output of the LSTM layer and the current covariate. After each LSTM operation, the resulting hidden state is concatenated with the current covariate, enriching the information before passing through a fully connected layer. This ensures that the prediction at each time step is made considering not just the historical sequence but also the present external influences. This will repeat for the entire length of the predicted sequence.

This model's ability to integrate covariates provides a comprehensive approach to MTS forecasting. It stands to significantly improve the operational efficiency of electric mobility solutions by offering detailed and contextually enriched power consumption forecasts, thereby ensuring better resource management and operational reliability.

4 Experiments

4.1 Datasets

The project centers on analyzing a comprehensive dataset from 209 drone flight campaigns, focusing on power consumption prediction. This dataset [7] encompasses 257,895 entries across 28 columns, featuring data such as flight number, time, wind parameters, battery readings, drone positioning, and movement metrics. The selected useful features we extracted for training are Payload, Wind Speed, Wind Angle, Power, Linear Acceleration x, Linear Acceleration y, and Linear Acceleration z. The raw dataset was first filtered (removal of NaN and empty entries from flights 211 to 219) and then added with derived features such as mean, maximum, and minimum altitudes per route, along with instantaneous power consumption by multiplying voltage and current readings.

A critical aspect of the data handling is the calculation of energy consumption. Mathematically, energy is the accumulation result of power by time. Therefore, instead of predicting energy consumption, we predict the power consumption instead for this task. The dataset further implies future covariate features: anticipated drone positions ($x_{future}, y_{future}, z_{future}$), based on known routes and speeds at dispatch. Additionally, incremental position changes ($x_{change}, y_{change}, z_{change}$) calculated for future time stamps are also included for covariates. To provide a preliminary justification for the effectiveness of covariates on optimizing the model performance: we explored the correlation between future covariates and the power consumption using ridge regression method. Linear ridge regression test fitted and performed on selected input features with training data demonstrated z_{change}, y_{change} have the largest coefficient values besides battery voltage and current. The ridge regression reached a MSE of 0.000116.

4.2 Models Evaluation

In the absence of covariates, the model evaluation focused on the ability of LSTM-LSTM and TCN-LSTM architectures to predict future power and energy consumption based solely on historical data. The LSTM-LSTM model, representing a traditional approach, utilizes a sequence-to-sequence framework without external factors. The TCN-LSTM model, on the other hand, combines the strengths of temporal convolutional and recurrent networks.

With the inclusion of covariates, our evaluation aimed to understand how additional contextual information could enhance the forecasting capabilities of our models. Here, we considered the LSTM-LSTM, DeepTCN, and TCN-LSTM architectures, now augmented with covariate data, to predict future power and energy consumption. The covariates provided supplementary information that was expected to refine the models' predictive power.

4.3 Results Analysis

The performance of the models was quantitatively assessed using the Mean Absolute Percentage Error (MAPE) for both power and energy predictions. The MAPE for flight i can be formulated as follows, where N_i denotes the number of time steps at flight i, $y_t^{(i)}$ is the ground truth at time t of

Model	Power MAPE(%)	Energy MAPE(%)
Naive	15.19035	0.28255
LSTM_LSTM	9.16999	0.11375
TCN_LSTM	8.87271	0.10853

Table 1: Model Evaluation Without Covariates

Model	Power MAPE(%)	Energy MAPE(%)
Naive	14.89410	0.28213
LSTM_LSTM	8.68546	0.11146
DeepTCN	9.78302	0.15051
TCN_LSTM	8.62113	0.10880

Table 2: Model Evaluation With Covariates

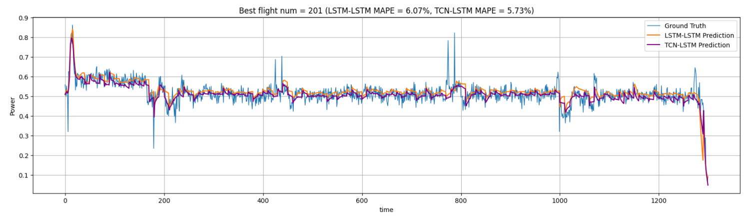


Figure 5: Best prediction by LSTM-LSTM and TCN-LSTM

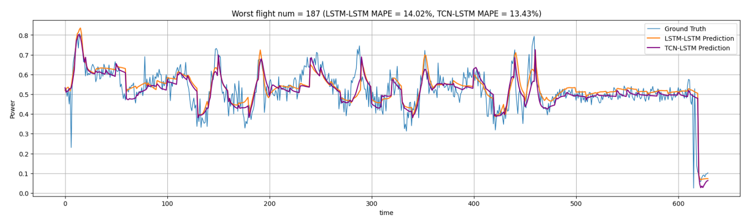


Figure 6: Worst prediction by LSTM-LSTM and TCN-LSTM

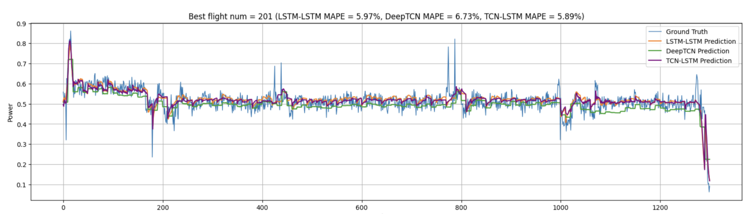


Figure 7: Best prediction with covariates by LSTM-LSTM, DeepTCN, and TCN-LSTM

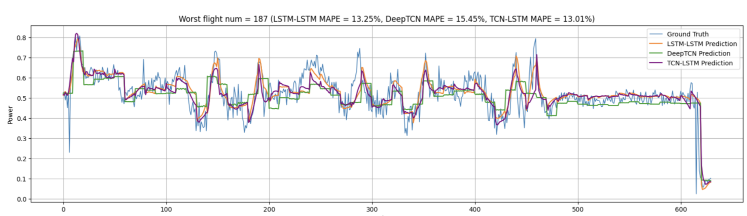


Figure 8: Worst prediction with covariates by LSTM-LSTM, DeepTCN, and TCN-LSTM

flight i, and $\hat{y}_t^{(i)}$ is the prediction at time t of flight i.

$$MAPE_i = \frac{1}{N_i}\sum_{t=0}^{N_i}\left|\frac{y_t^{(i)} - \hat{y}_t^{(i)}}{y_t^{(i)}}\right| \times 100 \tag{2}$$

We are interested in power prediction, and subsequently, with the power, one can compute the energy consumption by $\hat{E} = \sum \hat{P} \times \Delta t + E_0$. Thus in this case, the MAPE on power and energy predictions are computed. To compare, we also created a naive model that predicts the power by simply taking the average of powers as the predictions for each flight. Table 1 summarizes the results of models without covariates, and Table 2 includes the results of three models that consider covariates.

From Table 1, we can tell that the TCN-LSTM model outperformed the Naive and LSTM-LSTM models. Additionally, the number of parameters of TCN-LSTM and LSTM-LSTM are closely the same. This indicates the superior capability of TCN-LSTM in capturing the temporal dynamics of the dataset without additional covariate information. As for the non-covariate models, the results detailed in Table 2 demonstrate the impact of covariates in improving the power forecasting accuracy of both architectures, though LSTM-LSTM model showed a pronounced improvement, TCN-LSTM still outperforms all other models in this case. DeepTCN distinctly behaves as a non-autoregressive model, and the results show that its performance is worse than the autoregressive ones, demonstrating the importance of an autoregressive temporal module for time-series sequence predictions.

The figures 5 - 8 present time-series sequence prediction results of the best and worst flights for TCN-LSTM in both cases. Generally, the predictions by the autoregressive models can follow the peaks and troughs of power consumption, while DeepTCN yields roughly constant predictions for every sequence. That is reasonable as the prediction by DeepTCN at every time step depends on the encoder output, which is the same for every time step, and covariates input at each time step, respectively; once the covariates at each time step are closed, the decoder will pass similar inputs and output similar results at each time step, therefore generates a constant sequence. By comparing the results between TCN-LSTM and LSTM-LSTM, one can tell that the sequence predictions by TCN-LSTM are more temporally dependent in some edge cases and hence produce more accurate results robustly. That can be attributed to the outstanding ability of TCN to capture temporal dynamics based on historical features. By incorporating the future covariates, both autoregressive models show a better capability to follow the extreme temporal change, which can credited to the relationship between the future trajectories and power consumption.

5 Conclusion

In this project, our primary focus has been on the sequence-to-sequence model for MTS forecasting. Our approach involved designing models for MTS, considering both models with and without future covariates. In addition to traditional models like LSTM-LSTM, we extended the model by incorporating TCN as the encoder in the sequence-to-sequence architecture. We emphasized the significance of employing an autoregressive model for accurate time-series sequence prediction. To further investigate this, we introduced DeepTCN, a model where predictions at each time step rely on historical features and future covariates independently.

Through the application of these designed models to drone data, we showcased the superior performance of TCN-LSTM compared to other models in terms of long sequence prediction accuracy and capabilities of capturing time dependencies. Our results also highlighted the potential improvement in prediction performance with the inclusion of future covariates which are spatial features in this case. The outcomes of DeepTCN, when compared to all other models, demonstrated our argument regarding the crucial role of utilizing an autoregressive temporal module, such as a recurrent neural network, in enhancing sequence forecasting.

A Contribution By Group Members

- Author 1
  - Designed and built LSTM-LSTM models with and without covariates
  - Designed and built TCN-LSTM models with and without covariates
  - Training and testing, parameter tuning for both models in both cases to improve performance
  - Summarize all prediction models, generate and analyze the results, evaluate their performances
- Author 2
  - Implemented dataloaders and related support functions such as parametrizable sequence generations, consumptions/future covariates calculations.
  - Analyzed data quality, feature correlations. Explored references and imported new data features to further improve models
  - Performed various regression model analysis to demonstrate model feature soundness
- Author 3
  - Crafted the Residual Blocks, Encoder, Decoder components, and the overarching DeepTCN class.
  - Solved features dimension mismatch issue during training of the model by fixing the forward pass function of DeepTCN class.
  - Trained and evaluated the DeepTCN model on agreed metrics and optimized the model applying hyper parameter searching, such as number of residual blocks, and number of channels through Optuna Framework.
  - Wrote paper sections of Introduction, Related Work, and Implementations.

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

[2] Yitian Chen, Yanfei Kang, Yixiong Chen, and Zizhuo Wang. Probabilistic forecasting with temporal convolutional neural network, 2020.

[3] Arnav Choudhry, Brady Moon, Jay Patrikar, Constantine Samaras, and Sebastian Scherer. Cvar-based flight energy risk assessment for multirotor uavs using a deep energy model. In *Proceedings of (ICRA) International Conference on Robotics and Automation*. IEEE/RAJ, May 2021.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[5] Dohee Kim, Hong Gi Shim, and Jeong Soo Eo. A machine learning method for ev range prediction with updates on route information and traffic conditions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12545–12551, Jun. 2022.

[6] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks, 2018.

[7] Thiago A. Rodrigues, Jay Patrikar, Arnav Choudhry, Jacob Feldgoise, Vaibhav Arcot, Aradhana Gahlaut, Sophia Lau, Brady Moon, Bastian Wagner, H Scott Matthews, Sebastian Scherer, and Constantine Samaras. Data Collected with Package Delivery Quadcopter Drone. 2 2021.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[9] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.