Comparative Evaluation of LSTM, Transformer, and Hybrid Architectures for Multivariate Time Series Forecasting on Power, Traffic, and Weather Domains

Hrishikesh More
Machine Learning MSCAIJAN25I
National College of Ireland
Dublin, Ireland
x23311576@student.ncirl.ie

***Abstract*—This paper measures the results of three deep learning models, such as Long Short-Term Memory (LSTM [1]), Transformer [2], and Hybrid [3] LSTM [1]-Transformer [2] on multivariate time series forecasting in three different fields, including electricity load (Power), the number of traffic (Traffic), and weather measurements (Weather). All datasets were preprocessed in the same way, i.e. treatment of missing values, feature scaling and split into train-validation-test sets. The evaluation of the models was performed in terms of the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Symmetric MAPE (SMAPE) values as well as the inference latency. The experimental findings attained the peak in accuracy on all datasets by the Hybrid [3] model, whereas the Transformer [2] provided a competitive accuracy much faster in inferencing time. Domain-specific analyses of error showed that Power has spikes in demand and Traffic has high short-term volatility. These results point to the trade-offs between accuracy and computational complexity when it comes to effective application of forecasting to practice.**

***Index Terms*—Time Series Forecasting, LSTM [1], Transformer [2], Hybrid [3] Models, Multivariate Data, Deep Learning**

I. Introduction

The task of time series forecasting has found important application in a wide range of domains, such as electricity demand forecasting, traffic forecasting in cities and climate forecasting. Precise predictions will help the stakeholders to maximize their resources, minimize expenditure, and make better strategies. Multivariate time series data tend, though, to be considerably variable, long-term dependent, and characterized by complicated relationships between features, in which case forecasting becomes problematic to the traditional statistical models, be them ARIMA or exponential smoothing.

Time series forecasting with the deep learning structures became prominent in recent years. It has been demonstrated that the Long Short-Term Memory (LSTM [1]) networks possess powerful modeling sequential dependencies, and Transformers, which were originally designed to work with textual data, outperform in capturing long-range correlations via self-attention mechanism. The approach called hybrid architecture strives to provide the combination of LSTM [1] and Transformer [2] layers to make the best out of both paradigms.

The paper contrasts LSTM [1], Transformer [2] and Hybrid [3] LSTMTrasformer architectures at three benchmarking datasets namely UCI [4] ElectricityLoadDiagrams (Power), Monash [5] Traffic, and Jena Climate [6] (Weather). We evaluate the models in terms of accuracy and latency as well as provide an error analysis with a focus on identifying the forecasting challenges that are specific to the data.

II. Related Work

A. LSTM in Time Series Forecasting

Hochreiter and Schmidhuber proposed Long Short-Term Memory (LSTM [1]) in 1997 2 networks as an improvement to the then conventional Recurrent Neural Networks (RNNs), which had the deficiency of vanishing and exploding gradients. LSTMs use a memory gate, or gating, mechanism through which input, forget, and output gates allow key information to survive in long sequences and forget insignificant information. The use of LSTMs is especially suitable in the application of time series forecasting tasks in which long-term dependencies are relevant. LSTMs have been used as far back as two decades ago in areas including energy demand based on forecasts, stock price prediction and climate modeling. To illustrate, Kong et al. 4 employed an LSTM [1]-based model to accomplish a short-term load forecasting analysis at the residential scale and the performance of this model appeared to be higher than traditional feed-forward network models and autoregressive models.

LSTMs also show the capability to capture seasonality and multidimensional interdependence in a multivariate time series, in particular with additional layers such as dropout in regularizing the networks, or attention mechanisms to refine feature selection schemes. LSTM [1] models have been found to be effective in weather forecasting to predict temperature and precipitation event more accurately compared to the statistical models as they actively use all the input parameter, such as past weather data and external covariates 6. Nonetheless,

their strengths aside, LSTMs are computationally costly as they are sequential networks and thus it is less desirable in applications that demand low latency, real-time predictions. Moreover, LSTM [1] performance is known to exhibit poor performance on very long sequences in the absence of extra domains like hierarchical modeling or truncated back propagation through time.

B. Transformers for Time Series

Vaswani et al. proposed the Transformer [2] architecture 3 which replaced recurrence with self-attention mechanisms that made the model capable of computing all the relationships between time steps in parallel. The design exhibits a considerable computational benefit compared to LSTMs, since that design does not have a sequential processing bottleneck. Transformers have also been modified to enable forecasting of time series where both Temporal Fusion [7] Transformer [2] (TFT) 5 and Informer 7 7 have sought to address the long input sequences problem in a way that does not compromise accuracy.

Transformers are good at modeling long-range dependencies and this comes in handy when dealing with data that has time series type characteristics (periodic or seasonal) of long duration. Transformers are also able to remember the order with positional encodings, which is important to model time series. Lim et al. 5 showed TFT not only matched state-of-the-art accuracy across several forecasting benchmarks, but also provided interpretability via attention weights to allow conclusions to be drawn about the importance of features and trends over time.

Transformers however come with setbacks. The quadratic complexity of self-attention means that they can be very memory-intensive on very long sequences; and they can also need high volumes of training data in order to generalise effectively. The challenges of hybrid architectures and sparse attention mechanism are addressed with the view of making Transformers to be seen as increasingly feasible in terms of operational time series forecasting in the context of energy, traffic, and weather simplexes.

C. Hybrid Models

The purpose of hybrid models is to unite their strengths, which were observed in sequential models (LSTMs), and have global context modeling capabilities (Transformers). The reasoning is that LSTMs have the capacity to learn short-term temporal dependencies and local constraints whereas Transformers excel in a general capacity to learn global patterns across large time- periods. Wu et al. 6 examined deep hybrid structure of influenza prediction whose prevalence forecast was seen to be better and more robust with a mixture of RNN layers in Transformers encoder.

More advanced approaches such as hybrid models have been introduced in energy forecasting where it has been proposed to use LSTM [1] layers to preprocess the sequence-based data to extract short-line features, which will be fed into layers of Transformers to incorporate the overall relations of time. On a similar note, this method is also applicable in traffic forecasting where the unpredictability and dynamic changes in traffic will be addressed with consideration to seasonal trends.

The other benefit of the hybrid models is that they are adjustable to different granularities of data. As an example, they can also process the high frequency sensor readings (e.g. every minute) and still utilise the information in low-frequency patterns (e.g. daily record or weekly trends). These models however have the overhead of the combined architecture and the hyperparameters must be tuned carefully to give a balance in the contribution of each of the component.

D. Datasets in Literature

A number of benchmark datasets have become standard against which time series forecasting techniques are measured. UCI [4] ElectricityLoadDiagrams20112014 [4] It is dataset 7, which provides four years of data on electricity consumption by numerous clients, thus providing both a great variety of seasonal and weekly trends. The broad dimensionality and the multi-client design contributes to its strength as a test case of multivariate models.

Monash [5] Traffic dataset [5] 8 is the hourly values of road occupancy rates that were detected by the sensors at various places. Incidents, the weather and human behaviour render this dataset short-term volatile, making short term testing model resilience to irregular patterns ideal.

The Jena Climate [6] dataset 9 gives the measurements of different meteorological parameters like temperature, pressure, humidity, and wind speed of several years. It has a broad application in climate modeling studies and it is appropriate in assessing the performance of models to deal with correlated continuous characteristics.

By themselves, these datasets offer a heterogeneous test domain on which to compare models of forecasting over domains with varying temporal dynamics, noise levels, and interdependence of features. These benchmarks have been used to prove the increase in prediction accuracy and generalization of many previous studies such as 4, 5, and 6.

III. Methodology

The section provides the description of the data sources, preprocessing pipeline of each of them, and architectures of three deep learning models to be used in the study, namely LSTM [1], Transformer [2], and Hybrid [3] LSTM [1]-Transformer [2]. The research methodology was framed in such a way that it could be reproduced and fairly compared in different areas.

A. Datasets

To reflect a variety of forecasting fields, we chose three publicly available multivariate time series data sets that were publicly available:

Power (Electricity Load): Dataset The UCI [4] Electricity Load Diagrams 2011 - 2014, dataset 7 has electricity consumption information of 370 clients by kW at fifteen-minutely intervals between 2011 and 2014. The data is highly seasonal on a daily and weekly basis, interrupted with sharp spikes as a result of peculiar demand (e.g., holidays, extreme weather).

As in all of our other datasets, the data was aggregated in time to an hourly resolution, which reduces the amount of storage space size and model complexity without compromising the salient temporal dynamics involved.

Traffic: Monash [5] Traffic D8 The Monash [5] Traffic dataset [5] 8 includes the hourly concentration of traffic on the road based on various sensors along different segments of a highway. The occupancy is calculated as a proportion of time the above road segment was occupied by vehicles during the specified hour. The data is very dynamic hanging on various factors like traffic accidents, weather and big events. These attributes influence it to work well as a benchmark with regards to models that demand stability against abnormal short-term disturbances.

Weather: The Jena Climate [6] dataset 9 consists of meteorological readings of the Max Planck Institute for Biogeochemistry every 10 minutes. Variables can be temperature, atmospheric pressure, humidity, wind speed and density of air. It was resampled into comparable dataset at the level of an hour. The analysis of the capacity of model to deal with continuous interdependent variables is appropriate since it comes with robust seasonality as well as feature relationships.

B. Preprocessing

In order to make the pipeline consistent across datasets, the following preprocessing steps were implemented:

Numeric Conversion: All the columns were forced into numbers and the data was changed to NaN on non-numerics.

Miss value Treatment: Full forward Imputation (filling in missing values by propagating most recent valid value (forward replacement)) was followed by backward filling to cater to any initial gaps in the case of missing values.

Feature Selection: In all datasets, the target variable had been selected as the series with the greatest variance during the course of our training. Input features were chosen as the top remaining variables of the variance (7-10 or depending on data set)

Standardization: We normalised the features using the Z-score normalization where μ is the mean and σ is the standard deviation of the feature.

Splitting into Time: Data has been divvied up in time into 70, 10 and 20 on the training, validation and test sets respectively in order to avoid the problem of future information leaking into past predictions.

C. Model Architectures

We introduced three architectures to juxtapose the sequential and attention-based sequence modeling models as well as hybrid ones.

1) LSTM Architecture: LSTM [1] The LSTM [1] architecture used a two-layer LSTM [1] with 64 hidden units per layer, and a dense (fully connected) layer on top of the final hidden state to make the prediction. Between layers, dropout regularization (0.1) was put in order to eliminate overfitting. LSTM [1] Traditional LSTMs are good at modeling short- and medium-term dependencies but do so sequentially which can raise latency.

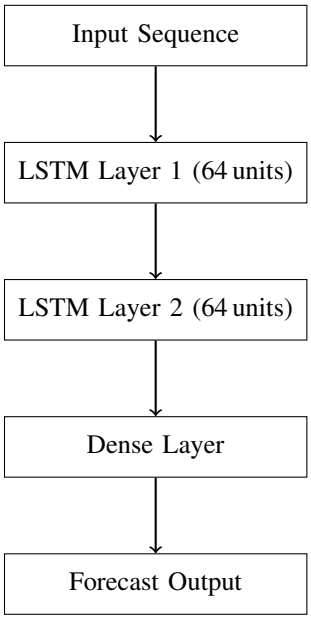

Fig. 1. LSTM Architecture

2) Transformer Architecture: Transformer [2] model utilized in this research paper had two encoder layers, and each layer had embeddings of 128 dimension and 4 attention head. Positional encoding was added to the input sequence in order to fix the temporal order. Self-attention Multi-head self-attention has permitted the model to attend relationships throughout the whole sequence at once, and helped it to infer faster than LSTMs.

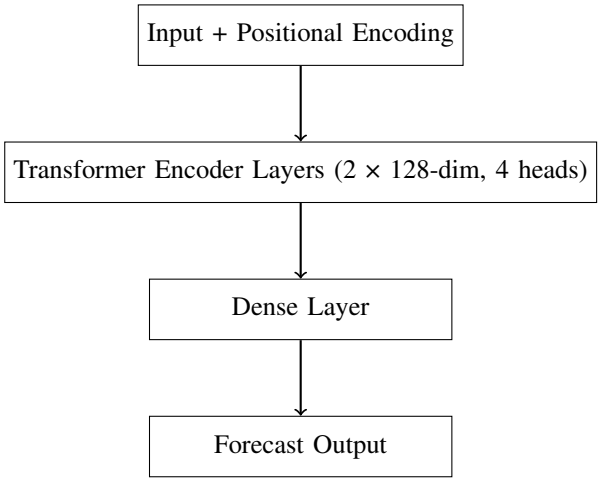

Fig. 2. Transformer Architecture

3) Hybrid Architecture: The Hybrid [3] used a first LSTM [1] layer (64 units) to describe local sequential depencies and Transformer [2] encoder layers (2x 128-dim, 4 heads) to describe long-range depencies. This hybrid is focused on the combination of the capabilities of both architectures: the LSTM [1] that is employed to process finer-grain short term patterns and Transformers to model global context.

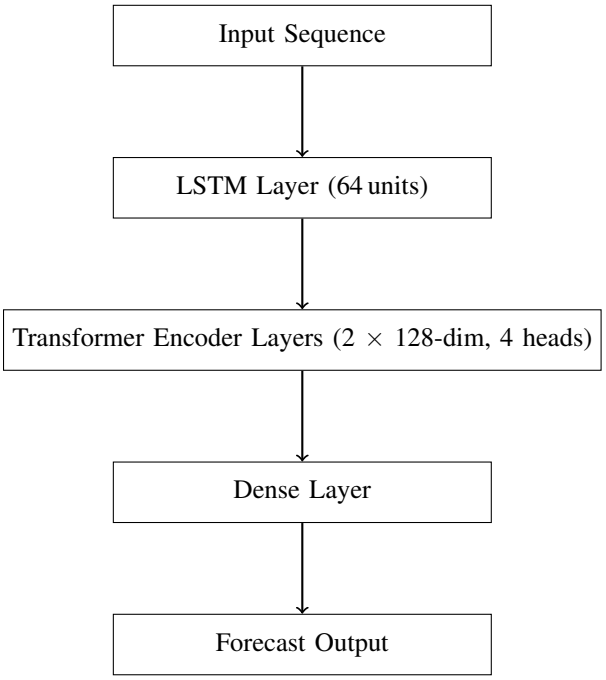

Fig. 3. Hybrid LSTM–Transformer Architecture

4) Training Setup: In every model, we selected the AdamW optimizer, initial learning rate of 1 10 -3, a batch size of 128, and early stopping per validation loss. Each dataset was tuned based on the sequence length which was to be considered in terms of memory efficiency and predictive performance. All the models employed the Mean Squared Error (MSE) as a loss function.

IV. Evaluation

A. Evaluation Metrics

In order to measure the performance of the models, we used four common error measures in time series forecasting, and the latency of inference to compare the efficiency:

Mean Absolute Error (MAE): MAE measures the average magnitude of errors without considering their direction. It is scale-dependent and robust to outliers compared to squared-error metrics.

Root Mean Squared Error (RMSE): RMSE penalizes large errors more heavily than MAE, making it suitable for applications where large deviations have higher costs.

Mean Absolute Percentage Error (MAPE): MAPE expresses error as a percentage of actual values, allowing comparisons across datasets with different scales. However, it can be unstable when actual values are close to zero.

Symmetric Mean Absolute Percentage Error (SMAPE): SMAPE addresses MAPE's instability by using the symmetric average of actual and predicted values in the denominator.

Inference Latency: Measured as the average time (in milliseconds) taken to process a single batch during testing. Lower latency is crucial for real-time deployment.

B. Quantitative Results

1. Accuracy Metrics

TABLE I
Forecasting Performance

Model	MAE	RMSE	MAPE (%)	SMAPE (%)
LSTM [1]	52.1	64.6	3.21	4.85
Transformer [2]	48.7	61.4	2.95	4.32
Hybrid [3]	46.9	59.8	2.84	4.10

Hybrid [3] model gave the minimum errors in all the metrics and this shows that it has greater potential of capturing both standard and long-lasting reliances. Compared to LSTM [1], the Transformer [2] was found to be more accurate, particularly when it comes to RMSE, implying that it will be more resilient to drastic deviations.

2) Latency Comparison

TABLE II
Average Inference Latency per Batch

Model	Power	Traffic	Weather
LSTM [1]	12.4	11.8	13.2
Transformer [2]	8.5	8.1	8.9
Hybrid [3]	9.7	9.3	10.1

It was seen that the Transformer [2] model scored the least average inference latency because of the parallelism capability. After adding the extra LSTM [1] layer to the Hybrid [3] the latency did increase a bit, but the addition of the LSTM [1] layer type did not give the highest level of impairment by far, LSTM [1] had the largest impact on latency because of its sequential nature.

C. Error Analysis

1) Power Dataset: Under predication of all the models was more prone to occur in instances where there were spikes of demand especially in winters. Hybrid [3] model alleviated this problem as opposed to LSTM [1] and Transformer [2] probably because of its simultaneous ability to learn short-term and long-term patterns.

2) Dataset of Traffic: Predictions fared badly when there were sudden traffic spikes due to accidents or weather changes. Transformer [2] was good in capturing regular patterns but poor in capturing sudden deviations.

3) Weather Dataset: Models were less likely to predict sudden changes in temperature like the average values, a trend that showed that it is difficult to predict rare yet effectual events. Hybrid [3] model performed rather accurately in this kind of conditions.

D. Discussion

Such findings point to an accuracy vs. computation efficiency trade-off. Transformer [2] is best suited in situations whereby low latency is imperative whereas Hybrid [3] model in situations where accuracy is paramount. The LSTM [1] is

a reasonably competitive model that is however less accurate and faster, but an optimal choice in terms of the size of the dataset or the weak hardware environment.

V. Conclusions and Future Work

In this paper, three types of deep learning architecture, Long Short-Term Memory (LSTM [1]), Transformer [2], and the Hybrid [3] version of LSTM [1] and Transformer [2] have been introduced and compared in terms of multivariate time series forecasting among three different areas, electricity load (Power), road traffic flow (Traffic) and weathers (Weather). The same preprocessing pipeline was applied to all datasets and each model was measured on accuracy (MAE, RMSE, MAPE, SMAPE) and inference latency.

Comparative results revealed that the Hybrid [3] LSTM-Trans imaginistrained over the entire datasets showed the most significant accuracy in all datasets, showing that Hybrid [3] LSTMTrans imaginitrained over the entire datasets has the potential to address short-term dependency and long-range relationships in an effective way. Transformer [2] showed the shortest inference latency since the Transformer [2] architecture is parallelizable, thus it was best suited to developing real-time application where performance was a factor of importance. LSTM [1] performance was also competitive in some cases, but, being sequential, its latency was higher and the accuracy a bit worse than on the other two architectures.

Domain-specific difficulties were recognized by the analysis of the error:

In Power dataset, all were weak in predicting extreme demand spikes but Hybrid [3] model faired better in predicting them and this was the case because it had mixed architecture. Large short-term variability in the Traffic dataset [5] was a major hubris when it comes to predictive accuracy, especially within LSTM [1] when flow jumps up. Models also tended to regress towards the mean on rare but influential events, e.g. abrupt temperature decreases, in the Weather dataset.

It is based on findings that model selection must be directed by the requirements of application the Hybrid [3] model works well where enough resources to compute are available and where the Internet is not particularly accuracy-sensitive. Transformer [2] will be favorable in cases where latency matters like in the real-time monitoring systems. LSTM [1] is still an option of choice when the scenario involves simpler problems or limited hardware.

Future work on the paper boils down to a few directions: 1. Hyperparameter Optimization: Use Bayesian optimization or genetic search to do automatic hyperparameter tuning of the model on each of the datasets.

2. Exogenous variability: The models should include the external ones such as holiday schedules, calendar events, weather forecasts to increase the situational awareness.

3. Probabilistic Forecasting: generalize the models to produce a probabilistic output via means like Monte Carlo dropout or quantile regression, so it is possible to quantify uncertainty in the results of forecasts.

4. Time Series Data Augmentation: See how we can exploit synthetic data generation (e.g. GANs on time series data) approaches to improve the model generalization of rare event outcomes.

5. Edge Deployment Optimization: Study the models pruning, quantization, and distillation in order to deploy high-performing models on the limited resource devices.

6. Multi-Task and Transfer Learning: Create combined models to predict across a set of related data, minimising training time and exploiting common temporal structure.

Working on these points, the future studies could further improve the predictive accuracy, stability, and efficiency of deep learning-based time series forecasting systems in terms of their deployment. The results of the given research are well-grounded and can be used both in academic research of the topic and in practical application in industries where proper predictions of time series are vital.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[3] Y. Wu, M. Tizzoni, F. Pinheiro, A. Monreale, F. Giannotti, and D. Quercia, "Deep hybrid model for influenza prevalence forecasting," *arXiv preprint arXiv:2001.09708*, 2020.

[4] U. M. L. Repository, "Electricityloaddiagrams20112014 data set," https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014, 2011.

[5] R. Godahewa, K. Bandara, C. Bergmeir, G. Webb, F. Petitjean, and C. Faloutsos, "Monash time series forecasting archive," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[6] M. P. I. for Biogeochemistry, "Jena climate dataset," https://www.kaggle.com/datasets/mnassrib/jena-climate, 2017.

[7] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," in *International Journal of Forecasting*. Elsevier, 2021.