Evaluating the effectiveness of predicting covariates in LSTM Networks for Time Series Forecasting

Gareth Davies

Neural Aspect
garethmd@googlemail.com

Abstract

Autoregressive Recurrent Neural Networks are widely employed in time-series forecasting tasks, demonstrating effectiveness in univariate and certain multivariate scenarios. However, their inherent structure does not readily accommodate the integration of future, time-dependent covariates. A proposed solution, outlined by Salinas et al 2019[1], suggests forecasting both covariates and the target variable in a multivariate framework. In this study, we conducted comprehensive tests on publicly available time-series datasets, artificially introducing highly correlated covariates to future time-step values. Our evaluation aimed to assess the performance of an LSTM network when considering these covariates and compare it against a univariate baseline. As part of this study we introduce a novel approach using seasonal time segments in combination with an RNN architecture, which is both simple and extremely effective over long forecast horizons with comparable performance to many state of the art architectures. Our findings from the results of more than 120 models reveal that under certain conditions jointly training covariates with target variables can improve overall performance of the model, but often there exists a significant performance disparity between multivariate and univariate predictions. Surprisingly, even when provided with covariates informing the network about future target values, multivariate predictions exhibited inferior performance. In essence, compelling the network to predict multiple values can prove detrimental to model performance, even in the presence of informative covariates. These results suggest that LSTM architectures may not be suitable for forecasting tasks where predicting covariates would typically be expected to enhance model accuracy.

1 Introduction

Forecasting future events, is a critical endeavour across various domains, facilitating informed decision-making and resource allocation. These forecasting tasks heavily rely on historical data to make accurate predictions. Given the inherent sequential and time-dependent nature of such data, Recurrent Neural Networks (RNNs) emerge as a natural choice for modelling temporal sequences due to their ability to retain memory across time steps.[2]

However, traditional forecasting methods, particularly autoregression, which relies on using past observations to predict future values, encounter challenges as forecasting horizons extend. As predictions are recursively dependent on preceding values, errors can compound over time, resulting in diminished accuracy.

Furthermore, predicting values solely from prior observations prevents analysis of the underlying features that influence future outcomes. Therefore making it impossible to simulate scenarios or identify corrective action. For example during the Covid pandemic the number of positive tests informed future hospitalisation and mortality rates, or the volume of sales meetings could inform

Preprint. Under review.

an organisation of future sales. Clearly the inclusion of these leading indicators would be highly desirable both in terms of improving accuracy and our understanding of the domain.

To address these limitations and enhance prediction accuracy, researchers have explored the incorporation of additional information, known as covariates, into forecasting models. Covariates can provide valuable context and assist the network in making more informed predictions. They can be either static time-independent, which are fixed for an entire time-series, or time-dependent, where provided values can change at each time step. When future time-dependent covariates are known in advance, such as week or month numbers, they can be leveraged to augment autoregressive predictions. Conversely, in scenarios where future covariates are unknown, they must either be estimated beforehand or predicted simultaneously with the target variable, rendering each prediction multivariate.

In this study, we aim to evaluate the efficacy of employing autoregressive multivariate Long Short-Term Memory (LSTM) models compared to traditional univariate models without covariates. We conduct our investigation within the context of well-established time series datasets from the Monash repository [3], a widely recognised benchmark for evaluating forecasting models. Our methodology involves constructing baseline univariate models using an autoregressive LSTM, ensuring their performance aligns with established benchmarks. Subsequently, we augment each dataset by introducing covariates engineered from future time-step values, effectively guiding the network in predicting both the target variable and the associated covariate. We then assess the performance of our models based on predicting target values over a forecast horizon H.

The contributions of this study are: 1. A novel approach to modelling timeseries data with LSTM networks that is extremely effective in a univariate setting and does not require specialised feature engineering. 2. Quantification of the effect cross correlation between covariates and target variables has on the resulting performance of a trained model. 3. An evaluation of how model performance is affected as forecast horizons increase. 4. A simple approach to synthesise covariates from univariate datasets for running experiments in a controlled and repeatable manner.

2 Related Work

The use of covariates in State Space models have been studied notably by Wang[4] and more recently by Puindi and Silba [5]. Wang proposed ESCov a method that extended HoltWinters to utilise informative covariates which were calculated in advance of the main modelling task. Wang observed that the inclusion of covariates could improve the overall accuracy of the model, the degree to which is in part determined by the quality of the predictions of the covariates.

Lai et al. [6] developed LSTNet, which supports multivariate scenarios and which combines a CNN and a GRU to capture both long and short-term dependencies over time, as well as local dependencies between variables.

Salinas et al. [1] proposed DeepAR, an autoregressive LSTM model intended to be used specifically for time-series forecasting. Their approach included time dependent covariates as inputs into each timestep combining them with the outputs of the previous timestep to assist the network.

Salinas et al subsequently extended DeepAR to a multivariate setting with DeepVAR and GPVAR [7], which supported scenarios where datasets of multiple timeseries are forecasted simultaneously through vector autoregression. The objective being to model the dynamics between the target variables in each timeseries. As the dimension of the output vector scales, the multivariate predictions can become very high dimensional, and encounter high computational costs as a result. GPVAR addresses this by modifying the output model with a low rank covariance structure.

More recently, transformer-based models such as Informer, Autoformer, and Crossformer [8, 9, 10] have emerged for multivariate predictions. Notably Informer used a Weather, WTH, which could be configured to use in a covariate setting. These models extract feature maps from the temporal dimension and extend the vanilla Transformer attention mechanism to produce more accurate predictions over longer forecast horizons. A study by Zeng et al [11] found that the inclusion of covariates in transformer networks were largely ineffective, proposing instead simple linear models with seasonal decomposition similar to traditional state space ETS models.

Chen et al, proposed TSMixer [12], a FCC architecture which reshapes the inputs to force the network to learn across feature and temporal dimension in an attempt to improve multivariate performance.

3 Methodology

In this section, we outline our methodology for assessing the effectiveness of LSTM networks in multivariate time series prediction tasks, particularly focusing on the impact of time-dependent covariates.

3.1 Problem Statement

We consider a dataset containing M timeseries where each timeseries contains T historical observations, $\mathbf{y}$ and covariates $\mathbf{X}$ with k dimensions or individual features such that:

$$\mathbf{y} = [y_1, y_2, \ldots, y_T]$$

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(k)} \\ x_2^{(1)} & x_2^{(2)} & \ldots & x_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ x_T^{(1)} & x_T^{(2)} & \ldots & x_T^{(k)} \end{bmatrix}$$

Let $\mathbf{Z} = \mathbf{y} \oplus \mathbf{X}$. The objective of the training task is to forecast a vector of future values and covariates across a forecast horizon H. At each timestep t, this vector is represented as $\hat{\mathbf{z}}_t = [\hat{y}_t, \hat{x_t}^1, \hat{x_t}^2, \ldots, \hat{x_t}^k]$ where $\hat{y}_t$ is the predicted target variable and $\hat{x_t}^1, \hat{x_t}^2, \ldots, \hat{x_t}^k$ are the predicted future covariates.

During model evaluation, we derive the vector of predicted target variables, $\hat{\mathbf{y}}$ from $\hat{\mathbf{Z}}$ such that $\hat{\mathbf{y}} = [\hat{y_1}, \hat{y_2}, \ldots, \hat{y_H}]$. The aim here is to generate $\hat{\mathbf{y}}$ in a manner that ensures it's more accurate than what could be achieved solely by relying on historical observations.

3.2 Datasets

We selected four publicly available datasets from the Monash repository [3] namely `Hospital` [13], `Tourism` [14], `Traffic` [15] and `Electricity` [16]. These datasets are from diverse domains and have been extensively benchmarked in prior research by Godahewa et al [3] and are commonly used for evaluating time series forecasting models. The `Hospital` dataset consists of 767 monthly time series depicting patient counts related to medical products from January 2000 to December 2006. Similarly, the `Electricity` dataset was used by Lai [6] and represents the hourly electricity consumption of 321 clients from 2012 to 2014. The `Tourism` dataset comprises monthly figures from 1311 tourism-related time series, while the `Traffic` dataset includes 862 weekly time series showing road occupancy rates on San Francisco Bay area freeways from 2015 to 2016. Each dataset is accompanied by specific context lengths C (lookback window) and forecast horizons H, providing a robust evaluation framework.

3.3 Covariate Data Augmentation

Since the original datasets do not contain time-dependent covariates, we artificially augment them to introduce covariates correlated with future target values such that x_t^k is correlated with y_{t+k}. When k=3, for example, the network is supplied with 3 covariates correlated with the following 3 target values. The network would have to learn that the value of the covariate, x_t^k, would take effect on target, y, at a time k timesteps in the future.

Noise is added to each leading indicator to control the level of cross correlation between the covariate and its corresponding target. The noise is calculated as follows:

- For each time series in the dataset the mean μ and standard deviation σ of y are computed.
- For each covariate value a sample ϵ is drawn from a unit normal distribution; $\epsilon \sim N(0, 1)$
- The noise is computed by scaling the random ϵ values by μ and σ and then further scaled by an error level factor γ: $\gamma \in \{0, 0.1, ..., 1.9\}$

Let $\mathbf{x}^{(k)}$ be the augmented covariate input sequence $[x_0, x_1, ..., x_T]$, then $x_t^k = y_{t+k} + \gamma \cdot \mu \cdot \epsilon + \gamma \cdot \sigma \cdot \epsilon$ We then compute the cross correlation between the covariates and the target variables with the Pearson

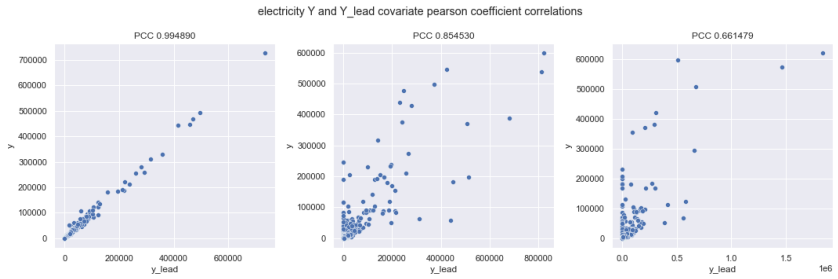


Figure 1: Plots of various correlations between covariate and target variables on the Electricity dataset

correlation coefficient (PCC) and aim for each experiment to be run with a strong to perfect positive correlation in the range 0.5 to 1.0. This value is then directly comparable to autocorrelation of lagged variables used in techniques such as ARIMA as described by Hyndman et al [17].

3.4 Model architectures

To better understand how covariates affect our results and attribute their impact more accurately, we employed two straightforward LSTM based architectures. Unlike the deep learning models benchmarked by Monash [3] which used the GluonTS [18] framework for training, we did not add covariates such as lag variables and static time-series identifiers in order to accurately evaluate the true effect of the augmented leading covariates.

Of these models: base-lstm is used as a baseline and is benchmarked against the Monash repository results using the same context length C and forecast horizons H; while the seg-lstm model is intended to be more capable of forecasting over longer horizons using longer context lengths.

3.4.1 Baseline model (base-lstm)

Baseline is a vanilla LSTM with a scaling strategy inspired by DeepAR[1]. The input data is scaled using mean absolute scaling within each mini-batch. An additional time dependent feature $\log(scale)$ is concatenated with the input vector. The scaled values are fed into a 2 layer LSTM with 40 cells in each layer. The output of the LSTM is then fed into a single fully connected layer with 40 cells followed by ReLU activations. Finally, an output layer generates a vector $\mathbf{z_t}$ containing the predicted target variable $\hat{y}_t$ and predicted covariates $\hat{\mathbf{x}}_\mathbf{t}$. An inverse scaling operation finally transforms the predictions back into the original scale.

3.4.2 Segment model (seg-lstm)

The seg-lstm model addresses the limitation of the baseline model in forecasting over longer time horizons. Our approach aims to maximise historical data without relying on extra features, such as lag variables that get applied with GluonTS [18] models and is typical with transformer based approaches [8, 9, 10, 19].

To achieve this, we propose an LSTM with a simple modification to handle input data. We reshape each context window of C values into a vector of dimension $\frac{C}{d} \times d$, where d represents the segment length, typically set to the seasonality of the data (e.g., 24 for hourly readings).

The model architecture is similar to the base-lstm, consisting of a straightforward LSTM with 2 layers and 40 cells in each layer, followed by fully connected layers with ReLU activations. We use mean absolute scaling but omit the additional time-dependent feature $\log(\text{scale})$.

During training, we compute the loss by comparing the last timestep of each segment output to the corresponding timestep in the target vector shifted one step into the future. The network is trained to predict one timestep ahead from each segment, with hidden state representations derived from segments are separated by a time period d.

During inference, we predict the next timestep vector, append it to the previous timestep segment, and drop the oldest vector $\mathbf{z}_{t_0}$. This forms an autoregressive loop, predicting successive segments of adjacent timesteps rather than spanning a fixed time period d.

3.5 Evaluation Metrics

We evaluate the performance of each model using MAE, RMSE, and sMAPE, computed on the raw (i.e. unscaled) values of the dataset. This approach ensures direct comparability with results from the Monash archive[3], enabling comprehensive assessment and comparison.

$$\text{MAE}(\hat{Y}, Y) = \frac{\sum_{t=1}^{H} |\hat{Y}_t - Y_t|}{H} \qquad \text{RMSE}(\hat{Y}, Y) = \sqrt{\frac{\sum_{t=1}^{H} (\hat{Y}_t - Y_t)^2}{H}}$$

$$\text{sMAPE}(\hat{Y}, Y) = \frac{100\%}{H} \sum_{t=1}^{H} \frac{|\hat{Y}_t - Y_t|}{(|\hat{Y}_t| + |Y_t|)/2}$$

4 Experimental Setup

We now describe the details of the experiments including how the data was processed, the model training and evaluation.

4.1 Data Processing

To facilitate model training, we generate time windows of data, each of which is equal in size to the sum of the context length (C) and the forecast horizon (H). During each training epoch, complete and overlapping windows are extracted from each time series.

The dataset is partitioned into training, validation, and test sets, following a chronological order. The validation set excludes values within the last forecast horizon, while the training set excludes values from the last two forecast horizons. Evaluation for both validation and testing is conducted on the last complete forecast horizon period of each time series. This approach ensures direct comparison to the Monash benchmarks [3] and maximises the utilisation of training data while preventing data leakage between the three sets.

We employ a variation of mean absolute scaling to standardise the data within each mini-batch (local normalisation), accounting for batches with values equal to zero.

$$\text{scale}(\mathbf{z}) = \max\left(\frac{\sum_{t=1}^{C} |z_t^{(y)}|}{C}, 1\right) \tag{1}$$

4.2 Training

We utilised teacher forcing which trains the network to predict a single timestep ahead and calculated the loss for the entire sequence (ie the context length and the forecast horizon). The network was trained as a regression task with a Smooth F1 Loss objective.

$$\text{SmoothL1Loss}(\hat{z}, z) = \begin{cases} 0.5(\hat{z} - z)^2 & \text{if } |\hat{z} - z| < 1 \\ |\hat{z} - z| - 0.5 & \text{otherwise} \end{cases} \tag{2}$$

A number of hyperparameters were set to match the default GluonTS parameters [18] including weight decay of $1e-8$; dropout of 0.1; and a learning rate of 0.001. We used OneCycle scheduling to anneal and then decrease the learning rate[20] over 100 epochs. The models were optimised using AdamW. We made no attempt to optimise hyperparameters.

Performance was measured on the validation set at the end of each epoch by measuring the Smooth L1 loss using free running (i.e. using the predicted outputs of one timestep as the input into the next). The model checkpoint with the lowest validation loss was selected for testing. The hidden state of the LSTM was initialised to 0.

Table 1: Benchmark results for each dataset

Models		FFNN	DeepAR	N-Beats	Wavenet	Transformer	base-lstm*	seg-lstm*
Hospital (12)	sMAPE	18.33	**17.45**	17.77	17.55	20.08	17.52	18.05
	MAE	22.86	18.25	20.18	19.35	36.19	**18.03**	19.95
	RMSE	27.77	**22.01**	24.18	23.38	40.48	22.03	24.19
Tourism (24)	sMAPE	20.11	**18.35**	20.42	18.92	19.75	21.50	19.85
	MAE	2022.21	**1871.69**	2003.02	2095.13	2146.98	2336.42	1956.07
	RMSE	2584.10	**2359.87**	2596.21	2694.22	2660.06	2964.96	2413.64
Traffic (8)	sMAPE	12.73	13.22	1**2.40**	13.30	15.28	12.77	12.97
	MAE	1.15	1.18	**1.11**	1.20	1.42	1.15	1.17
	RMSE	1.55	1.51	**1.44**	1.61	1.94	1.56	1.58
Electricity (168)	sMAPE	23.06	**20.96**	23.39	-	24.18	34.12	21.20
	MAE	354.39	329.75	350.37	**286.56**	398.80	525.50	287.95
	RMSE	519.06	477.99	510.91	489.91	514.68	675.03	**469.07**

* Mean values from 5 runs.
Forecast horizons are given in the brackets

4.3 Experimental Procedure

Initially, we trained base-lstm models for each dataset using the context length and forecast horizon values consistent with those used in the Monash repository which allowed us to compare the performance of our models against benchmarked results.

Next, we trained and evaluated models under various scenarios involving 1, 2 and 3 covariates. For the number of covariates in the scenario a set of base-lstm models were trained with increased noise to vary the level of cross correlation with the target variables. This allows us to assess the impact of covariates on model performance under different correlation levels. These models were trained to a context length and forecast horizon that equalled the benchmarked results from Monash [3]

We then trained seg-lstm models using the same procedure with the exception that we use a longer context length C which was set to 3x the forecast horizon H on `Hospital`, `Tourism` and `Electricity` and 8x the forecast horizon on `Traffic`.

5 Results

Table 1 presents the mean error metrics for each dataset evaluated against the neural network architectures benchmarked by Godahewa et al [3]. Among these architectures, FFNN and N-BEATS [21] are fully-connected models, while DeepAR [1] is based on an LSTM network. WaveNet [22], originally designed for audio synthesis, was adapted by Alexandrov et al [18] for time-series tasks in GluonTS. Additionally, the Transformer architecture closely follows the implementation described in the original paper by Vaswani et al [23].

Comparing the base-lstm model across datasets, it yielded comparable results on the `Hospital` and `Traffic` datasets, slightly inferior results on `Tourism`, and significantly poorer results on `Electricity` when measured by sMAPE. However, the performance is marginally better when evaluated using MAE and RMSE. Notably, the base-lstm model exhibited relatively better performance on datasets with shorter forecast horizons.

The seg-lstm model emerged as the top performer on Electricity with RMSE and second with MAE and sMAPE, whilst on `Tourism` it ranked second with MAE and RMSE , both of which have longer forecast horizons. Overall, both of our models demonstrated comparable performance to the benchmarks on the shorter forecast horizons of `Hospital` and `Traffic`, while the seg-lstm model also displayed competitive performance on `Tourism` and `Electricity`.

Table 2: sMAPE results for covariates $k \in \{0, 1, 2, 3\}$ and cross correlation $PCC \in \{1, 0.9, 0.5\}$

Models		**base-lstm**			**seg-lstm**		
PCC		1	0.9	0.5	1	0.9	0.5
Hospital (12)	k=0	17.52 ± 0.041	17.52 ± 0.041	17.52 ± 0.041	18.05 ± 0.135	18.05 ± 0.135	18.05 ± 0.135
	k=1	16.13	17.45	17.77	16.07	17.96	18.09
	k=2	14.94	17.65	18.06	17.33	18.49	18.51
	k=3	13.49	17.71	17.72	17.71	17.68	18.77
Tourism (24)	k=0	21.50 ± 0.531	21.50 ± 0.531	21.50 ± 0.531	19.85 ± 0.62	19.85 ± 0.62	19.85 ± 0.62
	k=1	20.54	22.17	28.18	19.14	20.32	21.56
	k=2	20.26	24.33	27.64	19.07	20.49	21.84
	k=3	23.08	27.59	28.13	18.61	19.96	23.06
Traffic (8)	k=0	12.77 ± 0.065	12.77 ± 0.065	12.77 ± 0.065	12.97 ± 0.108	12.97 ± 0.108	12.97 ± 0.108
	k=1	11.36	12.59	12.89	11.64	12.85	13.00
	k=2	10.73	12.15	12.62	11.10	13.12	12.86
	k=3	9.57	11.86	13.51	9.88	11.94	12.90
Electricity (168)	k=0	34.12 ± 2.38	34.12 ± 2.38	34.12 ± 2.38	21.20 ± 0.232	21.20 ± 0.232	21.20 ± 0.232
	k=1	33.10	30.41	33.95	21.01	22.37	21.45
	k=2	33.62	30.42	38.16	21.14	22.48	22.11
	k=3	36.18	33.83	31.50	21.61	23.04	22.20

Forecast horizons given in the brackets. Univariate k=0 intervals are 95% CI.

5.1 Covariates

The results of covariates compared to univariate cases are presented in table 2. For brevity we have included PCC values of 1, 0.9 and 0.5 (See appendix for details of complete results) and show the sMAPE for each dataset at the full forecast horizons as measured in the benchmark tests. We present the results for each value of k covariates tested (1, 2 and 3) and compare it to the univariate case where k=0. To some extent the relative differences between the covariate and univariate cases are somewhat specific to the dataset. The models trained on `Traffic` appear to benefit the most from covariates giving improved results in 14 of the 18 experiments, which is perhaps to be expected as this has the shortest forecast horizon. Conversely models performed worst on, `Tourism` only outperforming univariate in 5 scenarios, and that was with perfectly correlated covariates. Electricity, which has the longest forecast horizon of 168, marginally outperformed univariate in both models with PCC = 1.0 using 1 and 2 covariates.

5.1.1 Comparing PCC at forecast horizons = k

We are interested in analysing the performance of models in the special cases where the forecast horizon matches the number of covariates, as in these cases the model is effectively provided with information required to predict each timestep. Specifically, we wanted to understand the role that the cross correlation plays on model performance. Figure 2 shows how the base-lstm model sMAPE degrades as a function of PCC on each of the datasets where the forecast horizon is 3. Models with 1 and 2 covariates exhibit similar traits, see Appendix. Unsurprisingly the performance is best when the correlation is perfect (ie PCC = 1.0) to the extent that an almost perfect prediction was obtained in the cases of `Traffic` and `Hospital`. However the performance degrades as the PCC decreases in a non linear manner, with the error increasing rapidly as the PCC falls to 0.9. Interestingly even in this highly artificial and advantageous setting, the error on `Tourism` and `Electricity` are still worse than the univariate case at correlation levels less than 0.9.

5.1.2 Comparing PCC over longer forecast horizon trajectories

With the exception of `Traffic`, as the forecast horizon extends any performance advantage over a univariate setting diminishes for a PCC value of 0.9 and below when forecast horizon exceeds 3-4 timesteps. Figure 3 plots the base-lstm sMAPE as a function of forecast horizon on the `Tourism` dataset.

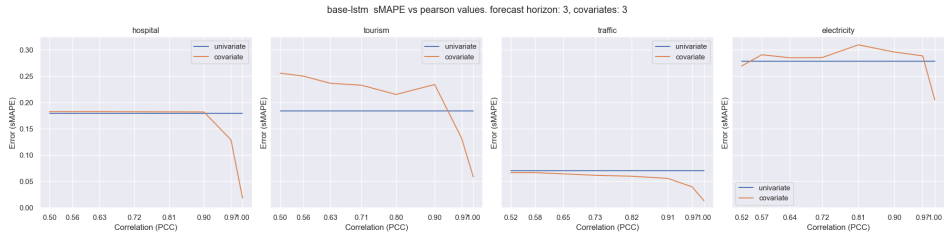


Figure 2: base-lstm smape as a function of PCC for 3 covariates

5.1.3 Comparing base-lstm and seg-lstm performance

Looking at the differences between base-lstm and seg-lstm. We see that both models share common characteristics when examining the errors on forecast horizon trajectories. The stronger model in univariate scenarios will also perform better in a covariate setting. Figure 3 shows how the sMAPE accumulates over the full forecast horizon on the `Tourism` dataset for both base-lstm and seg-lstm. Note how the variance of errors is smaller across the range of PCC values on the stronger seg-lstm model, indicating that the additional noise added to lower PCC covariates combines with the model's inherent performance limitations to compound errors over time.

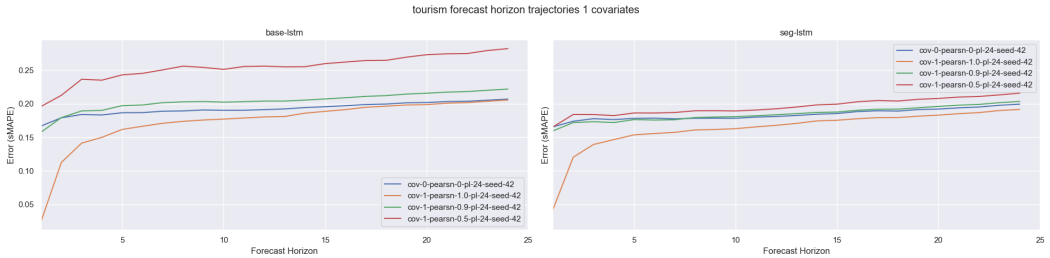


Figure 3: `Tourism` smape for 1 covariate for base-lstm and seg-lstm

5.1.4 Comparing covariates across Forecast Horizon trajectories

Turning to comparing multiple covariates. Fig 4 shows sMAPE across full forecast horizons for 1, 2 and 3 covariates on the `Traffic` dataset with the special case of perfect correlation (PCC = 1). Note that the error reduces as the number of covariates increases. Interestingly, the addition of each covariate shifts the onset of significant errors by one timestep. (i.e. errors start to accumulate at t=1 for k=1, at t=2 for k=2 and t=3 for k=3). We observe identical patterns on the other datasets tested. Clearly the network is effectively utilising the covariates to learn the values for the first k

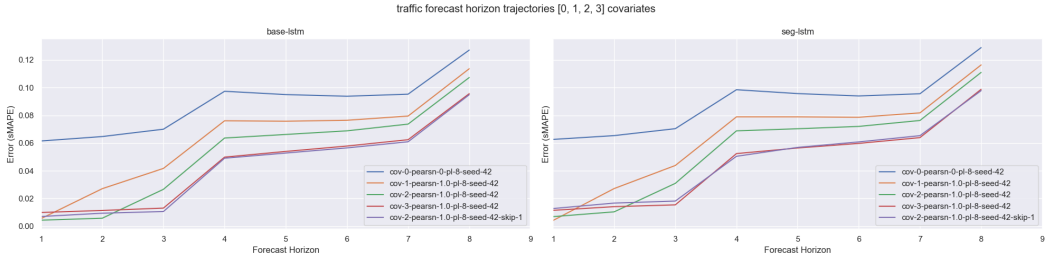


Figure 4: `Traffic` smape comparing univariate to 1, 2 and 3 covariates for base-lstm at a PCC = 0.9

timesteps, however it is conceivable that a model may be using covariate values from the current timestep alone rather than utilising covariates from previous timesteps. For example in the case of 3 covariates x_t^1, x_t^2, x_t^3 are leading indicators for target values y_t, y_{t+1}, y_{t+2}, and therefore the network

does not require covariates from previous timesteps in order to predict the subsequent 3 timesteps to produce the error improvement that we observe. Consequently we don't know if the model is utilising covariates across the temporal dimension or if it is just using the current timestep's covariates.

One way to help answer this is to omit the x_t^2 covariate meaning that at any given timestep the input vector contains lead indicators for y_t and y_{t+2} and the model would have to obtain the leading covariate for y_{t+1} from x_{t-1}^3 (ie the previous timestep). If the error trajectory in this example continues to exhibit the same 3 timestep offset that we observe with 3 covariates then we reason that the model must be learning across both temporal and feature dimensions simultaneously. Figure 4 illustrates the comparison between the univariate and covariate cases. It is evident from the plotted series, labelled as *cov-2-pearsn-1.0-pl-8-seed-42-skip-1*, that the error closely resembles that of using 3 covariates.

6 Discussion

We have presented a repeatable method for testing the effectiveness of predicting covariates in neural networks that could be employed for alternative model architectures. Furthermore we have developed an LSTM model (seg-lstm) capable of producing state of the art results in a univariate setting on long forecast horizons without necessitating the need for additional features.

Analysis of results has led us to reach the following conclusions:

- An LSTM is able to model both temporal and feature dynamics at short forecast horizons in the presence of short lead timestep covariates.
- The effect covariates have on performance is, in part, related to the characteristics of the task (dataset) that the model is trained on.
- The magnitude of the performance gain is related to the strength of the correlation between the covariate and the target variables and that this relationship is nonlinear with performance degrading most rapidly for PCC levels between 1.0 and 0.9.
- In most cases the presence of predicted covariates becomes either ineffective or a hindrance to performance as forecast horizons extend.
- The number of timesteps covered by multiple covariates can offset the accumulation of error equal to the number of timesteps.
- The magnitude of the performance advantage from using covariates is related to the model's underlying ability to forecast accurately. Models that perform better in a univariate setting will produce comparatively better results in a covariate setting.
- Increasing the number of covariates can enhance performance on short forecast horizons with the magnitude of the performance being related to the PCC

Given the aim of this study was to explore the effect of covariates on LSTM networks, it's evident that predicting covariates jointly with target variables can under very limited conditions result in a performance improvement. As forecast horizons extend however, it frequently hinders performance. While acknowledging the artificial nature of the experiments used in this study it nonetheless underscores some potential for covariates to assist neural networks in making more accurate predictions.

It may be possible that the vanilla LSTM's struggle to learn the relationships between covariates and target variables across long forecast horizons and that alternative architectures and/or additional feature engineering techniques (such as LSTNet [6] and the transformer based models of Zhou, Kitaev and Zhang [8, 19, 10]) may perform better and a future study could evaluate their effectiveness using the same conditions.

Future studies could also evaluate the use of covariates with more representative real-world data, such as the work by Zhou [8] using the weather dataset which contains multiple features related to predicting weather. Additionally, further exploration with artificial datasets could investigate the effects of other characteristics like extending the covariate lead time offset on target variables, non-stationarity or negative correlations. This raises the question of whether providing the network with information regarding the temporal influence of each leading indicator on the target variable could simplify the learning task.

References

[1] David Salinas, Valentin Flunkert, and Jan Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks, 2019.

[2] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: https://doi.org/10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.

[3] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *NeurIPS Datasets and Benchmarks*. Curran Associates, Inc., 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/eddea82ad2755b24c4e168c5fc2ebd40-Abstract-round2.html.

[4] Shuchun Wang. Exponential smoothing for forecasting and bayesian validation of computer models, 01 2006.

[5] AC Puindi and ME Silva. Dynamic structural models with covariates for short-term forecasting of time series with complex seasonal patterns. *J Appl Stat*, 48(5):804–826, 2020. doi: 10.1080/02664763.2020.1748178.

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

[7] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes, 2019.

[8] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021.

[9] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, 2022.

[10] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

[11] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022.

[12] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting, 2023.

[13] Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. Hospital dataset, apr 2021. URL https://doi.org/10.5281/zenodo.4656014.

[14] Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. Tourism monthly dataset, apr 2021. URL https://doi.org/10.5281/zenodo.4656096.

[15] Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. Traffic weekly dataset, apr 2021. URL https://doi.org/10.5281/zenodo.4656135.

[16] Rakshitha Godahewa, Christoph Bergmeir, Geoff Webb, Rob Hyndman, and Pablo Montero-Manso. Electricity hourly dataset, apr 2021. URL https://doi.org/10.5281/zenodo.4656140.

[17] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, Melbourne, Australia, 3 edition, 2021. URL https://OTexts.com/fpp3. Accessed on 2024-03-21.

[18] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, L. Stella, A. C. Türkmen, and Y. Wang. GluonTS: Probabilistic Time Series Modeling in Python. *arXiv preprint arXiv:1906.05264*, 2019.

[19] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer, 2020.

[20] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018.

[21] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting, 2020.

[22] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the claims and findings of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the study are discussed in the discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes full details of the experimental setup, model architectures and datasets are provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides open access to the code and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are provided in the main content and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Univariate test results are presented with confidence intervals. Covariate results are not, however as the covariate experiments generate 72 models

   per model architecture we do not believe it necessary to determine the error bars of each individual model as we examine groups of models together and the variance

   of the models in these groups is clear to see in the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational details are provided in the Appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPs Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work in the paper has no potential for societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The only applicable assets are the datasets which are credited and distributed under a Creative Commons Attribution License

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets include the code required to run the experiments described in the paper. Documentation is provided along with the code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Appendix / supplemental material

A.1 Datasets

All datasets used in this study are available through the Monash repository https://forecastingdata.org/ and are distributed under a Creative Commons Attribution 4.0 International licence.

A.2 hyperparameters

Table 3: hyperparameters

learning rate	0.001
epochs	100
hidden cells	40
lstm hidden layers	2
dropout	0.1
weight decay	1e-8
batch size	128
batches per epoch	200 (base-lstm) 500 (seg-lstm)
early stopping patience	30
optimizer	AdamW
scheduler	OneCycle

A.3 Computational Details

All models were trained using CPU.
memory: 18 GB
os: macOS-10.16-x86_64-i386-64bit
python 3.11

A.4 Source Code

The models were developed using Pytorch. Source code is available at https://github.com/garethmd/nnts/tree/rnn-covariates-paper

Table 4: total model training times

Dataset	Model	
	base-lstm (seconds)	**seg-lstm** (seconds)
Electricity	70442	51839
Hospital	11029	1151
Tourism	20412	6807
Traffic	15667	2652

* Each dataset is trained with 30 models.

A.5 Additional Results

CSV Files of the full results are available on Google Drive
https://drive.google.com/drive/folders/179UK0lOOSi7yRdXOhcd7AFKl9U9A1DpE?usp=sharing In this section we present additional tables and plots of results which supplement the results in the main paper.

Table 5 details the MAE error for the base-lstm model for each dataset with various k covariates and correlation PCC values.

Table 6 details the MAE error for the seg-lstm model for each dataset with various k covariates and correlation PCC values.

Table 7 details the RMSE error for the base-lstm model for each dataset with various k covariates and correlation PCC values.

Table 8 details the RMSE error for the seg-lstm model for each dataset with various k covariates and correlation PCC values.

Table 5: MAE base-lstm results for covariates $k \in \{0, 1, 2, 3\}$ and cross correlation $PCC \in \{1, 0.9, 0.5\}$ values.

Models		**base-lstm**		
PCC		1	0.9	0.5
Hospital (12)	k=0	18.03 $\pm$ 0.306	18.03 $\pm$ 0.306	18.03 $\pm$ 0.306
	k=1	16.71	19.03	20.18
	k=2	15.62	20.31	20.77
	k=3	14.49	20.18	20.23
Tourism (24)	k=0	2336.42 $\pm$ 147.6	2336.42 $\pm$ 147.6	2336.42 $\pm$ 147.6
	k=1	2131.98	2545.43	3437.56
	k=2	2352.32	3734.32	4014.21
	k=3	3593.42	3340.92	3574.85
Traffic (8)	k=0	1.15 $\pm$ 0.010	1.15 $\pm$ 0.010	1.15 $\pm$ 0.010
	k=1	1.03	1.14	1.16
	k=2	0.95	1.09	1.14
	k=3	0.84	1.06	1.23
Electricity (168)	k=0	525.50 $\pm$ 51.74	525.50 $\pm$ 51.74	525.50 $\pm$ 51.74
	k=1	505.01	526.33	596.59
	k=2	528.35	452.38	519.68
	k=3	631.98	510.47	525.58

Forecast horizons given in the brackets. Univariate k=0 intervals are 95% CI.

Table 6: MAE seg-lstm results for covariates $k \in \{0, 1, 2, 3\}$ and cross correlation $PCC \in \{1, 0.9, 0.5\}$ values.

Models		**seg-lstm**		
PCC		1	0.9	0.5
Hospital (12)	k=0	19.95 ± 0.309	19.95 ± 0.309	19.95 ± 0.309
	k=1	16.73	20.83	21.55
	k=2	20.05	23.14	23.46
	k=3	21.34	22.22	24.93
Tourism (24)	k=0	1956.07 ± 163.4	1956.07 ± 163.4	1956.07 ± 163.4
	k=1	2078.31	3180.39	2292.99
	k=2	1782.48	3176.35	2857.37
	k=3	2053.37	2136.06	2790.61
Traffic (8)	k=0	1.17 ± 0.021	1.17 ± 0.021	1.17 ± 0.021
	k=1	1.04	1.16	1.17
	k=2	0.98	1.19	1.15
	k=3	0.87	1.07	1.16
Electricity (168)	k=0	287.95 ± 50.88	287.95 ± 50.88	287.95 ± 50.88
	k=1	286.77	305.21	290.34
	k=2	288.11	306.79	297.81
	k=3	298.37	319.63	312.56

Forecast horizons given in the brackets. Univariate k=0 intervals are 95% CI.

Table 7: RMSE base-lstm results for covariates $k \in \{0, 1, 2, 3\}$ and cross correlation $PCC \in \{1, 0.9, 0.5\}$ values.

Models		**base-lstm**		
PCC		1	0.9	0.5
Hospital (12)	k=0	22.03 ± 0.339	22.03 ± 0.339	22.03 ± 0.339
	k=1	20.95	23.22	24.53
	k=2	20.18	24.62	25.15
	k=3	19.65	24.51	24.55
Tourism (24)	k=0	2964.96 ± 155.8	2964.96 ± 155.8	2964.96 ± 155.8
	k=1	2764.97	3178.28	4409.48
	k=2	3116.47	4706.51	5015.75
	k=3	4575.04	4361.75	4730.95
Traffic (8)	k=0	1.56 ± 0.010	1.56 ± 0.010	1.56 ± 0.010
	k=1	1.47	1.55	1.58
	k=2	1.40	1.49	1.52
	k=3	1.28	1.46	1.68
Electricity (168)	k=0	675.03 ± 5.865	675.03 ± 5.865	675.03 ± 5.865
	k=1	650.48	689.42	734.66
	k=2	680.63	606.75	666.07
	k=3	783.84	675.23	668.15

Forecast horizons given in the brackets. Univariate k=0 intervals are 95% CI.

Table 8: RMSE seg-lstm results for covariates $k \in \{0, 1, 2, 3\}$ and cross correlation $PCC \in \{1, 0.9, 0.5\}$ values.

Models		**seg-lstm**		
PCC		1	0.9	0.5
Hospital (12)	k=0	24.19 ± 0.434	24.19 ± 0.434	24.19 ± 0.434
	k=1	20.99	25.47	26.41
	k=2	24.55	27.93	28.27
	k=3	26.12	26.92	30.01
Tourism (24)	k=0	2964.96 ± 155.8	2964.96 ± 155.8	2964.96 ± 155.8
	k=1	2612.12	3803.94	2833.16
	k=2	2228.58	3875.71	3497.33
	k=3	2647.94	2660.67	3467.14
Traffic (8)	k=0	1.56 ± 0.010	1.56 ± 0.010	1.56 ± 0.010
	k=1	1.48	1.57	1.57
	k=2	1.44	1.61	1.55
	k=3	1.32	1.45	1.53
Electricity (168)	k=0	469.07 ± 7.734	469.07 ± 7.734	469.07 ± 7.734
	k=1	481.60	482.13	472.95
	k=2	474.75	488.84	473.14
	k=3	481.93	500.84	492.91

Forecast horizons given in the brackets. Univariate k=0 intervals are 95% CI.

Figures 5 plots the sMAPE as a function of correlation (PCC) for base-lstm models

Figures 6 plots the sMAPE as a function of correlation (PCC) for seg-lstm models

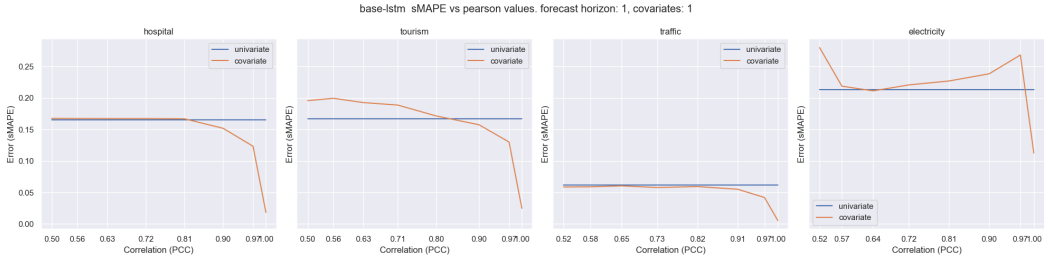
Figures 7 plots sMAPE for various covariates as a function of Forecast Horizon for correlation (PCC) = 1.0

Figures 8 plots sMAPE hospital forecast horizon trajectories for various covariates k and correlation values (PCC)

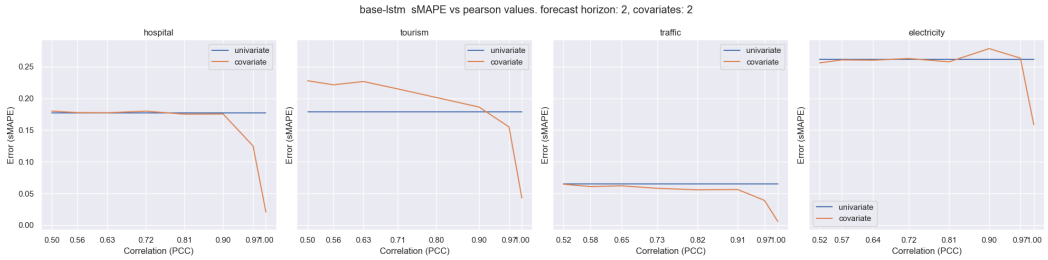
Figures 9 plots sMAPE tourism forecast horizon trajectories for various covariates k and correlation values (PCC)

Figures 10 plots sMAPE traffic forecast horizon trajectories for various covariates k and correlation values (PCC)

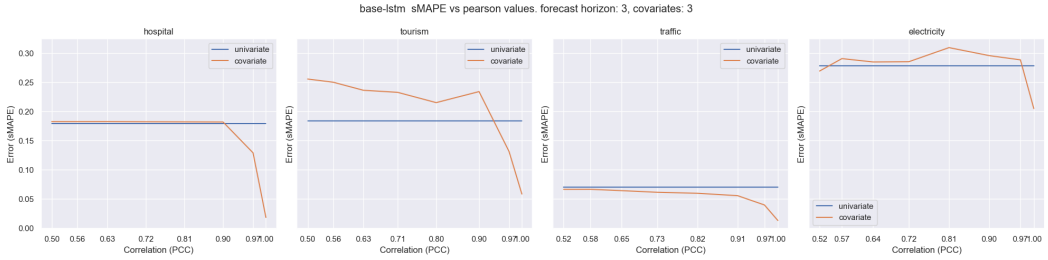
Figures 11 plos sMAPE electricity forecast horizon trajectories for various covariates k and correlation values (PCC)



(a) base-lstm smape as a function of PCC for 1 covariate

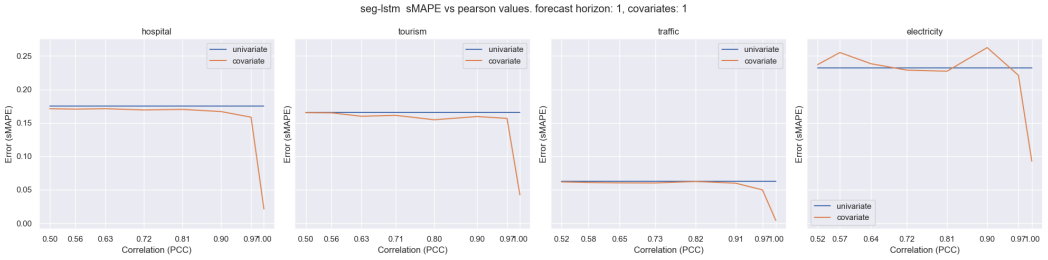


(b) base-lstm smape as a function of PCC for 2 covariates

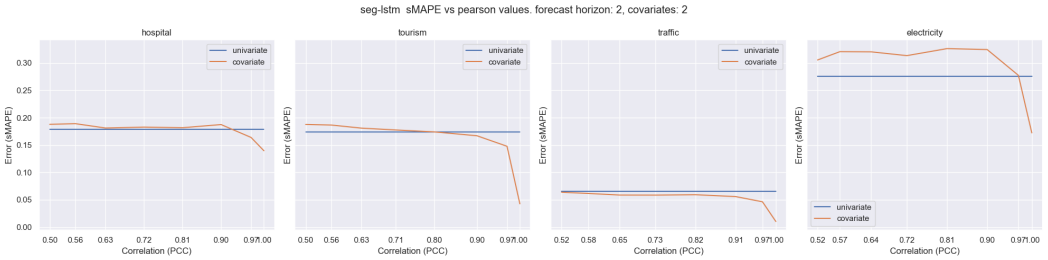


(c) base-lstm smape as a function of PCC for 3 covariates

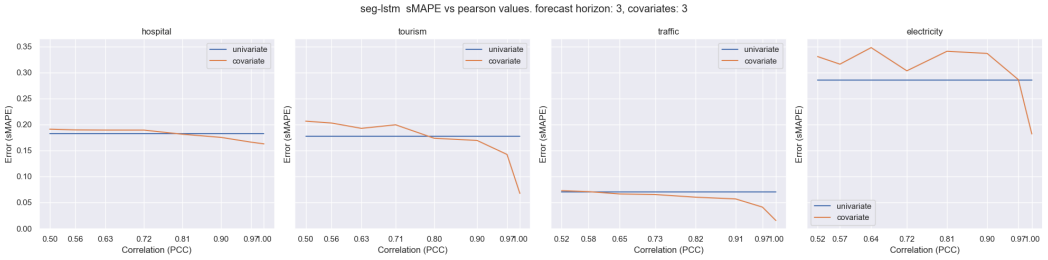
Figure 5: base-lstm sMAPE for various covariates as a function of correlation (PCC)



(a) seg-lstm smape as a function of PCC for 1 covariate

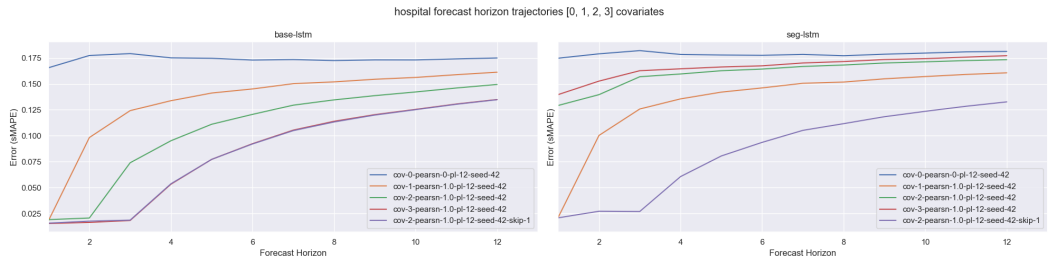


(b) seg-lstm smape as a function of PCC for 2 covariates

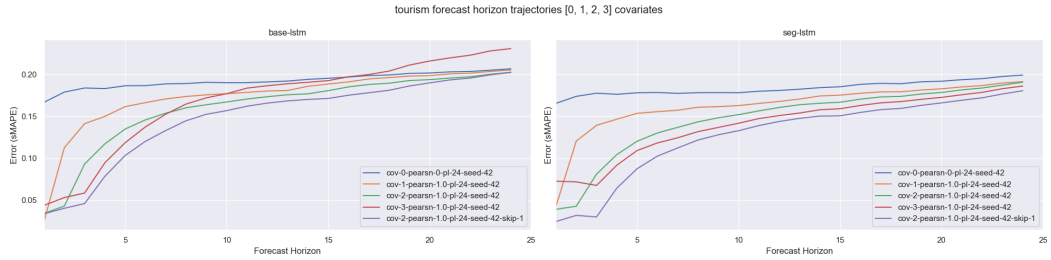


(c) seg-lstm smape as a function of PCC for 3 covariates

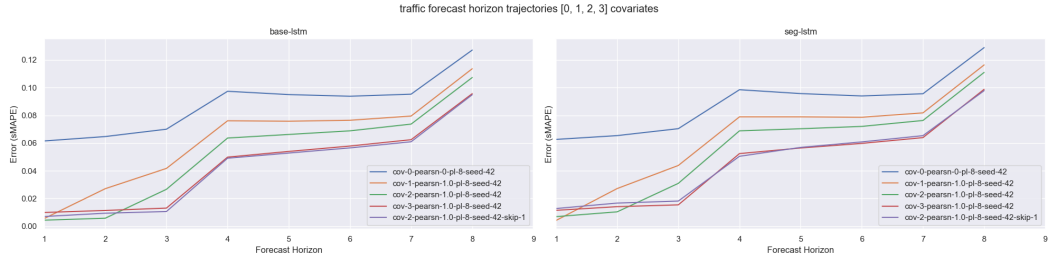
Figure 6: seg-lstm sMAPE for various covariates as a function of correlation (PCC)



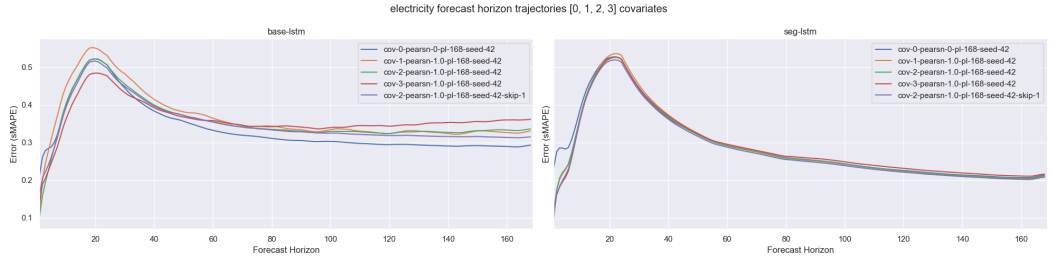
(a) hospital forecast horizon trajectories



(b) tourism forecast horizon trajectories

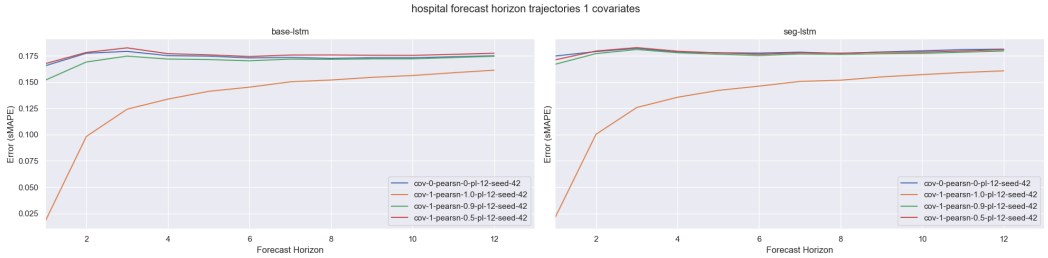


(c) traffic forecast horizon trajectories

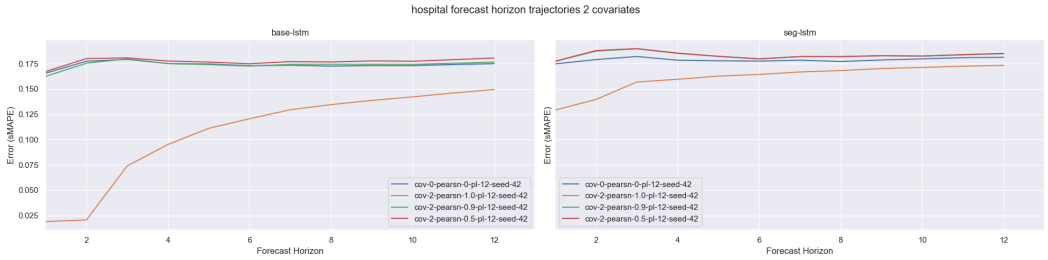


(d) electricity forecast horizon trajectories

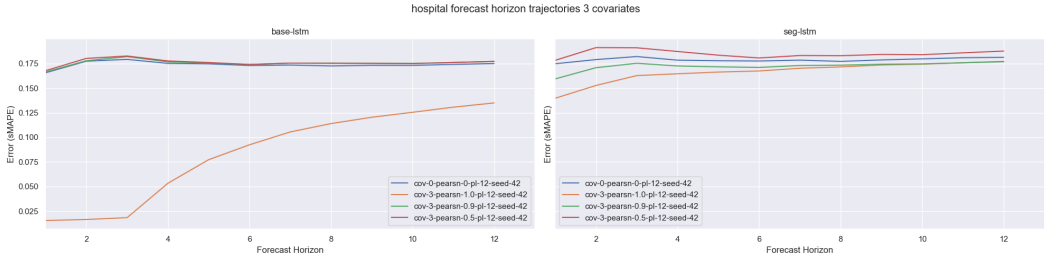
Figure 7: sMAPE for various covariates as a function of Forecast Horizon for correlation (PCC) = 1.0



(a) hospital forecast horizon trajectories k=1

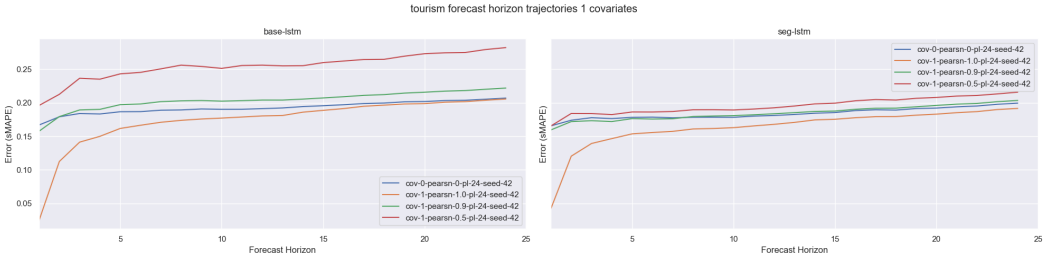


(b) hospital forecast horizon trajectories k=2

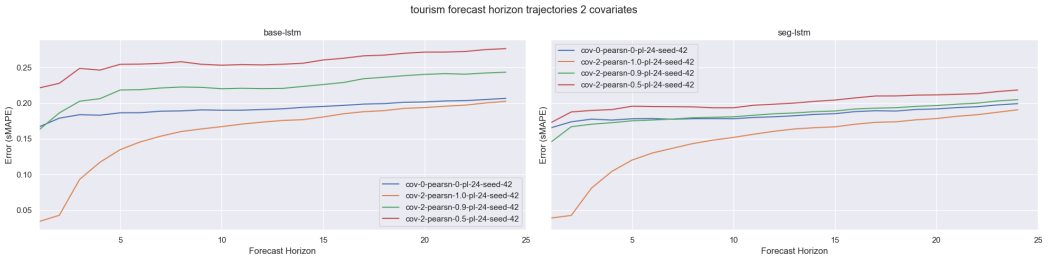


(c) hospital forecast horizon trajectories k=3

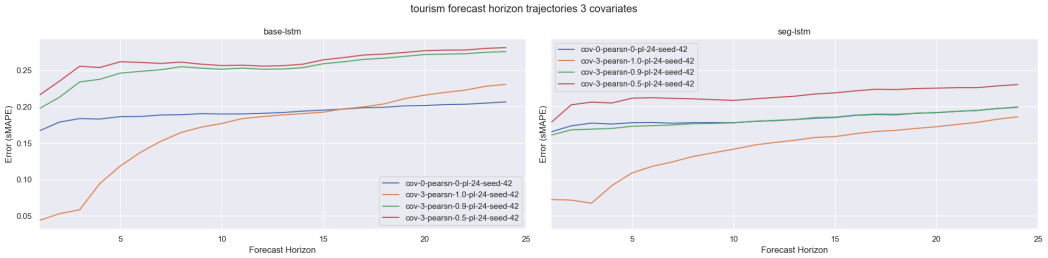
Figure 8: hospital forecast horizon trajectories for various covariates k and correlation values (PCC)



(a) tourism forecast horizon trajectories k=1

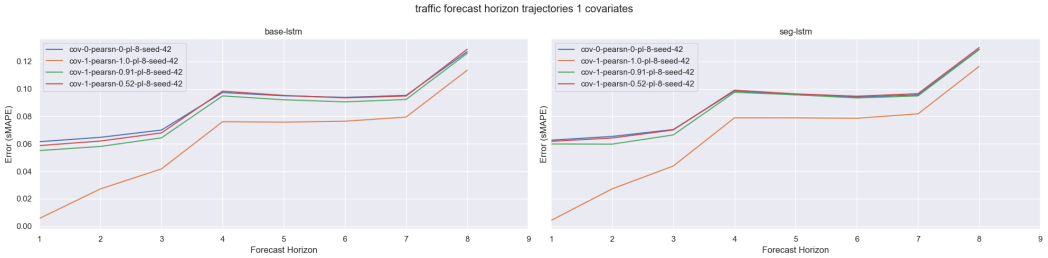


(b) tourism forecast horizon trajectories k=2

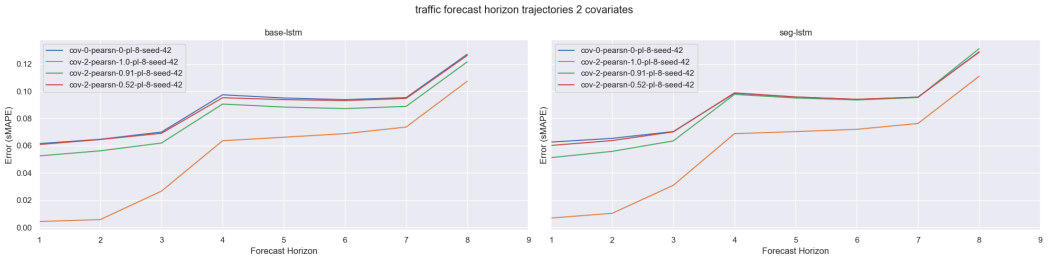


(c) tourism forecast horizon trajectories k=3

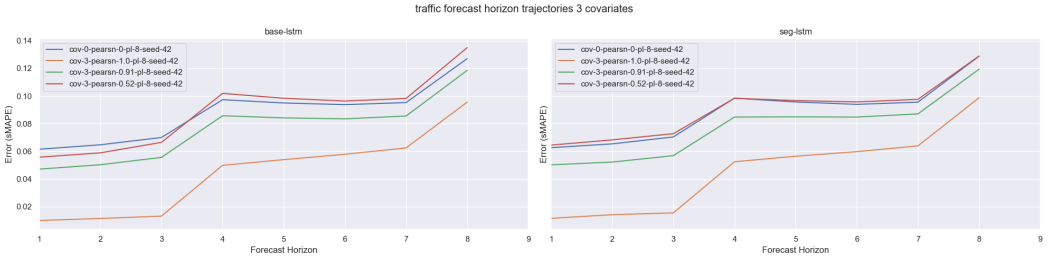
Figure 9: tourism forecast horizon trajectories for various covariates k and correlation values (PCC)



(a) traffic forecast horizon trajectories k=1

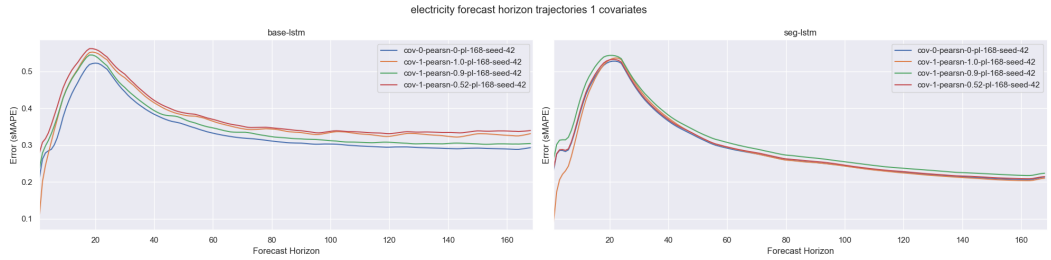


(b) traffic forecast horizon trajectories k=2

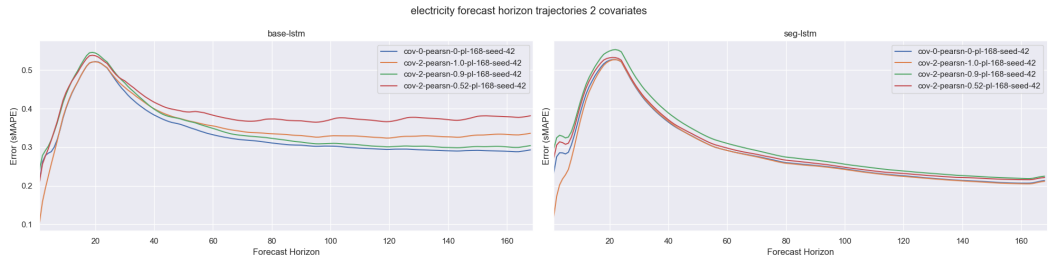


(c) traffic forecast horizon trajectories k=3

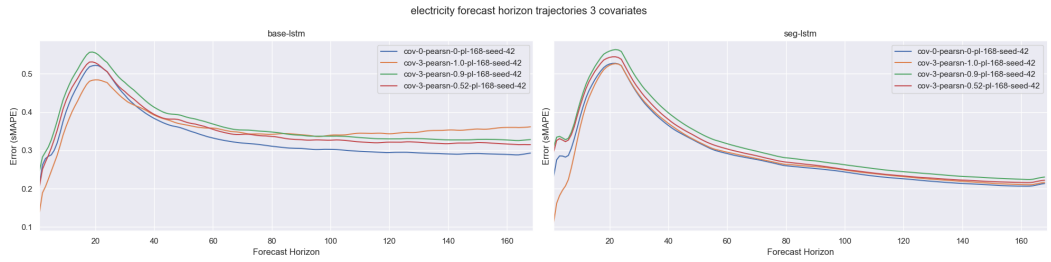
Figure 10: traffic forecast horizon trajectories for various covariates k and correlation values (PCC)



(a) electricity forecast horizon trajectories k=1



(b) electricity forecast horizon trajectories k=2



(c) electricity forecast horizon trajectories k=3

Figure 11: electricity forecast horizon trajectories for various covariates k and correlation values (PCC)