Appendix D

Paper D - Comparing Recurrent Neural Networks using Principal Component Analysis for Electrical Load Predictions

Nils Jakob Johannesen and Mohan Kolhe and Morten Goodwin

Faculty of Engineering and Science, University of Agder, PO Box 422, NO 4604 Kristiansand, Norway.

Abstract **- Electrical demand forecasting is essential for power generation capacity planning and integrating environment-friendly energy sources. In addition, load predictions will help in developing demand-side management in coordination with renewable power generation. Meteorological conditions influence urban area load pattern; therefore, it is vital to include weather parameters for load predictions. Machine Learning algorithms can effectively be used for electrical load predictions considering impact of external parameters. This paper explores and compares the basic Recurrent Neural Networks (RNN); Simple Recurrent Neural Networks (Vanilla RNN), Gated Recurrent Units (GRU), and Long Short-Term Memory networks (LSTM). Vanilla RNNs are fully connected neural networks where the output from the previous time step is being fed to the next time step. GRUs are networks with a gating mechanism: a forget gate. LSTM networks also, in addition to a forget gate, include an output gate. Even though the recurrent structure in itself is robust for efficient forecasting, pre-processing of data (including load, weather) is important to enhance the performance. Principal Component Analysis (PCA) reduces and extracts the main components of available data. This work shows that PCA improves the performance of RNNs with use of weather parameters. The historical electrical load dataset from Sydney region is used to test the load forecasting using these techniques considering meteorological parameters. Through load forecasting, it is observed that for the 30 minutes predictions,**

GRU trained with a reduced number of principal components performs best for a typical period with a mean absolute percentage error (MAPE) of 0.74%.

Keywords **-** *load forecasting, principal component analysis, smart grid, load time series, recurrent neural networks*

D.1 Introduction

The smart electrical energy network grid requires more accurate demand and prediction for control and managing the demand in coordination with intermittent renewable energy sources [1]. The smart grid will require advanced control and management, including reliable forecasting to anticipate the events involved in dispatching, control and management of the operating grid. The accurate load prediction can help in managing peak demand and to reduce overall capital cost investment [2]. Demand prediction is important for short term load forecasting. The aim of demand response in the long term is to reduce overall plant and capital cost investments and to postpone the need for network upgrades. For effectively implement demand response programs, short-term load forecasting will provide useful information [3].

Time series analysis has traditionally been performed in meteorology, energy and economics [4]. The Box Jenkins method for time series analysis has been further developed by the research community to a robust parsimonious Autoregressive Moving Average (ARMA) for multivariate forecasting, requiring less human intervention [5]. By observing changes in economic and weather related variables in a Box-Jenkins time series model, refined forecasts are obtained [6]. The AutoRegressive Integrated Mean Average (ARIMA) model was introduced to deal with trends in the dataset. For the multivariate case the exogenous variable is introduced in AutoRegressive Integrated Moving Average with Exogenous variables (ARIMAX). This is further developed into Seasonal AutoRegressive Moving Average with Exogenous variables (SARIMAX), that also accounts for seasonal behaviour [7]. These methods are useful for the modeling of time series and aids the electrical load analysis. Cycles, trends and periodicity can be found through tests provided by time series analysis [11].

Stack Generalization functions on the principle that two minds work better than one. When Geoffrey Hinton first introduced 'Deep Learning' in 2006 composing artificial neurons in stacked layers [9]. The stacking layers of neurons showed that Deep Learning is possible, with the aid of computer power and big amounts of data [10].

State of the art research in electrical load demand forecasting focuses on three main aspects in order to make sound predictions. These inputs are from weather parameters, holidays and time of day. The mentioned relations has been found equally important both for simpler instance based machine learning models to the more complex black box neural networks [11] [36]. And the results of this are provided in the research for short term [10] [37] [38] , mid-term [39], as well as long-term forecasting [40]. The impact of

external weather parameters has proven also to be important for forecasting on limited data, such as for households and buildings [41], as well as cabin areas [35]. Hybrid forecast combining neural networks with autoregression has proven to aid in tracing the curvature of the peak in the volatile electricity markets [5].

In short-term electric load demand forecasting, Recurrent Neural Networks (RNN) by Levenberg-Marquardt and Bayesian regularization on 30 minutes predictions had achieved a mean absolute percentage error (MAPE) of an average in one week 1.4792 [44]. One hour ahead prediction, has been performed on hourly power consumption in Toronto Canada using Long Short-Term Memory (LSTM), achieving a MAPE of 2.639, which was an improvement of the Vanilla RNN of 3.712 MAPE [45]. The Resnetplus model for the ISO-NE dataset proposed a day-ahead load forecasting model based on deep residual networks. A basic structure of several fully connected layers to produce preliminary forecasts of 24 hours. A forecast is then made on the residuals of the preliminary forecast provided with a formulation of Monte Carlo dropout for probabilistic forecasting, achieving an average MAPE of 1.447 [46]. Gated Recurrent Unit (GRU) was used to predict the electricity market in Singapore. Multi-features input models of different time structural architecture named Multi-GRU has been used to give 30 minutes predictions [47].

This article is organised in the following sections: Section D.2 the principal components analysis is explained, the Section E.4 outlines the methodology, Section D.4 includes the data pre-processing, results are discussed in Section E.5, and finally the Conclusion is provided in Section E.6.

D.1.1 Scaling data, normalising

Data is scaled. The general method of calculation is to determine the distribution mean and standard deviation for each feature. Next we subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation.

$$x'_{ij} = \frac{x_i - \hat{x_j}}{\sigma_j} \tag{D.1}$$

x'_{ij} is the value of the input variable of row i and column j, $\hat{x_j}$ is the mean of the values in column j, and finally σ_j is the standard deviation of the values in column j [24].

D.2 Principal Component Analysis for Electrical Load Forecasting

Principal components analysis (PCA) is a multivariate technique that can be applied to many fields for feature reduction. It is the number of samples in the features that are reduced, not the entirety of a feature in itself.

PCA has been found useful in many areas such as daily urban demand forecasting [97].

PCA is extracting the important information for later to represent it in a new set of orthogonal vector input constituting the principal components. These principal components is linear transformation of the data so that the first coordinate explains the most of the variation, the second coordinate the second most, and so on. The components are found through the eigen-decomposition and Singular Value Decomposition [98] [99].
In this work, Sydney region load profile data set is used, which includes meteorological parameters (e.g. DryBulb and WetBulb Temperature, Humidity, weekday and time of use) [106]. In the further feature engineering, a lower indicator variable is designed to differentiate over working-days / non-working days with a binary switch [29]. The RNNs purposefully search in a higher category space to find meaningful relations between the vectors, and therefore the time input is coded using circular coding. The circular coding identifies the time of day according to the unit circle, giving both a sine and cosine coordination as its parameters. They are used as training inputs for the target vector, the electric load demand. The data pre-prossesing in this case leaves the entire feature space with 9 principal components.

Fig. D.1, depicts the proportion of variance that are captured by each number of principal components after feature engineering for the Sydney Data. The red dashed line signifies that when we include the 6 principal components the PCA-process capture 95 % of the variance.

To perform PCA the the input matrix is transposed and crossed with its non-transposed version, stored in matrix L. By diagonalising L, find a matrix M and diagonal matrix W:

$$L = M^T W M \tag{D.2}$$

The feature space is reduced by restricting inputs based on the number of columns that sums up M to make a rotated matrix. The eigenvalues from W are related to the variance of the principal components. PCA reduces the input feature space, yet remains to capture and keep the variation for future inputs and is a important step in the feature engineering.

The proportion of variance needed for optimal feature space may vary. The reference [97] refers to a meta-heuristic practice of principal components explaining 85% of the variance, yet their optimal value was found at 92%.

D.3 Method

The traditional deep neural networks learn patterns on the assumption that inputs and outputs are independent of each other. A RNN depend on the prior elements within the sequence, to perform its decision making. The RNNs used in this work are all based on Keras [30]. RNNs was first developed in natural language processing and the Vanilla RNN is a fully-connected RNN where the output from previous time step is to be fed to next time step by an additional set of units. These units provide for limited recurrence, hence the name 'simple'. The units have also proven to be successful in other time

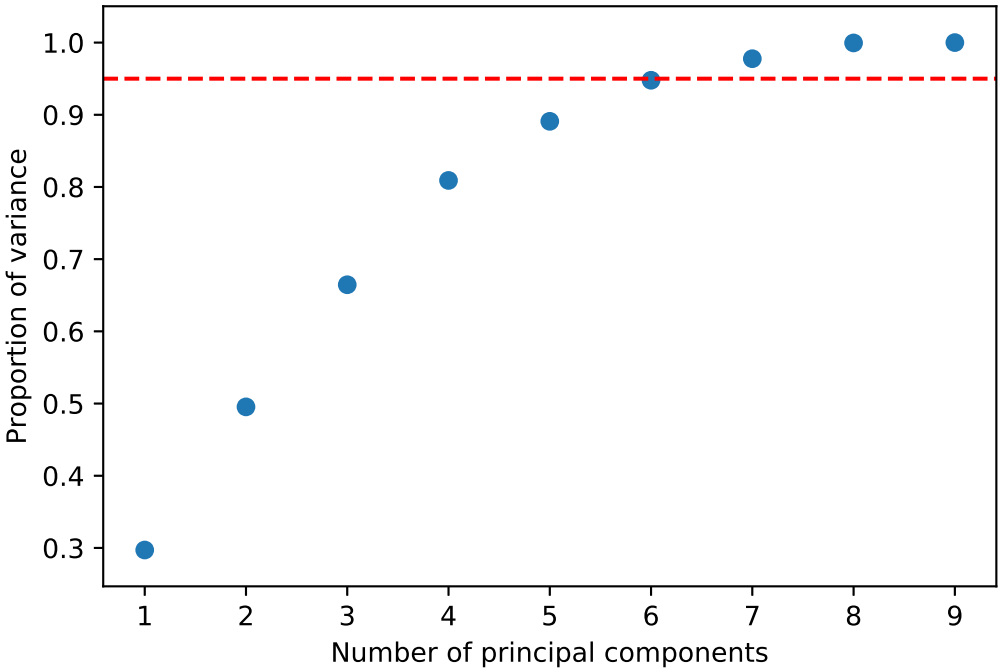

Figure D.1: The cumulative variance per introduced principal components, with red dashed line indicating 95% variance.

series application, and for all problems constituted by sequences, such as electrical load demand. To find the intrinsic nature of linguistic representation Principal component analysis (PCA) has been performed on the hidden unit activation patterns to reveal that the network solves the task by developing complex distributed representations which encode the relevant time relations and hierarchical constituent structure [96]. Vanilla RNN's are fully-connected neural network where the output from previous time step is being fed to the next time step. GRU's are networks with a gating mechanism, a forget gate. Long short-term memory networks also, in addition to a forget gate includes an output gate.

In a recurrent network, in addition the weight layer is combined with the previous state, called the recurrent weight layer U [32]:

$$net_j(t) = \sum_i^n x_i(t) + w_{ji} + \sum_h^m x_h(t-1)u_{jh} + \theta_j \tag{D.3}$$

The set of weights in net_j^t is a candidate value, and through learning finds a candidate solution, $\hat{h}_t^j$, that combines the present state with the previous state. The Vanilla RNN remembers the near future quite well due to the introduction of the hidden state, h, in practice they seem to forget quickly. In LSTM network a memory state is introduced alongside the hidden states, to evaluate long term state dependencies. As illustrated in Fig. D.2, at the bottom the input comes in together with the hidden state (as explained by Vanilla RNN), at the bottom left forget gate f_n:

$$f_n = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{D.4}$$

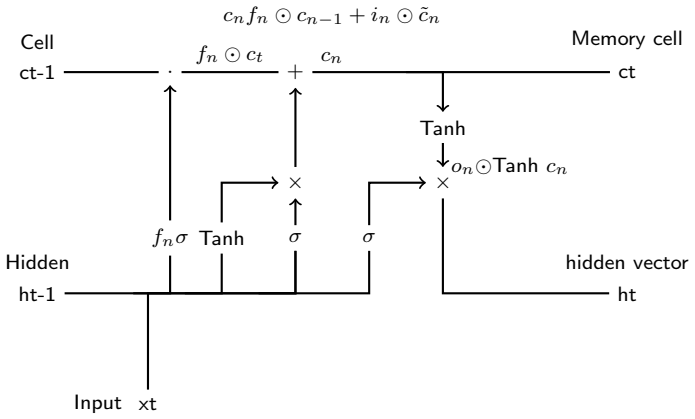

Figure D.2: LSTM network with hidden state and memory cell

and input gate:

$$i_n = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{D.5}$$

In the top memory timeline of Fig. D.2 is a memory cell or cell state, c, the new memory cell is concatenated with previous cell state, added to the input concatenated with cell $\tilde{c}_n$:

$$c_n = f_n \odot c_{n-1} + i_n \odot \tilde{c}_n \tag{D.6}$$

The c_n is updated by forgetting memory as well as adding new memory content $\tilde{c}_n$. For each LSTM unit there exists a memory attached to it c_n at time t. The activation, of the LSTM unit is:

$$h_t = o_n * \tanh\left(c_n\right) \tag{D.7}$$

Where the output gate o_t is computed as:

$$o_t = \sigma\left(W_o \cdot [U_o h_{t-1}, x_t] + V_o c_t\right) \tag{D.8}$$

GRU has only two gates, reset gate r, and update gate z. The first determines the relation of new input to previous memory, and the latter defines to what degree of previous memory is kept. The reset gate is directly applied to the hidden state:

$$r = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \tag{D.9}$$

$$z = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \tag{D.10}$$

When r=1 and z=0, it equals the Vanilla RNN [33].

D.4 Load data pre-processing with Time organisation and training, validation and testing

Time dependent structures are composed as vectors and fed as inputs to the RNNs. To avoid biases and overfitting the data is to be divided amongst training, validation and testing. In particular the algorithm must capture trends and seasonal variations. If the

time series can claim to be stationary, no means needs to be taken. To prove stationarity a search for no trend, constant variance and constant autocorrelation is conducted. Testing for stationarity is done by introducing the null hypothesis H_o: Time series is non-stationary due to trend. By the Augmented Dickey-Fuller (ADF) test, if certain criteria are met the null hypothesis is rejected and the time series is assumed to be stationary. The ADF basically searches for trends in the dataset by evaluating mean and variance over time. Based on this assumption that the time series is stationary, a division into training, validation and test set are done (Fig. D.3).

The training set ranges from the beginning of the recorded data on 01.01.2006 until 31.12.2008. The entire 2009 is used for validation and finally 2010 is for testing. The RNN is learned through a time-lag vector, also known as lookback, that for the multivariate case is a 3D-vector, containing he amount of data (samples), lags and number of inputs (features). Equally on the output, it aims for the target vector. In the training phase this is the next step ahead relative to the input vector.

The proposed model in this work finds suitable training, validation and test-sets by searching for stationarity through Augmented Dickey Fuller Test. The original training set is then reduced feature space and variation representation by performing its principal components analysis, reducing the principal components from an offset features of 9 to be represented by 8 principal components according for 99% of the variance. The training set has then been scaled, and trained on three different RNNs, Vanilla RNN, GRU, and LSTM. These different models have been tested for different seasons to analyse how they assimilate for seasonal variations. Finally the models using PCA, are compared to a version that does not reduce its feature space through PCA.

It is observed from training the RNNs with PCA that during 50 epochs of training and validation, the training loss and validation loss decreases to a point of stability with a minimal gap between the two final loss values, in the Fig. D.5 illustrated with the GRU with PCA, for the Vanilla RNN and LSTM the loss curves show the same convergence.

The RNNs have been tested for a week in January, April, July and October, respectively, and MAPE has been averaged. The results show that all of the RNNs are capturing the inherent structure of the electric load demand quite well, resulting in an acceptable MAPE around 1-2% through all seasons, see Table D.1.

D.5 Results and Discussion

In the winter season the correlations to weather parameters are higher than other seasons, as well as in general the winter season has a higher load demand. These are factors explaining the lower MAPE in winter season as opposed to other seasons.

In the case of GRU networks, the results for all the seasons are improved through PCA

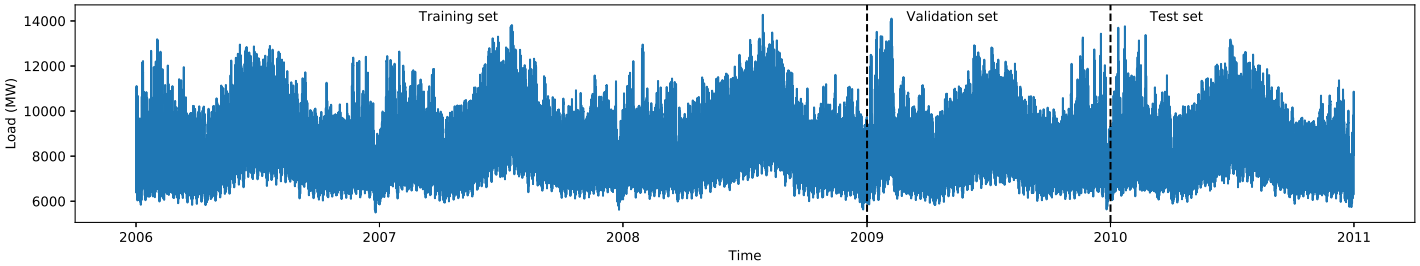

Figure D.3: The Sydney Region data with load measurements for every 30 minutes from 2006 to 2010. Dashed black lines indicates the separation into train-, validation- and test-set.

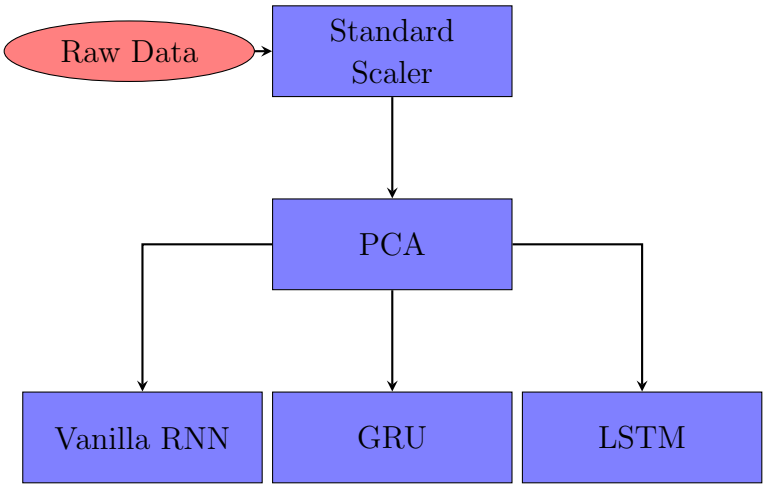

Figure D.4: The Model applied scales the raw Sydney Data, and through PCA predicted by different RNNs

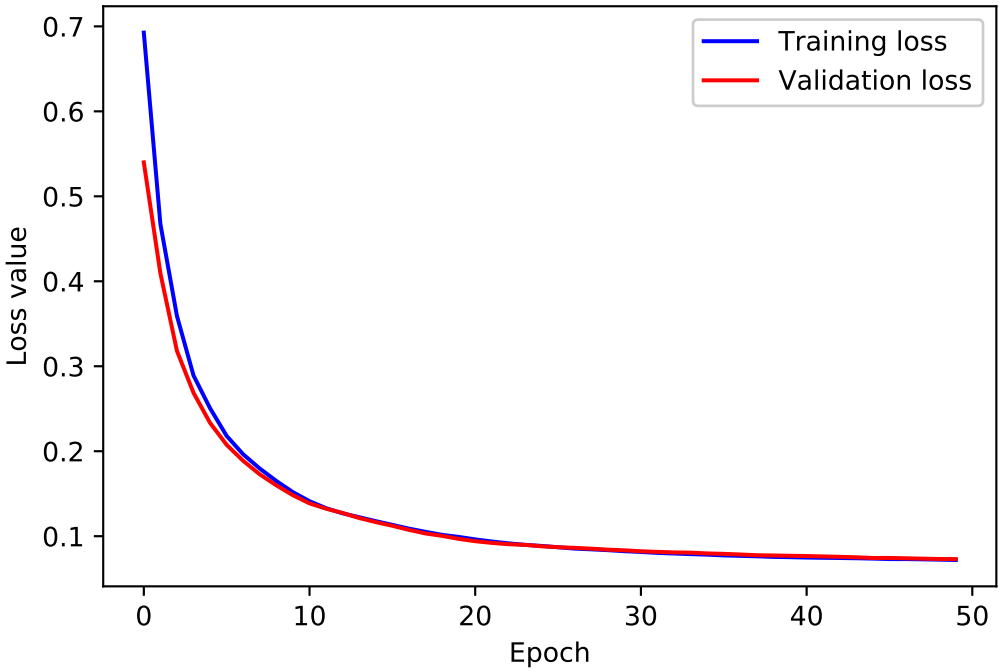

Figure D.5: GRU networks training and validation loss decreases to a point of stability

concluding with 99% of the variation captured by the 8 principal components, see Table D.2. Also for the Vanilla RNN there is a benefit from reduced number of principal components in a lesser MAPE, and for the summer test on a week in July (Fig. D.8), it scores best of all RNNs. Yet for the LSTM it does not benefit from an improved MAPE from the PCA. The best results are measured in January when also the electrical load demand is at the highest (Fig. D.6), and the impact of external weather parameters is influencing greatly on the load demand. The curvature of the load profile is dominated by a high peak at noon, and GRU captures this very good.

The results from the week of April (Fig. D.7), has a lower load demand than January. In January the load demand is highly correlated to the weather parameters readings in winter season. In April, as in January, GRU with PCA achieves the best forecast MAPE result for the week in April, yet with a slightly higher MAPE than for January. This can be explained by the lower load demand in April, and that correlations to weather parameters are usually lower in spring and autumn. In the test week of October (Fig. D.9), which has the same range in load demand (6000 - 10000 MW), it is also GRU with PCA that scores best with a MAPE of 0.94, see Table D.2.

When comparing the results in Tables D.1 with D.2, the MAPE is in the same range for Vanilla RNN (1.45 for April, and 1.38 for October), GRU (1.21 for April and 1.26 for October) and LSTM (1.25 for April, and 1.24 for October). The similarity in results from spring (observed from the test results for the week in April) and autumn (observed from the test results for the week in October) can be explained by similar load range and

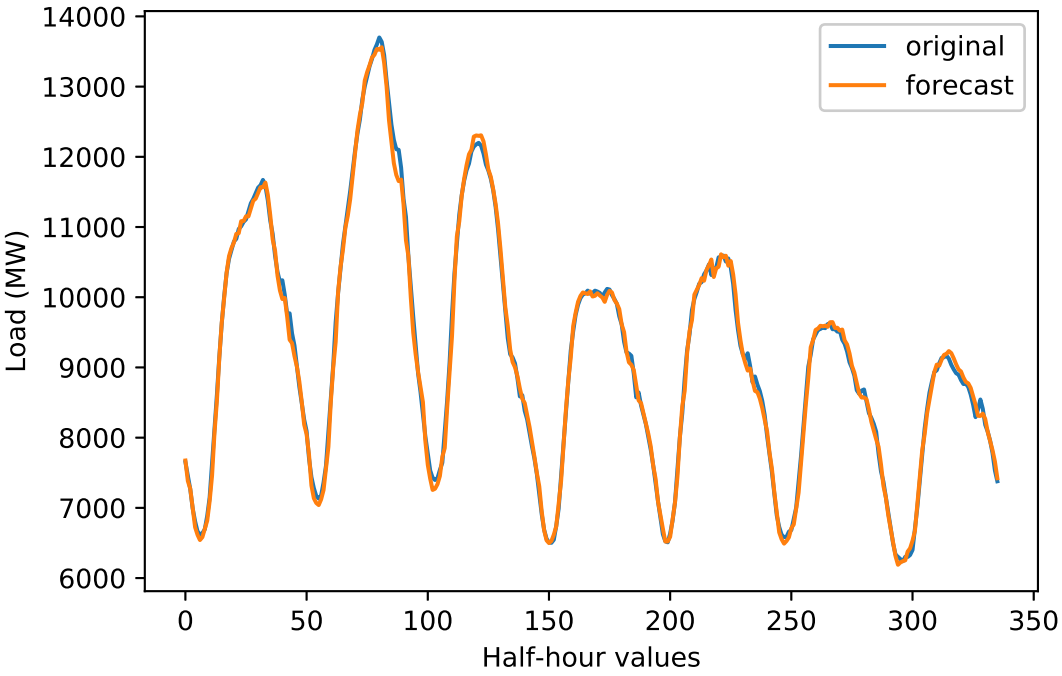

Figure D.6: GRU with PCA tested on a week in January, with a MAPE of 0.74

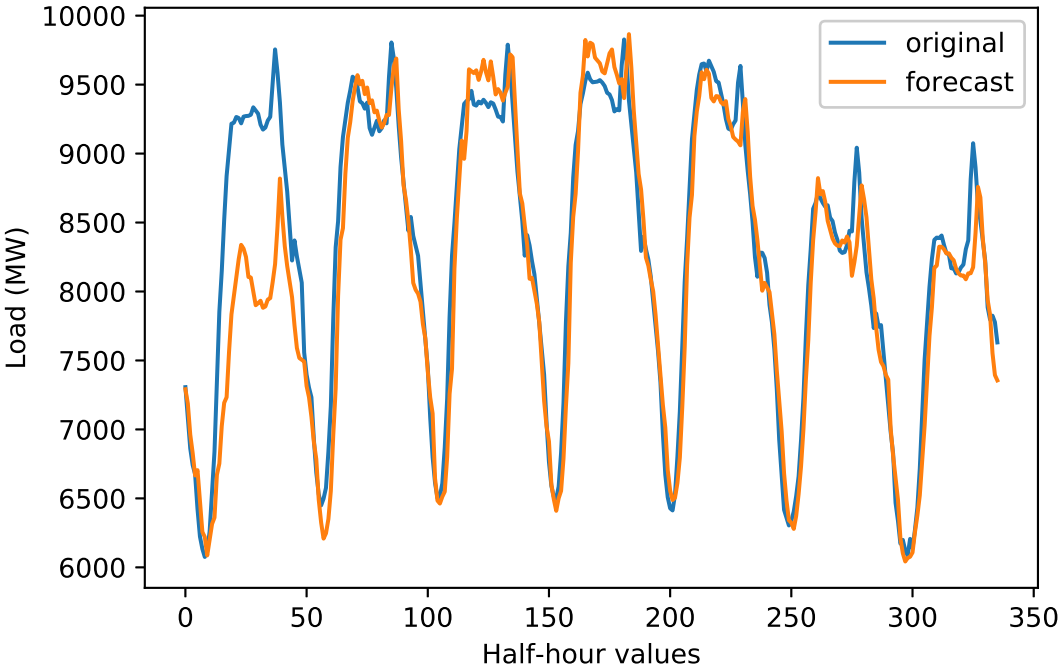

Figure D.7: GRU with PCA performing best of the RNNs for the test week in April

meteorological conditions. In the case of Vanilla RNN and GRU, the explanations of the compared results indicates the same when investigating the results on the RNNs tested with PCA. The exception is the LSTM tested with PCA, that shows a higher MAPE. It is

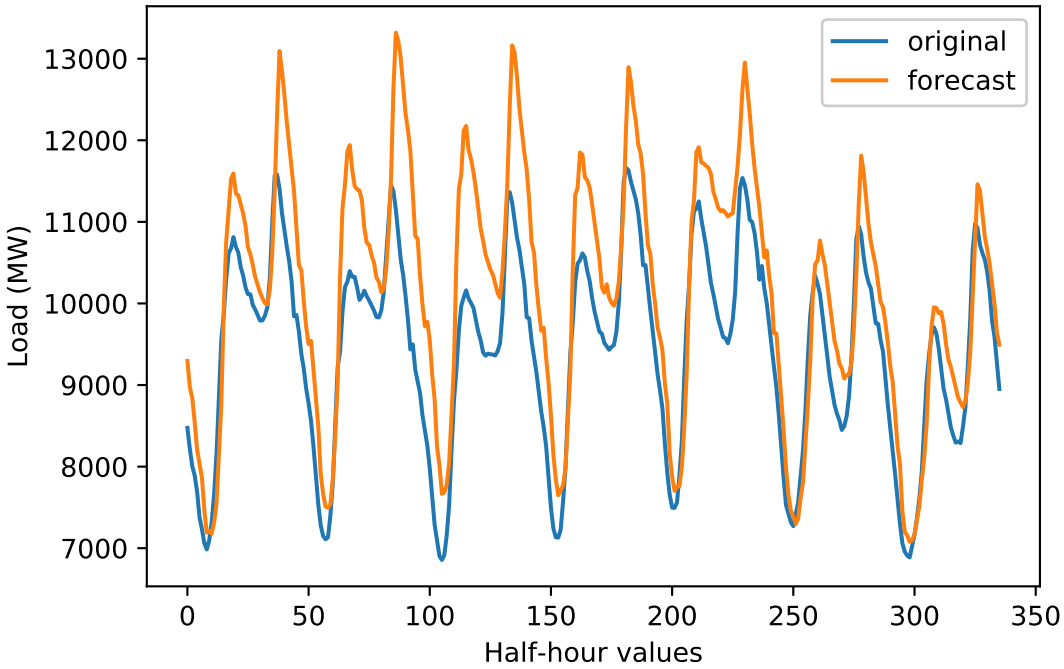

Figure D.8: Vanilla RNN is performing best of all the RNNs on the test week with the lowest load demand, July.

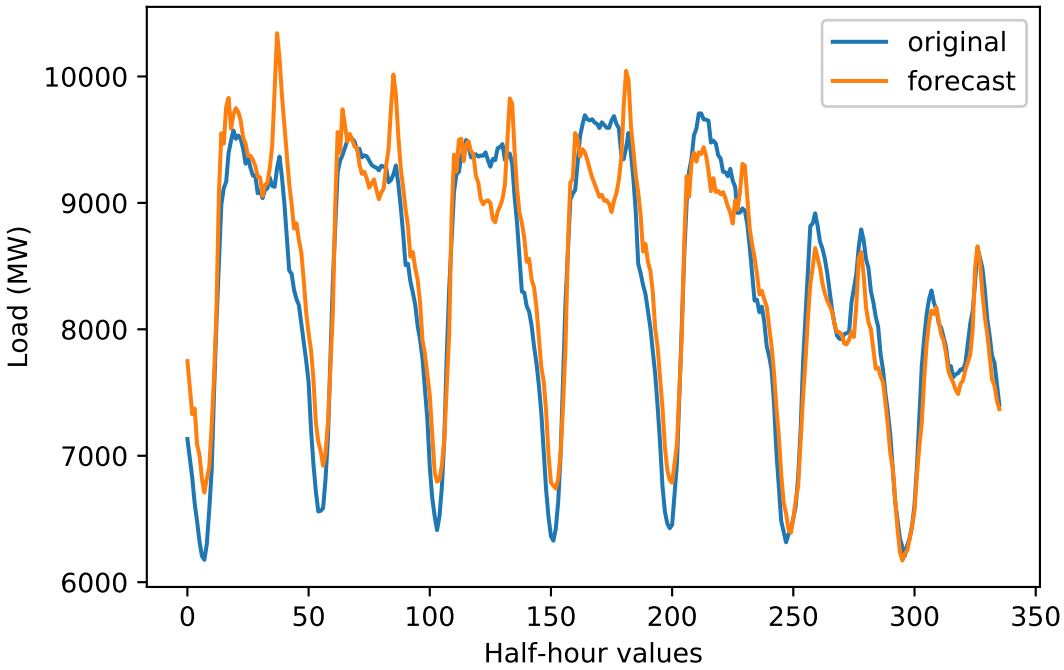

Figure D.9: GRU with PCA performing best of the RNNs for the test week in October

observed that LSTM is a more complex algorithm, than the Vanilla RNN and GRU, and when it is trained with relatively lesser data, although it is analysed using its principal components, it is not able to improve the predictions. It is observed that for the for the

Table D.1: Performance (MAPE)

MAPE –	**Recurrent Neural Networks**		
	Vanilla RNN	***GRU***	***LSTM***
Jan	0.95	0.87	0.90
April	1.45	1.21	1.25
July	1.84	1.64	1.30
October	1.38	1.26	1.24

Table D.2: Performance using PCA (MAPE)

MAPE PCA	**Recurrent Neural Networks**		
	Vanilla RNN	***GRU***	***LSTM***
Jan	0.87	0.74	0.89
April	1.11	1.16	1.60
July	1.39	1.53	1.75
October	1.06	0.94	1.27

week in July with the lowest load demand the simplest RNN (Vanilla RNN) with reduced principal components achieves the preferred MAPE, amongst all of the predictors.

D.6 Conclusion

This paper explores and compares the load prediction analysis through basic RNN; Vanilla RNN, GRU, and LSTM, using PCA. The winter season load behaviour is more influenced by weather parameters, which explains why in the winter season the RNNs scores relatively higher than in other seasons. It is found that PCA can be used to reduce the number of principal components for Vanilla RNN, GRU and LSTM networks. Not only is the reduced feature input space the preferred option in terms of dimensionality reduction, yet also the predictive output is improved. For the electric load demand forecasting the preferred RNN is GRU trained with a principal component of 8, and it is shown through MAPE. After comparing with the version without PCA, the results show that MAPE is reduced when using PCA. For the 30 minutes forecasting GRU with PCA performs best MAPE of 0.74%. This work will benefit the reliable forecasting to anticipate the events involved in dispatching, control and management of the operating grid.

Bibliography

[1] Pierluigi Siano. Demand response and smart grids—a survey. *Renewable and Sustainable Energy Reviews*, 30:461–478, 2014.

[2] Stephen Haben, Siddharth Arora, Georgios Giasemidis, Marcus Voss, and Danica Vukadinovic Greetham. Review of low-voltage load forecasting: Methods, applications, and recommendations. *arXiv preprint arXiv:2106.00006*, 2021.

[3] Xavier Serrano-Guerrero, Marco Briceño-León, Jean-Michel Clairand, and Guillermo Escrivá-Escrivá. A new interval prediction methodology for short-term electric load forecasting based on pattern recognition. *Applied Energy*, 297:117173, 2021.

[4] George.E.P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

[5] S. Vemuri, W. L. Huang, and D. J. Nelson. On-line algorithms for forecasting hourly loads of an electric utility. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(8):3775–3784, 1981.

[6] Noel D. Uri. Forecasting peak system load using a combined time series and econometric model. *Applied Energy*, 4(3):219 – 227, 1978.

[7] S. Rahman and R. Bhatnagar. An expert system based algorithm for short term load forecast. *IEEE Transactions on Power Systems*, 3(2):392–399, 1988.

[8] Nils Jakob Johannesen and Mohan Lal Kolhe. Application of regression tools for load prediction in distributed network for flexible analysis. In *Flexibility in Electric Power Distribution Networks*. CRC Press, 2021.

[9] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[10] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition, 2017.

[11] Nils Jakob Johannesen, Mohan Kolhe, and Morten Goodwin. Relative evaluation of regression tools for urban area electrical energy demand forecasting. *Journal of Cleaner Production*, 218:555–564, 2019.

[12] A. Songpu. External parameters contribution in domestic load forecasting using neural network. *IET Conference Proceedings*, pages 6 .–6 .(1), January 2015.

[13] Ernesto Aguilar Madrid and Nuno Antonio. Short-term electricity load forecasting with machine learning. *Information*, 12(2), 2021.

[14] Abderrezak Laouafi, Mourad Mordjaoui, Salim Haddad, Taqiy Eddine Boukelia, and Abderahmane Ganouche. Online electricity demand forecasting based on an effective forecast combination methodology. *Electric Power Systems Research*, 148:35–47, 2017.

[15] S. Mirasgedis, Y. Sarafidis, E. Georgopoulou, D.P. Lalas, M. Moschovits, F. Karagiannis, and D. Papakonstantinou. Models for mid-term electricity demand forecasting incorporating weather influences. *Energy*, 31(2):208–227, 2006.

[16] Yongxiu He, Jie Jiao, Qian Chen, Sifan Ge, Yan Chang, and Yang Xu. Urban long term electricity demand forecast method based on system dynamics of the new economic normal: The case of tianjin. *Energy*, 133:9–22, 2017.

[17] Peter Lusis, Kaveh Rajab Khalilpour, Lachlan Andrew, and Ariel Liebman. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Applied Energy*, 205:654–669, 2017.

[18] Nils Jakob Johannesen, Mohan Lal Kolhe, and Morten Goodwin. Load demand analysis of nordic rural area with holiday resorts for network capacity planning. In *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–7, 2019.

[19] Nils Jakob Johannesen, Mohan Kolhe, and Morten Goodwin. Deregulated electric energy price forecasting in nordpool market using regression techniques. In *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, pages 1932–1938, 2019.

[20] Muhammad Amri Yahya, Sasongko Pramono Hadi, and Lesnanto Multa Putranto. Short-term electric load forecasting using recurrent neural network (study case of load forecasting in central java and special region of yogyakarta). In *2018 4th International Conference on Science and Technology (ICST)*, pages 1–6, 2018.

[21] Lei Zhang, Linghui Yang, Chengyu Gu, and Da Li. Lstm-based short-term electrical load forecasting and anomaly correction. In *E3S Web of Conferences*, volume 182, page 01004. EDP Sciences, 2020.

[22] Kunjin Chen, Kunlong Chen, Qin Wang, Ziyu He, Jun Hu, and Jinliang He. Short-term load forecasting with deep residual networks. *IEEE Transactions on Smart Grid*, 10(4):3943–3952, 2019.

[23] Weixian Li, Thillainathan Logenthiran, and Wai Lok Woo. Multi-gru prediction system for electricity generation's planning and operation. *IET Generation, Transmission & Distribution*, 13(9):1630–1637, 2019.

[24] Xavier Serrano-Guerrero, Guillermo Escrivá-Escrivá, and Carlos Roldán-Blay. Statistical methodology to assess changes in the electrical consumption profile of buildings. *Energy and Buildings*, 164:99–108, 2018.

[25] Baigang Du, Qiliang Zhou, Jun Guo, Shunsheng Guo, and Lei Wang. Deep learning with long short-term memory neural networks combining wavelet transform and principal component analysis for daily urban water demand forecasting. *Expert Systems with Applications*, 171:114571, 2021.

[26] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[27] Hervé Abdi and Lynne J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010.

[28] Vasudev Dehalwar, Akhtar Kalam, Mohan Lal Kolhe, and Aladin Zayegh. Electricity load forecasting for urban area using weather forecast information. In *2016 IEEE International Conference on Power and Renewable Energy (ICPRE)*, pages 355–359, 2016.

[29] Nils Jakob Johannesen, Mohan Lal Kolhe, and Morten Goodwin. Smart load prediction analysis for distributed power network of holiday cabins in norwegian rural area. *Journal of Cleaner Production*, 266:121423, 2020.

[30] Francois Chollet et al. Keras, 2015.

[31] Jeffrey L Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2):195–225, 1991.

[32] Mikael Boden. A guide to recurrent neural networks and backpropagation. *the Dallas project*, 2002.

[33] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.