

Forecasting IRAN's max daily demand for electricity in different weather types using deep learning approach

Ghahreman Abdoli (Corresponding Author)

Professor at University of Tehran, Faculty of Economics
Email address: abdoli@ut.ac.ir

Mohsen MehrAra

Professor at University of Tehran, Faculty of Economics
Email address: mmehrara@ut.ac.ir

Mohammad Ebrahim Ardalani

Ph.D. Candidate at University of Tehran, Alborz Campus
Email address: ardalani@ut.ac.ir

Abstract - Electricity network management is a complicated issue that needs a forecast of consumption. This predication is generally hard and a challenging topic due to nonlinearity and permanent fluctuation of load in the network. Although linearities of time-series load may give an estimation, linearization of nonlinear data to foresee removes the nature of information which shows anomalies in electricity demand. New approaches of (ANN) help to make a forecast model with keeping information properties. In this paper, we categorized max daily load data in IRAN based on climate zone to figure the following 2 months. Long Short-Term Memory (LSTM) from deep learning approach picks and yields contrasted with different steps of epochs. The outcomes appear epochs do not necessarily improve the accuracy of prediction and best MADP for cold, tropical and temperament zone, are 1.26%, 2.5%, and 10.49%, respectively.

Keywords: Prediction Model, Electricity consumption Demand LSTM, Forecast Accuracy, epochs.

1. Introduction

Power utilization in developing countries has become quickly in recent decades. As the nature and size of the economy of these nations develop quickly, their requests increment extensively and the vitality the board system is getting progressively significant. To manufacture an effective system, it is unavoidable to figure vitality requests as per different circumstances. Estimation of consumption in a focused power showcase is fundamental for the market in arranging their tasks and dealing with their ability and upkeep, just as making monetary enhancement in the power business. In this way, estimating the development of intensity request assumes a significant job in the manageability of market interest. Even though, finding a fine estimation model that is exact and precise for power load anticipating is one of the most challenging. It is a direct result of the highly unstable, non-stationary and non-linear trend of electricity demand and other ecological variables. The long term prediction is not feasible because of unpredictable increment and vulnerability in the utilization of intensity with the developing populace and reliance on electric power. Making a strong forecasting model for developing countries given by unmodified data is a challenging issue. Seasonal ARIMA solution is more than once utilized in power utilization investigation and is a high-accuracy approach for regular information forecast. (Wang et al., 2012) On the other hand, since the behavior of the demand for electricity is with uncertainty and nonlinearity, neural networks can be fit for exploring the behavior of consumption for prediction purposes. (Kumar et al., 2018) . Cutting-edge approaches in data analysis permit increasing better estimate models. With the emerge of neural networks, With the development of neural systems, analysts went for surveying its precision of expectation versus linear models, like ARIMA.

In this paper, we compare LSTM as one of the most current strategies for the expectation for time-series in 3 distinctive climate classifications of electricity consumption in Iran. Since power consumption in Iran is more related to the number endorsers and temperature than some other variable, forecasting power consumption by climatic regions can report noteworthy outcomes. (Ardalani and Ghanbari, 2012; Ardalani et al., 2012). Iran's atmosphere can be isolated into three (cool, warm and moderate) zones. The model learns from historical data patterns. Data is built and optimized on a single historic dataset from TAVANIR. The forecast results are contrasted with real consumption values to measure the performance of the methods. Then the result compares with the ARIMA solution which is trained and calculated in the same data, program, and hardware. The following segment of this paper dispenses on related work. At that point, the philosophy of LSTM (in

examination with RNN) will clarify. From that point onward, our experimental work and spotlight on our outcomes will be displayed, and the last part goes to the conclusion.

2. Related Work

Here is some related research to LSTM for power demand estimating.

(J.-Y. Kim and Cho, 2019) proposed a technique dependent on deep learning that comprises of a projector for characterizing a proper state for a given circumstance and an indicator that conjectures vitality request from the characterized state. Results gave better execution (MSE = 0.384) than the ordinary models. (Zheng et al., 2019) declared provided that the prediction of household-level electricity demand, first was built at the appliance-level and afterward aggregated to get the residential-level, forecast performance could be significantly improved by LSTM. (Kandanand, 2019) found ARX (Autoregressive with Exogenous Output) outperformed ARIMA in the accuracy of electricity demand prediction. (Weinberg, 2019) utilized Multi Seasonal ARIMA models for long and short-range estimates of load requests and arrived at 15.1% and 21.6% MAPE volatile and nonvolatile settings, respectively. (Yu et al., 2019) offered a crossover of ARIMA and ANN could give a better expectation for load requests in time categories of consumption. (Aradhye et al., 2019) found a hybrid model of ARIMA and SVM outperformed ARIMA, SVM, and ANN individually in terms of MSE and MAE for load prediction. (Bedi and Toshniwal, 2019) presented a deep learning-based framework for forecasting electricity consumption, and contrasted the outcomes and ANN, RNN and SVR models. (Badal and Franzén, 2019) expected to analyze LSTM and SVM. Results shew that the SVM-model had marginally higher precision and a lower standard error of the mean. The conclusion made was that in this specific case, SVM outperformed RNN in prediction accuracy, however, there is an opportunity to get better of the two executions of these strategies. Regarding specificity and sensitivity, the choice of an SVM or RNN would be highly dependent on the implementation of real-world applications. (Muzaffar and Afshari, 2019) compared LSTM prediction accuracy over the horizons of 1, 2, 7 and 30 days with traditional methods by picking up an electrical load data with exogenous variables including temperature, humidity, and wind speed. Results shew, the trained LSTM network is better than other methods and has the potential to further improve the accuracies of forecasts. (Bano et al., 2019) anticipated cost and load of New York city by MLP, SVM and LR. The feature selection was done by CART and RFE. (Khan et al., 2019) represented the lowest mean absolute percentage error with a DCNN model to forecast the electricity demand in comparison with RNN and CNN models. (Mujeeb et al., 2019) with the help of big data of load and price of smart grid, proposed DLSTM for prediction which outperformed ANN, NARX and ELM in terms of MAE and NRMSE. (Wu et al., 2019) used GRU for electricity cost in short time prediction. Based on their results, This solution could make the precision for a forecast of short-time demand, better and gives preferable execution over traditional approaches. (Bissing et al., 2019) found a hybrid model of ARIMA and Holt-Winters gave a better prediction for hourly day-ahead price. (Guo et al., 2019) used ARIMA for the prediction of predictive energy management of HEVs. They believed with the ARIMA; consumption is reduced by around 5%-7% compared with when no predictor used. (Balaji et al., 2019) and (Tokgöz and Ünal, 2018) and (Kumar et al., 2018) generated half/two-hour-ahead electric demand forecast models with CCN, GRU, ELM and LSMT and results were acceptable. (Chang et al., 2018) believed that AdaGrad and RMSProp, made LSTM perform better based on their results. (Agrawal et al., 2018) used LSTM-RNN for their prediction which gave 2.25% in confidence interval and 6.54 for MAPE. (Jiang and Hu, 2018) used LSTM to forecast the day-ahead electricity price. In addition to historical prices, they considered holidays, an hour of the day, day of the week, climate change, oil as the price of substitutions, the trend of consumption, etc. The result said that the LSTM network outperformed four forecasting methods in the paper. (Rahman et al., 2018) found the accuracy of forecasting in RNN is not better than MLP for making weekly predictions in commercial and residential buildings in the U.S. (Bouktif et al., 2018) used LSTM with multiple configurations to compare predict models for short to medium term aggregate load forecasting with France electricity consumption data. Results shew, LSTM gave high accuracy. (N. Kim et al., 2018) got prediction accuracy of around 82.5% with LSTM for month-ahead electricity consumption prediction. (T.-Y. Kim and Cho, 2018) claimed CNN-LSTM hybrid networks, LSTM and DNN could separate unpredictable features of electric power utilization. The CNN-LSTM could appraise control utilization, and CNN-LSTM RMSE works better than traditional forecast models for the individual residential demand. (Peng et al., 2018) used DELSTM for an electricity price forecast. The conclusion of the paper shows that the offered model works better than the current predictive models. (Bedi and Toshniwal, 2018) presented an EMD-based deep learning which gets the EMD with LSTM, together. Then for assessing the performance of the given model, they compared results with RNN, LSTM, and EMD-based RNN models. (Kuo and Huang, 2018) found a combined model of 2 deep neural networks, (CNN & LSTM) gave better performance in load prediction than each of them, individually. (Choi and Lee, 2018) developed LSTM for four real-life time series with the aim of electric load forecasting which resulted in acceptably. (Shi et al., 2017) found the power of prediction in household load demand ARIMA, SVR, and deep RNN is 19.55%, 13.1%, 6.5% respectively. (Amarasinghe et al., 2017) presented a load prediction based on deep learning. According to their reports, the CNN outperformed SVR. (Marino et al., 2016) presented two LSTM (standard and Sequence to Sequence) for electricity

consumption forecast and while the first model works well in one-hour data, it failed at one-minute time-series, S2S performed well on both datasets. (Amini et al., 2016) in their research improved the precision of ARIMA by improving the incorporated and auto-backward request parameters. (Voronin and Partanen, 2014) integrated NN, ARIMA and wavelet transforms for price and demand prediction. The results of the paper represented that the proposed model gave an improvement in both request and value forecast precision in comparison with and models utilizing a different systems approach. (Wang et al., 2012) tried to improve the precision of seasonal ARIMA for load consumption. The results indicated that the accuracy of prediction in the three residual modification models outperformed than the single seasonal ARIMA. The integration of models gives the best result. (Shafie-Khah et al., 2011) found a hybrid model of ARIMA (linear patter) and RBFN (nonlinear patter) could provide a considerable improvement for the forecasting accuracy. They also used Particle Swarm Optimization to optimize the network structure. PSO makes the RBFN be adapted to the specified training set, reducing computation complexity and avoiding overfitting. (Kandanand, 2011) compared the power of prediction of ARIMA and ANN and MLR in load demand. The reports said that MAPE of ANN was 0.996% and MAPE of MLR and ARIMA 3.2 and 2.8. (Che and Wang, 2010) with the help of SVRARIMA (a combination of SVR and ARIMA) demonstrated that the model outperformed the NN model, the traditional ARIMA and other hybrid models with the index of RMSR and MAPE.

3. Methodology

1.1. Long Short-Term Memory

It is a kind of deep learning approach with memory and equipped for adapting long haul conditions and consecutive information. A Cell as memory part and "controllers" (made from input, output and Foregate gate) are the regular engineering of this system. LSTM is intended for applications where the info is an arranged succession. It is especially similar to an RNN however with increasingly entangled innards.

The incredible thing about RNN is that it remembers everything. It has long-ago conditions that can catch the whole grouping, but at the same time, that is a drawback! In LSTM we let RNN specifically pick whatever individuals and what it slips its mind. It successfully chooses how a lot of the system is going to convey information from the past time-step and what amount should be utilized in the present time-step.

Like every single neural network, the node plays out an estimation utilizing the sources of info and returns yield esteem. LSTM reuses the yield from a past advance as a contribution for the subsequent stage, while in RNN, the yield is utilized alongside the following component as the contributions for the following stage. An ordinary internal neuron of RNN and LSTM are as underneath:

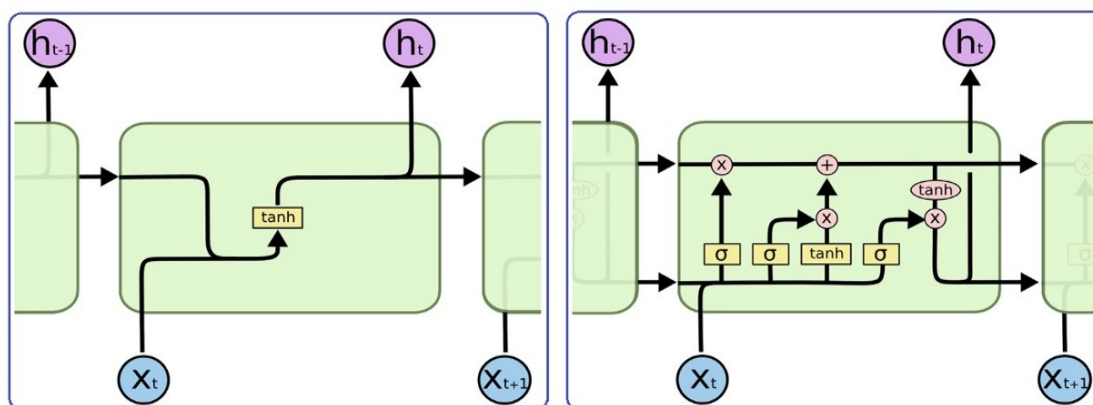


Diagram 1: An inner neuron of an RNN (left) and LSTM (right) (Olah, 2015)

Diagram.1 shows the RNN takes and joins the past shrouded state as input.1 with the present time-step vector (X_t) as input.2. At that point goes the outcome through some non-linearity (\tanh) for squishing values between $[-1\ 1]$ to direct the system. At that point, it delivers another shrouded state (h_t) and goes to time-step (X_{t+1}). By rehashing the procedure, it might make a forecast model toward the finish of layers. (relies upon what we need). LSTM has more segments within. This is based on gates. Gates choose if the information is significant to keep or ought to be overlooked during preparing. Every one of the entryways has sigmoid (σ) enactment to comprise smooth bends in the range 0 to 1 and the model stays differentiable. Information more like 0, ought to be overlooked and close to 1, keeps refreshing cell state. Diagram.1 shows three images of sigma (σ) which are answerable for overlooking, information and yield door, individually. Sigmoid either let data through or prevents the data from experiencing and this choice relies upon the info that it's getting. From the start entryway, it chooses whether to recollect the past state dependent on an element of the past concealed state (h_{t-1}) and the

present vector input (x_t). The condition is as beneath. LSTM compact conditions with compact first exhibited in 1997. (Hochreiter and Schmidhuber, 1997)

$$f_t = \sigma(W_f[h_{t-1}; x_t] + b_f)$$

The other choice LSTM needs to make is the amount it will add to contribute to its representation from the current time step. The conditions for that are as beneath:

$$i_t = \sigma(W_i[h_{t-1}; x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}; x_t] + b_c)$$

This is again a component of both the shrouded state (h_{t-1}) and current input (x_t), in addition to bias ($b_{i\ or\ c}$) and afterward, it goes through a non-linearity. There are two segments to this progression. The first one (i_t) says what amount is it going to contribute and the subsequent one (\tilde{C}_t) which is about what you will contribute. Function (tanh) disseminates slopes. Thus, averts vanishing or exploding. At that point, the subsequent stage goes to an introduction (like RNN) between the present state and the past state.

$$C_t = f_t * C_{t-1} + \tilde{C}_t * i_t$$

f_t is about how much past state will contribute and i_t is about how much the current contribution will add to the new shrouded state description. The following condition is about yield (o_t) in each time step and making a hidden state h_t which is a function of cell C_t , not h_{t-1} .

$$o_t = \sigma(W_o[h_{t-1}; x_t] + b_o)$$

$$h_t = \tanh(C_t) * o_t$$

A significant reality about LSTM is that there is no need to concern about reaching a locally optimal solution. The vanishing or fading gradient is a typical issue happens during ANN utilizing gradient-based learning techniques. This is accomplished by using a fractional derivative of the error function to each parameter in each iteration of the training process, however, gradient values (moving to the highest point of the matrix) continuously become little enough that the weight changes are insignificant.

Therefore, the learning procedure gets slower, and in progressively extreme cases, it stops the learning procedure. In feed-forward neural systems (MPL, CNN, ...) it happens on account of numerous layers and in RNNs, it is a direct result of the steps.

LSTM can learn to set explicit transient reliance. It implies units have memory for a short or a long-lasting period and neurons (inside their pipeline) kept a set of memory to take into consideration handling successive and worldly information. It is finished by the plausibility of passing data from the past cell with no change, rather than iteratively adjustment at each time-step or layer. Loads also are relied upon to meet their ideal qualities. In this manner, by including three entryways (input, yield, overlook) and \tilde{C}_t , this system takes care of the issue of the vanishing of inclination and helps with controlling mistakes during back-spread.

4. Experimental Work

The models in LSTM are controlled by Python and a piece of similar equipment, to have an equivalent circumstance to pass judgment.

1.1.1. Cross-validation and running model

Albeit cross-validation is a way to deal with maintaining a strategic distance from over/under-fitting, information is enough to abstain from overfitting. (Rashid et al., 2018) Now, we isolate the information to evaluate the intensity of expectation in the following part

1.1.2. Data Processing

For any system, we should initially import the libraries and information handling identified with our work. Here we called NumPy, matplotlib, pyplot and panda are the libraries from the start. At that point, we should downsize the information in a similar range. This is called highlighting scaling, and minmax scaler which is the device to be called.

The subsequent stage goes to making an information structure. This is presumably the most significant of making a model, in which we locate the best time steps. It is gained by tedious trials and, 60 was the best outcome for this examination. It implies we need our system to have a memory of 60 characters and one yield. We have to set our information as reliance and independence. By picking 60 as time steps, information Num.1 to information Num.60 makes the principal yield. At that point, information Num.2 to information Num.61, sends out the second yield, etc. From that point forward, we should reshape information dependent on RNN.

1.1.3. Building Network

Presently we have to call Sequential, LSTM, Dense and Dropout from Keras (backend by TensorFlow or PlaidML). In any case, we utilized a consecutive layer and afterward characterized LSTM with 50 neurons in each layer. Here we characterize the shrouded layers of the system which is 2. Our dropout for each layer is 20% and the thick layer is only for our yield in the last layer. Note that, since we only one yield, the unit characterizes one. For aggregating the system, which is the subsequent stage, we utilized "Adam" as a streamlining agent that chips away at non-stationary destinations and issues (Kingma and Ba, 2014) and "Mean_Squared_Error" for our loss.

The subsequent stage is fitting the model for the preparation set. Here we run the ages (number of preparing times) and cluster size. The group size clarifies all outnumber of preparing that information utilized. we won't give all information to NN without a moment's delay and rather than that, NN gets the information in a few phases and littler groups. The quantity of clumps rises to the quantity of emphasis to finish preparing. Since 50 in the cluster size of our model, the quantity of emphasis for the whole informational collection is 44. Age speaks to the all-out number of times; a learning calculation sees the total dataset. "Since deep learning utilizes gradient descent to upgrade model, it makes sense to pass the whole dataset through a solitary system multiple times to update the weights and thus obtaining a better and more accurate prediction model". (Siami-Namini and Namin, 2018) Our calculation is executed in python seeking after Keras (TensorFlow Back-End) and sorted in 3 sections with the subtleties as underneath:

1.1.4. Making the expectations and picturing the outcomes

Here was the last some portion of the model which goes to real and predicted load and envisioning the outcomes.

5. Experimental Results and Discussion

Figure.1 presents the data (train + test) categorizes in three climate zone and an hour of max load in the second row. As it shows the trend of consumption and max load is different in cases.

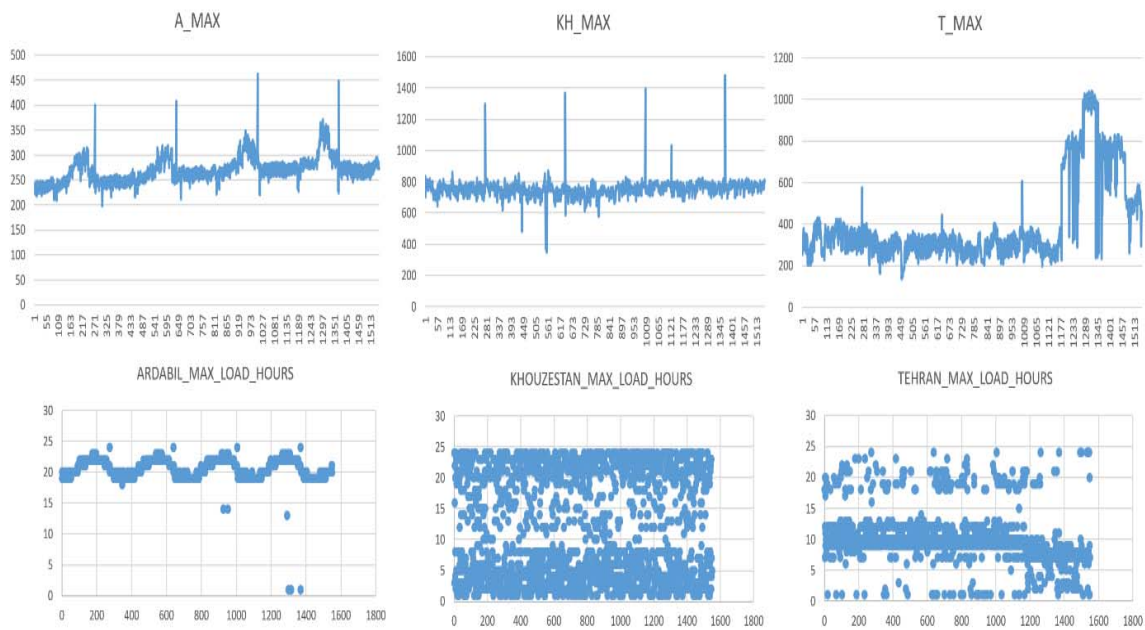


Figure 1: Data and hour of max load in cold (first column), tropical (middle column) and temperate (last column) zone

Total report of our investigations with LSTM is as underneath:

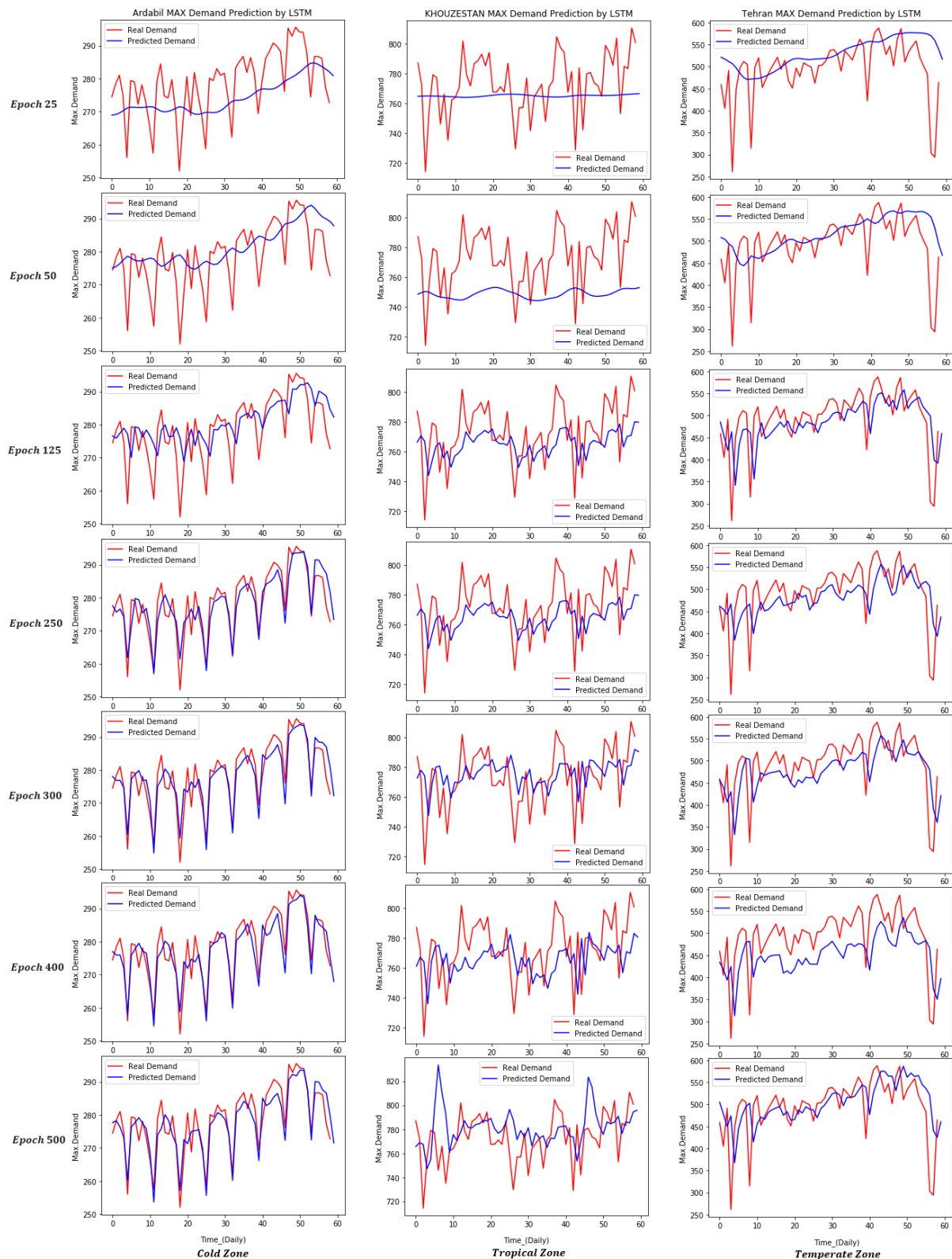


Figure 2: Total report of predictions in cold, tropical and temperate zone

Figure.2 exhibits the pattern of prediction in networks after increasing the number of epochs. Epochs may not be equivalent to the number of cycles and the quantity of best ages to prepare a model isn't clear, however, the trend of estimation get better by increasing epochs in most steps.

The table beneath shows the correlation between LSTM to figure exactness.

Table 1: Forecast Accuracy (ARDABIL)

ARDABIL	MAPE	MSE	MAD	TS
EPOCH.25	2.92%	85.46	8.14	30.47
EPOCH.50	2.35%	74.26	6.38	-27.63
EPOCH.125	2.33%	68.60	6.36	-18.98
EPOCH.250	1.36%	20.90	3.77	3.01
EPOCH.300	1.30%	18.14	3.60	3.36
EPOCH.400	1.26%	17.34	3.49	19.15
EPOCH.500	1.26%	17.55	3.51	13.12

Table 1: Forecast Accuracy (KHOUZESTAN)

KHOUZESTAN	MAPE	MSE	MAD	TS
EPOCH.25	2.25%	464.56	17.43	26.36
EPOCH.50	3.55%	980.89	27.69	51.46
EPOCH.125	2.28%	440.75	17.60	26.34
EPOCH.250	2.57%	523.62	19.88	33.28
EPOCH.300	2.05%	432.82	15.63	-9.56
EPOCH.400	2.50%	536.59	19.31	22.70
EPOCH.500	2.49%	673.93	18.98	-26.35

Table 1: Forecast Accuracy (KHOUZESTAN)

TEHRAN	MAPE	MSE	MAD	TS
EPOCH.25	11.58%	5518.53	45.58	-44.57
EPOCH.50	10.56%	4712.36	41.73	-30.75
EPOCH.125	10.70%	3821.57	45.61	11.35
EPOCH.250	10.78%	3912.66	46.15	16.95
EPOCH.300	10.81%	3983.79	47.05	21.22
EPOCH.400	13.95%	5490.38	64.16	38.67
EPOCH.500	10.49%	4046.14	43.01	-12.31

Formulas of methods in table 1,2,3 are as below:

$$MAPE \text{ (Mean Absolute Percentage Error)} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

$$MSE \text{ (Mean Squared Error)} = \frac{1}{n} \sum_{t=1}^n (A_t - F_t)^2$$

$$MAD \text{ (Mean Absolute Deviation)} = \frac{1}{n} \sum_{t=1}^n |A_t - F_t|$$

$$TF \text{ (Tracking Signal)} = \frac{\sum_{t=1}^n (A_t - F_t)}{MAD}$$

Where A_t and F_t are actual and forecast, respectively.

As the table shows for Ardabil (cold zone), the accuracy of prediction in epochs 400 and 500 is the same while in Khuzestan as tropical zone, the best answer is in epoch 300. Tehran, on the other hand, does not represent a direct relation with increasing epochs. The trend has fluctuated and the best result is in epoch 500 while in the best case, the accuracy of prediction is not like other zones,

6. Improvements

In this paper, "daily demand" information was utilized for forecasts, while utilizing different files in a board of information may improve the presentation of the models. In conclusion, the hybrid models or other complex advancement methods, for example, extrapolation or other metaheuristic calculations could report fascinating outcomes.

7. Conclusion

Developing countries need to have a prediction of demand and supply of electricity due to increasing consumption, especially in a max load of the day. In this paper, we used a daily load of power in IRAN and with filtering the max load in the day, tried to forecast the load with LSTM in deep learning. Since climate change in IRAN has the greatest effect of consumption, prediction categorized in 3 different climate zone and models ran by 6 types of epochs which proved for data in this research, number of epochs did not always get the best answer, and MADP is 1.26%, 2.5% and 10.49% in cold, tropical and temperment zone, which is completely acceptable.

References:

- [1] Wang, Y. and Wang, J. and Zhao, G. and Dong, Y. (2012). Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China. *Energy Policy*, 48, 284-294.
- [2] Kumar, S. and Hussain, L. and Banarjee, S. and Reza, M. (2018). Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster. Paper presented at the 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT).
- [3] Ardalani, M. E. and Ghanbari, A. (2012). The impact of abolishing electricity subsidies in residential, industrial and agriculture in Iran. *The 27th International Power System Conference (PSC2012)*.
- [4] Ardalani, M. E. and Sadeghi, H. and Goli, Y. (2012). The impact of climate change on residential electricity consumption in Iran. *The 27th International Power System Conference (PSC2012)*
- [5] Kim, J.-Y. and Cho, S.-B. (2019). Electric energy consumption prediction by deep learning with state explainable autoencoder. *Energies*, 12(4), 739.
- [6] Zheng, Z. and Chen, H. and Luo, X. (2019). Spatial granularity analysis on electricity consumption prediction using LSTM recurrent neural network. *Energy Procedia*, 158, 2713-2718.
- [7] Kandananond, K. (2019). Electricity demand forecasting in buildings based on ARIMA and ARX models. Paper presented at the Proceedings of the 8th International Conference on Informatics, Environment, Energy, and Applications.
- [8] Weinberg, D. (2019). Electrical power demand forecasting. In.
- [9] Yu, K. and Hsu, C. and Yang, S.-M. (2019). A Model Integrating ARIMA and ANN with Seasonal and Periodic Characteristics for Forecasting Electricity Load Dynamics in a State. Paper presented at the 2019 IEEE 6th International Conference on Energy Smart Systems (ESS).
- [10] Aradhya, G. and Rao, A. and Mohammed, M. M. (2019). A Novel Hybrid Approach for Time Series Data Forecasting Using Moving Average Filter and ARIMA-SVM. In *Emerging Technologies in Data Mining and Information Security* (pp. 369-381): Springer.
- [11] Bedi, J. and Toshniwal, D. (2019). Deep learning framework to forecast electricity demand. *Applied energy*, 238, 1312-1326.
- [12] Badal, L. and Franzén, S. (2019). A Comparative Analysis of RNN and SVM: Electricity Price Forecasting in Energy Management Systems. In.
- [13] Muzaffar, S. and Afshari, A. (2019). Short-Term Load Forecasts Using LSTM Networks. *Energy Procedia*, 158, 2922-2927.
- [14] Bano, H. and Tahir, A. and Ali, I. and Haseeb, A. and Javaid, N. (2019). Electricity Load and Price Forecasting Using Enhanced Machine Learning Techniques. Paper presented at the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.
- [15] Khan, S. and Javaid, N. and Chand, A. and Khan, A. B. M. and Rashid, F. and Afridi, I. U. (2019). Electricity load forecasting for each day of the week using deep CNN. Paper presented at the Workshops of the International Conference on Advanced Information Networking and Applications.
- [16] Mujeeb, S. and Javaid, N. and Ilaahi, M. and Wadud, Z. and Ishmanov, F. and Afzal, M. K. (2019). Deep long short-term memory: A new price and load forecasting scheme for big data in smart cities. *Sustainability*, 11(4), 987.
- [17] Wu, W., and Liao, W. and Miao, J. and Du, G. (2019). Using Gated Recurrent Unit Network to Forecast Short-Term Load Considering Impact of Electricity Price. *Energy Procedia*, 158, 3369-3374.
- [18] Bissing, D. and Klein, M. T. and Chinnathambi, R. A. and Selvaraj, D. F. and Ranganathan, P. (2019). A Hybrid Regression Model for Day-Ahead Energy Price Forecasting. *IEEE Access*, 7, 36833-36842.
- [19] Guo, J. and He, H. and Sun, C. (2019). ARIMA-based Road Gradient and Vehicle Velocity Prediction for Hybrid Electric Vehicle Energy Management. *IEEE Transactions on Vehicular Technology*.
- [20] Balaji, A. J., and Harish Ram, D. and Nair, B. B. (2019). A deep learning approach to electric energy consumption modeling. *Journal of Intelligent & Fuzzy Systems*(Preprint), 1-7.
- [21] Tokgöz, A. and Ünal, G. (2018). An RNN based time series approach for forecasting Turkish electricity load. Paper presented at the 2018 26th Signal Processing and Communications Applications Conference (SIU).
- [22] Chang, Z. and Zhang, Y. and Chen, W. (2018). Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting. Paper presented at the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS).
- [23] Agrawal, R. K., and Muchahary, F., and Tripathi, M. M. (2018). Long term load forecasting with hourly predictions based on long-short-term-memory networks. Paper presented at the 2018 IEEE Texas Power and Energy Conference (TPEC).
- [24] Jiang, L. and Hu, G. (2018). Day-ahead price forecasting for electricity market using long-short term memory recurrent neural network. Paper presented at the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV).
- [25] Rahman, A. and Srikumar, V. and Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied energy*, 212, 372-385.
- [26] Bouktif, S. and Fiaz, A. and Ouni, A. and Serhani, M. (2018). Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, 11(7), 1636.
- [27] Kim, N. and Kim, M. and Choi, J. K. (2018). LSTM Based on Short-term Electricity Consumption Forecast with Daily Load Profile Sequences. Paper presented at the 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE).

- [28] Kim, T.-Y. and Cho, S.-B. (2018). Predicting the Household Power Consumption Using CNN-LSTM Hybrid Networks. Paper presented at the International Conference on Intelligent Data Engineering and Automated Learning.
- [29] Peng, L. and Liu, S. and Liu, R. and Wang, L. (2018). Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy*, 162, 1301-1314.
- [30] Bedi, J. and Toshniwal, D. (2018). Empirical mode decomposition based deep learning for electricity demand forecasting. *IEEE Access*, 6, 49144-49156.
- [31] Kuo, P.-H. and Huang, C.-J. (2018). An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability*, 10(4), 1280.
- [32] Choi, J. Y. and Lee, B. (2018). Combining the LSTM network ensemble via adaptive weighting for improved time series forecasting. *Mathematical Problems in Engineering*, 2018.
- [33] Shi, H. and Xu, M. and Li, R. (2017). Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Transactions on Smart Grid*, 9(5), 5271-5280.
- [34] Amarasinghe, K. and Marino, D. L. and Manic, M. (2017). Deep neural networks for energy load forecasting. Paper presented at the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE).
- [35] Marino, D. L., and Amarasinghe, K., and Manic, M. (2016). Building energy load forecasting using deep neural networks. Paper presented at the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society.
- [36] Amini, M. H., and Kargarian, A. and Karabasoglu, O. (2016). ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electric Power Systems Research*, 140, 378-390.
- [37] Voronin, S. and Partanen, J. (2014). Forecasting electricity price and demand using a hybrid approach based on wavelet transform, ARIMA, and neural networks. *International Journal of Energy Research*, 38(5), 626-637.
- [38] Shafie-Khah, M. and Moghaddam, M. P. and Sheikh-El-Eslami, M. (2011). Price forecasting of day-ahead electricity markets using a hybrid forecast method. *Energy Conversion and Management*, 52(5), 2165-2169.
- [39] Kandanand, K. (2011). Forecasting electricity demand in Thailand with an artificial neural network approach. *Energies*, 4(8), 1246-1257.
- [40] Che, J. and Wang, J. (2010). Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling. *Energy Conversion and Management*, 51(10), 1911-1917.
- [41] Olah, C. (2015). Understanding lstm networks. Retrieved from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [42] Rashid, T. A., and Fattah, P., and Awla, D. K. (2018). Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms. *Procedia Computer Science*, 140, 324-333.
- [43] Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- [44] Siami-Namini, S. and Namin, A. S. (2018). Forecasting economic and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386. Retrieved from <https://arxiv.org/ftp/arxiv/papers/1803/1803.06386.pdf>