Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 154 (2019) 102–108

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology, ICICT 2019

Deep Learning-Based Power Usage Forecast Modeling and Evaluation

Fan Liang[a], Austin Yu[a], William G. Hatcher[a], Wei Yu[a], and Chao Lu[a]*

[a] *Dept. of Computer and Information Sciences, Towson University, 7800 York Road, Towson, Maryland, USA 21252*

Abstract

The growing Internet of Things (IoT) provides significant resources to be integrated with critical infrastructures to enable cyber-physical systems. More specifically, the deployment of smart meters for electricity usage monitoring in the smart grid can provide granular and detailed information from which power load forecasting can be carried out. However, the accurate prediction of long-term power usage remains a challenging issue. In light of many recent advances, deep learning has the potential to significantly improve the ability to assess data and make predictions, and is already rapidly changing the world we live in. As such, in this paper, we consider the use of deep learning, via Recursive Neural Network (RNN) and Long Short-Term Memory layers, for the long-term prediction of localized power consumption. In particular, we consider the optimization of both data feature sets and neural network models, developing three model-feature combinations to maximize prediction accuracy and minimize error. Through detailed experimental evaluation, our results demonstrate the ability to achieve highly accurate predictions over periods as large as 21 days through the integration of correlated features.

Keywords: Deep Learning, Smart Grid, Internet of Things, Cyber-Physical Systems

1. Introduction

The Internet of Things (IoT) envisions the massive deployment of smart computing sensors and actuators throughout our physical world that are broadly connected and reachable through the Internet [1]. Enabled by the

* Corresponding author. Tel.: +1-410-704-3950; *E-mail address:* clu@towson.edu

10.1016/j.procs.2019.06.016

decreasing costs and increasing power of computation and electronic storage, IoT systems are already being deployed worldwide, especially for consumer use. In addition, IoT enables intelligent critical infrastructure systems, such as smart cities, smart grid, smart transportation, etc. that can be leveraged for efficiency and fairness in resource management on industrial scales [2, 3].

In the context of the smart home and smart grid, the ability to absorb and analyze massive amounts of big data provide for unprecedented levels of command and control for electrical load distribution, industrially and by individual users alike, supporting a new level of situation awareness. Moreover, emerging tools for machine learning have been used to achieve increasingly accurate predictions on continuous time-series data in a variety of applications. As the state-of-the-art to achieve highly accurate data analysis, deep learning applies significant coalitions of computing neurons to approximate dataset distributions. What's more, deep learning has been applied to abstract problems and exceeded even human capabilities, and has been used to extract meaningful and hidden information from massive datasets [4, 5].

In this paper, we apply deep learning, via Recursive Neural Network (RNN) and Long Short-Term Memory layers, to address the problem of electrical load forecasting in the smart grid. Using a dataset comprised of real-world smart meter readings, local temperature information, and weekday/weekend identification, we construct several deep learning models for comparison and evaluation. Moreover, our designed models each implement a different subset of the entire dataset for comparison of accuracies based on feature set complexity.

To be specific, the key contributions of this paper are summarized as follows. First, we develop an approach to optimize deep learning for smart meter power consumption forecasting via feature selection and model design and consider three primary groups of distinct feature and model combinations. Second, we conduct a detailed experimental analysis of our developed groups, comparing the error and accuracy. Our experimental results demonstrate significantly accurate long-term prediction (as much as 21 days).

The remainder of this paper is as follows. In Section 2, we detail our approach toward smart meter power forecasting via deep learning. In Section 3, we provide an evaluation comparing several methods for power forecasting of smart meter data. Finally, in Section 4, we provide some concluding remarks.

2. Our Approach

In this section, we introduce our approach to forecasting smart meter power consumption.

The power consumption data we are utilizing was published by the Bristol City Council, of Bristol, England on March 13, 2014 [6]. The data details nearly a year's worth of measurements taken at half-hour intervals starting at 00:30 and ending at 24:00 for each day provided. Moreover, to expand the dataset and apply additional and relevant features, historical temperature data were collected [7]. The collected temperature data spans the same time and dates of the matching power consumption data and pertains to the same location in Bristol, England.

The data applied to machine learning models are quite significant, and determine the outcome of the prediction results in terms of accuracy, recall, precision, etc. In more detail, we are interested in predicting the bi-hourly power usage over a period of several days (11 to 21). Applying only the usage data (in KWh), we have only a small volume of data to predict from. In contrast, by increasing the number of correlated features, we can increase the complexity of our dataset and potentially increase our prediction accuracy.

Considering the usage dataset and additional temperature data detailed above, we can apply multiple features that have dependencies with power usage. Table I provides an example of the full input feature set. Note that we have additionally included the feature of Day Type, which is derived from the Date feature, where a "0" value indicates a weekday and a "1" indicates a weekend day. To compare the data at different levels of granularity, we define three distinct dataset groups (One for single feature input and two for multi-feature input) on which to train and evaluate our deep learning models.

These groups are defined as follows: (i) Group I: We select *Power Usage* as the single feature for training input, arranged by date and time. This data can be applied as a single input vector of a learning model. (ii) Group II: We select four input features (*Power Usage, Date, Time,* and *Day Type*), which can be considered to be interdependent. Unlike Group I, this multi-feature dataset must be applied as a multi-dimensional matrix. (iii) Group III: We utilize the same features as Group II, with the addition of the *Temperature* feature. In this case, we want to evaluate whether adding further dependency factors will affect the accuracy.

In applying deep learning to our problem, the proper network structures must be selected. In our case, we are conducting supervised learning to achieve accurate prediction [8]. In addition, we thus consider our problem as a regression problem to predict continuous, rather than discreet, output.

In our scenario, we want to predict the power usage over several days. For the single feature dataset (Group I), the data structure is a vector, and thus we consider the application of a dense fully-connected network with a dense input layer. In contrast, the multi-feature input required for Groups II and III necessitates input in the form of a multi-dimensional matrix. In this case, we select Long Short-Term Memory (LSTM) as the input layer and apply a Recursive Neural Network (RNN) structure to handle the matrices of time-series data.

Table 1. Dataset Example of Selected Features

ID	Date	Day Type	Time	Temperature	Usage
1	3/10/2013	1	00:30	34	24.2
2	3/10/2013	1	01:00	34	26
3	3/10/2013	1	01:30	34	30.8
4	3/10/2013	1	02:00	34	31.8
5	3/10/2013	1	02:30	34	31.5
6	3/10/2013	1	03:00	34	30.4
…	…	…	…	…	…

3. Evaluation

In this section, we detail the evaluation of our designed Deep Learning-based prediction models. First, we provide an overview of our methodology, detailing the testing environment, model parameter tuning, and evaluation metrics. We then present and analyze the detailed power usage forecast results.

3.1. Methodology

We now briefly introduce the key components of our testbed. We developed deep learning models in Python utilizing the TensorFlow machine learning back-end and Keras front-end API. The TensorFlow library also supports supervised learning, unsupervised learning, reinforcement learning, etc. In this case, we are interested in supervised learning, as noted above in Section III-B, using regression analysis. The testbed structure is shown in Figure 1.

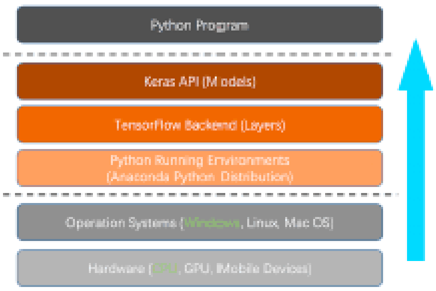

Figure 1. The Structure of our testbed

In addition, Keras is a high-level python library that serves as a high-level API to support different machine learning libraries. Through Keras, developers can easily build complex neural networks transferable between machine learning back-ends and simplify the coding process.

In this experiment, we have implemented deep recursive neural networks with dense (fully-connected) hidden layers, and the initial input layers distinguish the groups (dense vs. LSTM). In fully-connected layers, each neuron is directly connected with the all the neurons in the previous layer [9]. In addition, each neuron applies an activation function on the data that it receives. The weight values of the activation function are randomly initiated and then updated through each back-propagation step as determined by the optimizer used for gradient descent. In our neural networks, we have selected ReLU (Rectified Linear Unit) as the activation function, which is an implementation of the rectifier function. Compared with the traditional Sigmoid activation function, the ReLU function simply sets the threshold to zero, and the computational overhead is reduced significantly. Moreover, research has shown that ReLU improves the learning performance [4, 8, 10].

Based on our three input datasets, we identify different neural network parameters, the most critical being the number of epochs and the batch size. The number of epochs defines the iteration time for each set of input, and increasing the number of epochs typically increases the accuracy of the result. However, the computation time will also increase [11]. Moreover, the impact on accuracy is not linear. From a preliminary evaluation, we note that the multi-feature input networks experience much faster convergence than the single feature input. Thus, we set the number of epochs of the single feature input model (Group I) as 3,000 and multi-feature input models (Group II and III) as 100. Concerning the batch size, this is defined as the size or volume of data that is input into the model at one time. Large batch sizes will cost more in terms of memory usage but can improve calculation speed. After preliminary evaluation, we set the batch size as 100 for all three groups.

To evaluate our deep learning models, we use two primary metrics, Mean Absolute Error (MAE) and Mean Squared Error (MSE) [12]. These are loss functions that measure the accuracies of our prediction and are the two most common metrics to evaluate continuous variables. The formulas for MAE and MSE are given as:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| ; \qquad \text{MSE} = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2 ;$$

where y_j is the j^{th} sample value, $\hat{y}_j$ is the j^{th} prediction, and n is the total number of predictions. More specifically, the MSE loss function demonstrates a greater change due to large prediction, as the square compounds the observed effect in comparison with MAE. In addition, we further evaluate the predictions through observation of the predicted fit with the true result.

3.2. Evaluation results

We now detail the results of our designed evaluation. In our experiments, we divided the total dataset into training and testing subsets to carry out supervised learning. In this case, the ratio of the training set to test set is 70% to 30%. After completing the training process, we predict the next 11 days and 21 days of power usage separately.

In Figures 2 through 7, we can see the prediction results for each of the three dataset/model combinations (Groups I through III) for both 11 days and 21 days. Specifically, in each figure, we show the actual usage in blue and the predicted values in orange. Note that the x-axis represents Time, with each calculated prediction being 30 minutes apart, and the y-axis displays Power Usage (KWh).

Table 2. MAE and MSE Scores for Each Group.

Group	MAE (KWh)			MSE (KWh^2)		
	Training	Testing		Training	Testing	
		11 Days	21 Days		11 Days	21 Days
I	7.2923	8.4615	13.5279	71.6415	84.6825	91.5792
II	3.8553	4.3513	5.1388	41.3352	53.9723	62.4797
III	2.1338	2.3997	2.5113	14.8872	16.3652	16.3991

In Group I, the only feature that we have selected as input is Power Usage. After training our dense network model, we can see from Figures 2 and 3, for 11-day and 21-day predictions respectively for Group I, that the model is clearly under-fitting, and is not nearly accurate enough. Moreover, as we can see in Table II, Group I was the poorest performer with the highest training and test error. Specifically, the Group I Training MAE and MSE are

7.2923 and 71.6415, respectively, while the 11-day test MAE is 8.4615 and the MSE is 84.6825. Likewise, for the 21-day forecast, the MAE is 13.5279 and the MSE is 91.5792. In this case, the prediction curve is too soft and does not match the fine details in the crest and trough regions.

In adding more features and implementing the LSTM layers, the Group II model demonstrates much better accuracy, nearly halving the MAE and MSE results of the Group I model. Specifically, again as observed in Table II, the Training MAE result is 3.8553 and the MSE is 41.3352, in comparison with the Group I Training MAE and MSE of 7.2923 and 71.6415, respectively. Likewise, in Figures 4 and 5, the prediction curves achieve a noticeably better result. However, we can also observe that the prediction appears less accurate as time increases, with the details disappearing at around the seventh or eighth day. Thus, using only Date, Time, Day Type, and Power Usage features, accurate forecasting for the far future falls short.

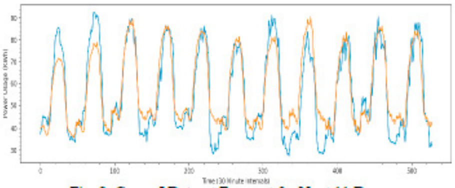

Fig. 2. Group I Dataset Forecast for Next 11 Days.

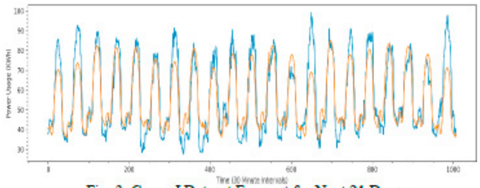

Fig. 3. Group I Dataset Forecast for Next 21 Days.

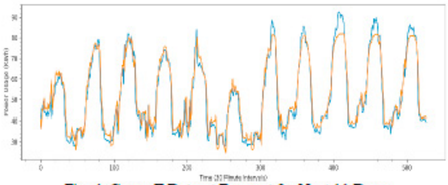

Fig. 4. Group II Dataset Forecast for Next 11 Days.

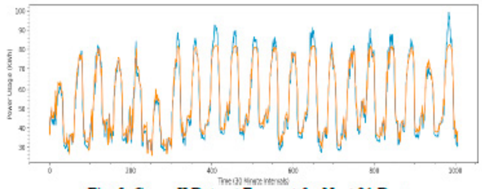

Fig. 5. Group II Dataset Forecast for Next 21 Days.

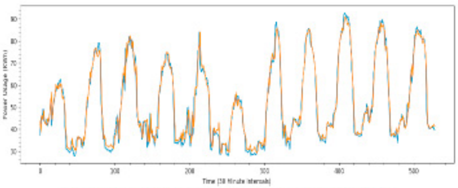

Fig. 6. Group III Dataset Forecast for Next 11 Days.

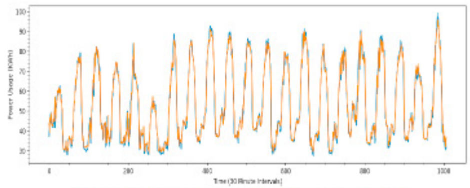

Fig. 7. Group III Dataset Forecast for Next 21 Days.

Finally, with the addition of the Temperature feature, we observe that Group III performs the best out of all three groups. As we can see in Figures 6 and 7, this model produces the most accurate prediction overall and also appears to maintain high accuracy over a long prediction window (21 days). Moreover, aside from having the lowest MAE and MSE scores of all groups in all categories, as observed in Table II, Group III's MAE and MSE scores also display the lowest deviation and variance across Training and Testing. For example, the absolute lowest MAE and MSE scores overall were 2.1338 and 14.8872, respectively, in Group III Training. The highest scores in Group III were MAE and MSE of 2.5113 and 16.3991, respectively. Thus, adding temperature to the dataset further reduced MAE and MSE, increased the complexity of the model and added relevant data, and the predicted curve better fits the actual data.

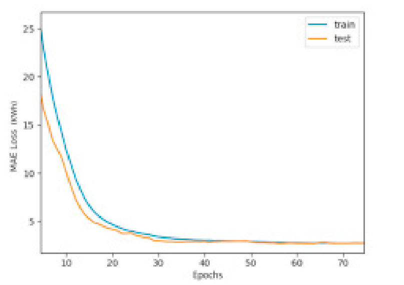

Fig. 8. Group III MAE for 11 Days Prediction.

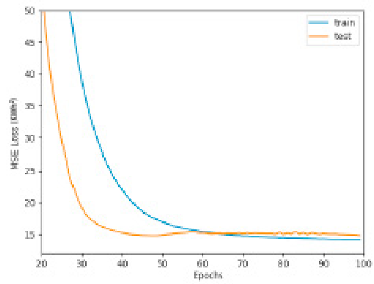

Fig. 9. Group III MSE for 11 Days Prediction.

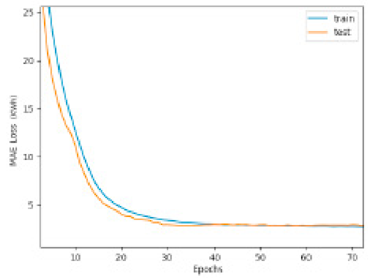

Fig. 10. Group III MAE for 21 Days

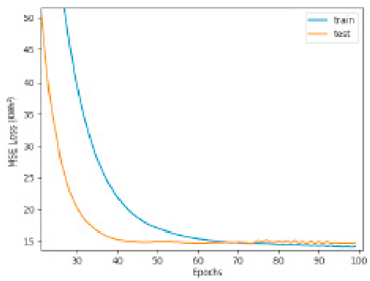

Fig. 11. Group III MSE for 21 Days

Furthermore, Figures 8 through 11 show the MAE and MSE trends during the training and testing processes for Group III. These help to demonstrate that the LSTM layers and ReLU activation function can accelerate the fitting process. Note that the MAE scores for Group III converge around 20 to 30 epochs, as does MSE testing scores, while MSE training scores converge around 50-60 epochs, much quicker than what is observed for Group I. This indicates that our deep LSTM model is functional and efficient.

4. Conclusion

In this paper, we have focused on creating an efficient deep learning model to forecast smart meter power usage based on the power consumption history. In particular, we have applied real-world datasets, processing the data into three groups, each with different sets of features. Furthermore, we have considered two deep learning models to analyze the data, one a fully-connected DNN and the other an LSTM RNN. Via thorough experimental evaluation, the multi-feature input dataset trained in the LSTM RNN deep learning model provides the most rapid fitting speed and the highest load forecasting accuracy. In addition, we applied the best performing LSTM RNN model to predict power usage for other nearby buildings using same training datasets, and the prediction results are highly accurate as well, indicating that our proposed model is both adaptable and stable.

5. Acknowledgement

This work was supported in part by the US National Science Foundation (NSF) under grants: CNS 1350145 (Faculty CAREER Award), and the University System of Maryland (USM) Endowed Wilson H. Elkins Professorship Award Fund. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the agencies.

6. References

1. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.
2. G. Xu, W. Yu, D. Griffith, N. Golmie and P. Moulema, "Toward Integrating Distributed Energy Resources and Storage Devices in Smart Grid," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 192-204, Feb. 2017.

3. W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, 6: 6900 - 6919, November 2017.
4. W. G. Hatcher and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," IEEE Access, vol. 6, pp. 24411-24432, 2018. doi: 10.1109/ACCESS.2018.2830661.
5. W. Yu, D. An, D. Griffith, Q. Yang, and G. Xu, "Towards Statistical Modeling and Machine Learning Based Energy Usage Forecasting in Smart Grid," ACM SIGAPP Applied Computing Review, vol. 15, no. 1, pp. 6–16, 2015.
6. "Energy consumption for selected bristol buildings from smart meters by half hour," https://data.gov.uk/dataset/1b41eb5c-4b7a-4f60-ae0f-3a78c73953bf/energy-consumption-for-selected-bristol-buildings-from-smart-meters-by-half-hour, March 2014, accessed: 2018-05-02.
7. https://www.timeanddate.com/, 2018, accessed: 2018-05-02.
8. S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised Machine Learning: A Review of Classification Techniques," Emerging artificial Intelligence Applications in Computer Engineering, vol. 160, pp. 3–24, 2007.
9. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, p. 436, 2015.
10. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (almost) from Scratch," ArXiv e-prints, Mar. 2011.
11. C. Liu, Z. Jin, J. Gu, and C. Qiu, "Short-term Load Forecasting Using a Long Short-term Memory Network," in Proc. of 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Sept 2017, pp. 1–6.
12. C. M. Cheung, R. Kannan, and V. K. Prasanna, "Temporal Ensemble Learning of Univariate Methods for Short Term Load Forecasting," in Proc. of 2018 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), Feb. 2018, pp. 1–5.