LENARD: Lightweight ENsemble LeARner for MeDium-term Electricity Consumption Prediction

Onat Gungor[1,3], Jake Garnier[2], Tajana S.Rosing[2], and Baris Aksanli[3]

[1]Department of Electrical and Computer Engineering, University of California San Diego
[2]Department of Computer Science and Engineering, University of California San Diego
[3]Department of Electrical and Computer Engineering, San Diego State University
ogungor@ucsd.edu, jgarnier@ucsd.edu, tajana@ucsd.edu, baksanli@sdsu.edu

***Abstract*—In this work, we propose a lightweight ensemble learner for individual house level electricity consumption prediction. We first implement five different prediction algorithms: ARIMA, Holt-Winters, TESLA, LSTM and Persistence. Among single prediction algorithms, LSTM performs best with 0.0195 MSE value on average. Then, we combine these predictions using neural network based ensemble learner which improves performance of best algorithm (LSTM) on average by 72.84% and by up to 99.13%. Finally, we apply pruning to the weights of our ensemble network to decrease the computational cost of our model. Applying pruning leads to 10.9% less error and 27% fewer number of parameters. We show that our pruned ensemble learner outperforms state-of-the-art ensemble methods.**

I. Introduction

According to the energy trend report of International Energy Agency (IEA) [1], by 2040 total electricity demand will reach 44 thousand TWh and remarkably residential electricity demand will constitute 25% of this total. Hence, it is essential for electricity providers/utilities to match electricity supply and demand at all times to reduce their operational costs and keep safe operation of their infrastructure. To do this, the utility providers rely on short- (or long) term predictions [2].

We can categorize electricity demand forecasting into three classes based on the forecast horizon. In short-term prediction, the electricity demand forecast is performed for a few minutes up to a few days ahead. Medium-term forecast can make predictions from a few days to a few months ahead. Lastly, long-term forecasting can predict in months, quarters or even years [3]. These three type of predictions can be used at different sectors and for different purposes. Besides, they all have an important role in reducing the total cost of an electric utility where even small improvements in prediction performance can lead to huge gains. To illustrate, a rough estimate of savings from a 1% reduction in the total prediction error for a utility with 1GW peak load is $500K per year due to long-term load forecasting, $300K per year due to short-term load forecasting, and $600K per year due to short-term load and price forecasting [4]. In our work, we focus on medium-term electricity prediction where we predict 14-day ahead.

978-1-7281-6127-3/20/$31.00 ©2020 IEEE

Utilities previously perform aggregate level predictions where they combine many individual house data. Nevertheless, focusing on individual houses is now possible with the advancements in smart technology (e.g. smart meters, smart appliances). This also brings deeper understanding about each house, allowing the electricity providers to have targeted programs for each customer (such as reduce your use day [5] etc.). Additionally, with the rise of smaller and more independent energy systems (e.g. micro grids), it has become more relevant to perform these predictions at the house-level, rather than at the aggregate level.

For electricity consumption prediction, there are distinctive number of approaches ranging from simple time series algorithms to complex machine learning approaches [6]. Using only one prediction algorithm across all houses may not lead to accurate predictions since each house represents a different system with distinct characteristics (e.g. number of people, their appliance usage habits, their employment status, the physical properties of the house etc.).

Hence, there is a need for a systematic approach where there are multiple prediction algorithms available and evaluated for suitability. For instance, Gungor et al. [7] show that choosing one algorithm to perform prediction across all houses might be severely under-performing, and they propose an algorithm selection methodology which selects the best prediction algorithm for a house given the house characteristics. In general, ensemble methods are preferred where different model's predictions are combined in an algorithmic manner. The advantage of these methods is that they break the assumptions inherent in single prediction algorithms and provide more generalizable models.

Neural networks are used as ensemble models [8]. The main issue with neural networks is that they are computationally expensive models since they have millions of parameters and costly matrix multiplications. Especially, in neural-network based ensemble models there is a need to alleviate this burden on the model by shrinking the network. There are different approaches to have smaller networks such as product quantization [9] and pruning [10]. In this work, we implement pruning to eliminate negligible weights from our ensemble network, which leads to increased prediction accuracy and fewer number of parameters, reducing the computational burden of

neural networks significantly.

In this paper, we propose a lightweight ensemble learner to predict individual house level electricity consumption. We use residential electricity consumption data from Office of Energy Efficiency & Renewable Energy (EERE) [11]. We first implement single prediction algorithms (namely ARIMA [12], Holt-Winters [13], TESLA [14], LSTM [15], and Persistence). Among single prediction algorithms, LSTM performs best with 0.0195 MSE value on average. Afterwards, we combine these predictions using a neural network based ensemble learner. Our ensemble learner improves best single prediction algorithm's (LSTM) performance on average by 72.84% and up to 99.13%. The improvement over other prediction algorithms is much higher. Finally, we apply pruning to the individual weights of our ensemble network to decrease the number of parameters of our model. Applying pruning on average leads to 10.9% less error and 27% fewer number of parameters. We also demonstrate that our pruned ensemble learner performs better than state-of-the-art ensemble methods.

II. Related Work

There are variety of methods used in energy consumption prediction. Many works focus on implementing single algorithm for prediction. Machine-learning based prediction methods extract features using time series data. Examples can be support vector machine [16], LSTM [17] etc. Time series forecasting methods are also very popular in prediction. ARIMA [18], Holt-Winters [19] are some examples for these type of models. For residential individual house level energy prediction, using single method may not be the best option since each house represents different systems fundamentally. Therefore, hybrid approaches such as ensemble models should be considered for better prediction.

Ensemble learning models utilize the predictions of different methods with a given learning algorithm to improve the robustness over a single method [20]. These models are heavily used in time series forecasting for different kind of predictions. The reason why these methods are used is twofold 1) better generalization (i.e. less over-fitting) 2) higher prediction accuracy. Alobaidi et al. [21] propose an ensemble method to predict day-ahead average energy consumption. The authors in [22] propose Stacking Multi-Learning Ensemble to predict global oil consumption. Avand et al. [23] propose a tree-based intelligence ensemble approach for spatial prediction of potential groundwater. Shan et al. [24] use ensemble prediction model called gravity gated recurrent unit electricity consumption model to predict electricity consumption of a five-star hotel building in Shanghai, China. Lee et al. [25] combine stacking ensemble learning method (SELM) and self-organizing map for electric load forecasting. Yao et al. [26] combine extreme gradient boosting, extreme learning machine, multiple linear regression with support vector regression. Al-Rakhami et al. [27] use extreme gradient boosting (XGBoost) for cooling and heating load predictions in a residential building.

Pruning in neural networks is one method to shrink the network size. Essentially, this method eliminates the insignificant weights. There are several examples from different applications such as deep learning [28], and convolutional neural networks [29]. To the best of our knowledge, there is no prior work combining both ensemble models and pruning to alleviate the computational complexity.

In our work, we combine all these three aspects (single prediction algorithms, ensemble models and pruning) into one framework where we present our readers to create a lightweight ensemble model for medium-term electricity consumption prediction.

III. Time Series Forecasting Methods

Electricity consumption data is uni-variate which is composed of pairs of a time stamp and a value. Time stamp granularity can be a second, a minute, or an hour. Value is the corresponding electricity consumption between two consecutive time steps. Since this uni-variate data is a type of time series data, it is cogent to utilize time series prediction algorithms. Essentially, these methods find repeating patterns in the past data for forecasting. In our work, we use 5 different state-of-the-art methods with varying complexities:

Holt-Winters: This model is formed by the combination of three parameters: mean, trend and seasonality. Mean (α) considers the average of data points; trend (β) estimates increasing or decreasing direction (i.e. positive or negative slope) in the provided data; seasonality (γ) finds recursively observed pattern [13]. There are multiplicative and additive versions of Holt-Winters. We select additive version of Holt-Winters with the following formulation:

$$Y_t = \alpha + \beta_t + \gamma_t + \epsilon_t \tag{1}$$

where α is mean, β is trend, γ is the seasonality factor which all can take any value in [0,1]. Random error, ϵ are assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2.

ARIMA: Auto regressive Integrated Moving Average is composed of three components: number of previous time periods (p), degree of difference (d) and number of previous time periods of error term (q). Equation (2) formulates an ARIMA model [12]:

$$\begin{aligned} y_t &= \theta_0 + \phi_1 * y_{t-1} + \phi_2 * y_{t-2} + ... + \phi_p * y_{t-p} \\ &+ \epsilon_t - \theta_1 * \epsilon_{t-1} - \theta_2 * \epsilon_{t-2} - ... - \theta_q * \epsilon_{t-q} \end{aligned} \tag{2}$$

where y and ϵ are the actual value and random error at time period t, respectively; ϕ and θ are model parameters. p and q are integers and often referred to as orders of the model. Random errors, ϵ, are assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2.

TESLA: Taylor Expanded Solar Analog Forecasting predicts solar energy availability which can be used for any kind of time series energy prediction. It finds the relation between data points and the desired prediction data using Taylor expansion. Hence, complex relationships (such as logarithmic, exponential, etc.) can be easily reduced to simple polynomials. TESLA can use first, second, or third-degree polynomials in

accordance with the available data size and desired accuracy level. The model is trained by providing time window (i.e. how many previous days our model is going to use) and prediction window (i.e. how many subsequent days our model is going to predict) [14]. Generic formulation of TESLA is as follows:

$$\begin{aligned} &\sum_{i=o}^{n} C_i x_i \quad (1^{st} order) \\ &\sum_{i=0}^{n} \sum_{j=0}^{i} C_{ij} x_i x_j \quad (2^{nd} order), \quad etc. \end{aligned} \tag{3}$$

where C_{ij} represents coefficients learned with observations, and $x_0 = 1$ (the constant factor). The resulting equation is $Ax = B$, where A is the matrix of input observations; x is the vector of coefficients, and B is the vector of output observations, each entry correlating with the corresponding row of A, and solved by least squares estimation.

LSTM: Long Short-Term Memory network is a specific type of recurrent neural networks (RNN). RNNs are capable of keeping past information over time. Distinctly, LSTM has a special memory cell which is able to store information over longer periods of time. Updates in this cell can happen by the activation of three various gates: 1) forget gate (the memory cell is cleared completely), 2) input gate (memory cell stores the received input), and 3) output gate (next neurons obtain the stored knowledge from memory cell) [30].

Persistence: This model constitutes our baseline for comparison purposes. The main idea of persistence is that tomorrow's prediction is equal to what we observe today. To illustrate, if we measure today's temperature as 27°C then it means that tomorrow is also going to be 27°C [31]. This approach can be extended by selecting different time frames, such as using previous 12 hours to predict next 12 hours. In our work, we select the portion before our test data to predict whole test data. We specifically match the time stamps in this method.

IV. Lightweight Ensemble Learner

A. Base Ensemble Learner

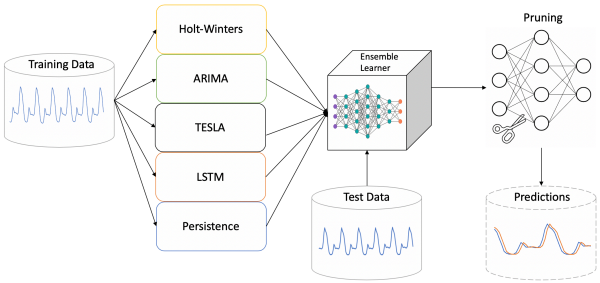

Fig. 1. Lightweight Ensemble Learner Framework

We utilize ensemble learner for hourly electricity consumption prediction. Given predictions of different methods, ensemble models blend these predictions systematically. These models are mostly used in classification tasks where bagging and boosting are really effective in developing single classifier's performance [32]. We focus on regression-based ensemble model where our goal is to predict residential electricity consumption values. We present our framework in Figure 1. First, we pre-process the electricity consumption data, then separate this data into training and test data sets. We train the prediction algorithms using the training data and make prediction on the test data. At this step, we give prediction values (obtained from single prediction algorithms) and real test data to our ensemble learner. Our ensemble learner is a fully connected neural network where it learns the importance of different algorithms, i.e. ensemble learner finds the optimal weights for each prediction method. Ensemble model finally provides prediction values for test data set.

As our ensemble learner, we select dense (i.e. fully connected) neural network which is composed of 1 input, 2 hidden and 1 output layers. Input layer has 5 neurons which represent our five different prediction methods, in both hidden layers we have 256 neurons, in output layer we have 1 neuron symbolizing the ultimate prediction value. As indicated in our framework, for ensemble learner we use test data. For ensemble learner training, this data (i.e. test data) is again divided into training and test and we perform error calculations over newly generated test data. Specifically, our ensemble learner uses 59 day data to predict 14 day electricity consumption. We call this model base ensemble learner which is going to be pruned in the following subsection.

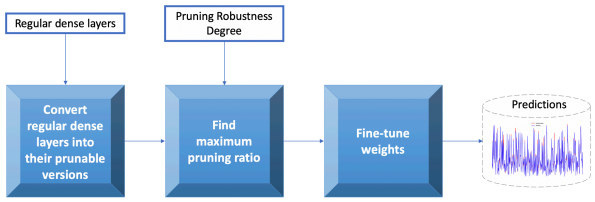

Fig. 2. Pruning Phase

B. Pruned Ensemble Learner

After we obtain predictions from our base ensemble learner, we initiate the pruning phase to decrease the complexity (and possible redundancy) of our ensemble model. Pruning, in its most basic explanation, is a process which eliminates redundant weights from the network. Redundant in any neural network signifies a value which is below a certain threshold. We eliminate small-valued weights from our network while keeping the same or better accuracy values. In other words, we get rid of the unimportant connections from the network. After elimination of redundant weights, we fine-tune the lighter model to adapt the weights. Pruning brings the advantage of smaller networks, faster training and lower total energy consumption for devices. To the extreme, for small amounts of compression, pruning can sometimes increase accuracy [33]. The most convoluted part of our network is between first and second hidden layer where we have 65,536 (256*256) weight values. Hence, we focus on the weights between these two

layers to alleviate the computational complexity. The main concern is to select an appropriate pruning ratio which is equal to the number of pruned weights divided by total number of weights. In order to find a proper pruning ratio, we define pruning robustness degree which identifies how much worse total test error can be. Accordingly, we select the maximum pruning ratio (among set of ratios) conforming to the selected pruning robustness degree. For example, if we select pruning robustness degree as 10% and total test error before pruning is 0.02 then we can tolerate an error up to 0.022. After we find maximum pruning ratio, we reconstruct the pruned model by eliminating corresponding weights. Afterwards, we fine-tune whole network's weights to minimize the total error. After fine-tuning, we can obtain error values of our pruned ensemble learner. Our pruning phase is summarized in Figure 2 for more clarity. To summarize, it is composed of three consecutive steps: 1) conversion of regular dense layers into prunable versions 2) determination of maximum pruning ratio given pruning robustness degree 3) fine-tuning.

V. Experimental Analysis

A. Dataset

We use residential electricity consumption data from the Office of Energy Efficiency & Renewable Energy (EERE) [11]. We use 936 houses from 50 states across the US. At each house, we have hourly electricity consumption information in kW, from year 2013. In total, we have 8760 data points per house. For each house, We divide our complete data set of 8760 data points into training and test sets, using a 80-20 training/test ratio, creating *training1* and *test1* sets. Then, we feed *test1* data set into the ensemble methods to construct the networks. We again use a 80-20 training/test ratio, creating *training2* and *test2* sets. For consistency among the individual algorithms and our ensemble methods, we present error values measured in the *test2* set, which corresponds to 14 days (i.e. medium-term horizon for energy consumption prediction).

B. Base Ensemble Learner

For our single prediction algorithms, we use 5 different methods: Holt-Winters, ARIMA, TESLA, LSTM, and persistence forecast. The prediction values obtained from these algorithms are given to the ensemble learner, that learns the importance of each method. We select our ensemble learner to be 4-layer dense neural network, i.e. all neurons in two consecutive layers are connected to each other. Particularly, our neural network consists of 5 neuron input layer, two 256 neuron hidden layers and 1 neuron output layer. The reason for the number of neurons selected in input and output layers is trivial (i.e. we have 5 prediction algorithms and 1 ultimate output prediction value). For the number of hidden layers, we select 2 since it can approximate any smooth mapping to any accuracy [34]. For the number of neurons in the hidden layers, we use a specific approximation function as following [34] which is suitable for dense neural networks:

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))} \tag{4}$$

TABLE I
Base Ensemble Learner's Average and Maximum Improvement over Single Prediction Algorithms

Prediction Method	Average (%)	Maximum (%)
Persistence	98.03	99.94
ARIMA	95.80	99.98
Holt-Winters	94.16	99.98
TESLA	88.86	99.26
LSTM	72.84	99.13

where N_i is the number of input neurons, N_o is the number of output neurons, N_s is the number of samples in training data set, and α is a scaling factor between 2 and 10. For our ensemble learner, we select 2 as α value and we round up the obtained value to comply with 2^n format (e.g. 64, 128, 256). We select 256 neurons for both hidden layers. With this neural network, we calculate prediction values and average error for our ensemble learner. We measure the performance of the prediction using the mean squared error (MSE) metric. For each house, we calculate its MSE value based on the deviation between predictions and actual values over its test data set. Then, for each algorithm, we calculate average MSE using Equation 5 where h denotes the index of the house and t denotes the instance of the time prediction over all houses.

$$\frac{1}{m}\sum_{h=1}^{m}(\frac{1}{n}\sum_{t=1}^{n}(Y_{h\,real}^{t} - Y_{h\,prediction}^{t})^2) \tag{5}$$

where $Y_{h\,real}^{t}$ is real electricity consumption, Y_{h}^{t} is our model's electricity consumption prediction at hour t and house h, m is the total number of houses in our test data and n is the total number of hours in our test data per house.

The following are MSE values with standard errors for our base ensemble learner and the individual prediction algorithms:

1) *Ensemble:* **0.00746±0.00052**
2) *LSTM:* 0.0195±0.00059
3) *TESLA:* 0.04954±0.0008
4) *Persistence:* 0.31369±0.01299
5) *ARIMA:* 0.34268±0.02825
6) *Holt-Winters:* 0.34511±0.029831

These results show that as a single prediction method LSTM outperforms other methods. However, LSTM's computational complexity is much higher than the other methods (see Section V-F). Our ensemble learner outperforms single prediction algorithms by a great margin. To demonstrate this statistically, we calculate the average and maximum improvement over single prediction algorithms, shown in Table I. Our ensemble learner enhances the performance of single prediction algorithms by up to 99.98%. Our base ensemble learner obtains up to 99.13% (72.84% on average) improvement against LSTM, the best individual prediction algorithm.

C. Pruned Ensemble Learner

We obtain the following average and maximum pruning ratios given 0.1 pruning robustness degree over all houses:

- Average Maximum Pruning Ratio (%): 27±0.41
- Highest Maximum Pruning Ratio (%): 45

By performing pruning, our aim is to decrease the number of parameters while keeping base ensemble model's average error (or obtain smaller error value). After we obtain maximum pruning ratios, we fine-tune the weights of the model. In 96.3% of the houses, we observe that the maximum pruning ratio is greater than zero, which clearly shows that some weights are indeed redundant. Among the houses where pruning is applied, we obtain smaller error values in 71% of the houses after pruning and fine-tuning. Overall, we have the following average and maximum improvement percentages over our base ensemble model using pruning:

- Average Improvement (%): 10.9±0.83
- Maximum Improvement (%): 67.88

D. Comparison with State-of-the-art Models

We compare our ensemble learners with an online expert selection methodology [35], and four state-of-the-art ensemble models: AdaBoost , gradient tree boosting, random forest regression, and voting regressor [36]. Over the entire set of houses, our ensemble learners outperform the state-of-the-art ensemble methods in 80% of the houses. We further analyze the improvement of our pruned ensemble model over the state-of-the-art ensemble models in more detail, as shown in Table II. Random forest regression's performance is the closest to our pruned ensemble learner, where our method is still 22% better (and up to 98%) on average.

TABLE II
PRUNED ENSEMBLE LEARNER'S AVERAGE AND MAXIMUM IMPROVEMENT OVER STATE-OF-THE-ART MODELS

Prediction Method	Average (%)	Maximum (%)
Online Expert Selection	72.45	98.70
AdaBoost	66.43	99.09
Voting	49.94	98.40
Gradient	49.87	98.24
Random Forest	22.11	98.30

E. Sensitivity Analysis

Next, we want to understand the impact of each individual method on the performance of our ensemble mechanism. We remove each method and rerun our model with the remaining input set, and compare the obtained performance against the individual methods as well as the full (base) ensemble model in Table III. In this table, each removed method column denotes an experiment where the corresponding method is removed from the original model and compared against the individual and the full ensemble methods. For example, the first column shows the experiment where LSTM is removed from the original input set, and the resulting ensemble model performs 35.8%, 23.92%, 78,8%, 89.26%, and 95.61% better as compared to individual LSTM, ARIMA, Tesla, HW, and Persistence algorithms, respectively; and 49.66% worse than the original full ensemble model. The other columns can be read similarly. We see that removing LSTM and TESLA makes our method perform worse (most important input algorithms), whereas removing the others improves the performance. Then, we conduct another experiment, where the input set consists of only LSTM and TESLA, as they are the most important inputs in the original set, shown as the last column of Table III. Interestingly, this case does not lead to better performance than the full ensemble model, with 40.25% worse values. This shows that having a diverse set of input algorithms is a key factor to obtain good performance with our ensemble model [37]. Even though individual algorithms may not seem important one by one, removing them all together does not make the performance better, but significantly worse.

F. Computational Complexity Analysis

LSTM, ARIMA, TESLA, Persistence, and Holt-Winters take 200.08, 130.38, 2.33, 1, and 0.13 seconds to complete per house, respectively. LSTM takes 100X more time than TESLA and 1500X more time than Holt-Winters. Even though LSTM is the best algorithm among these (with respect to average MSE), it is also the most computationally intensive one. For our ensemble learners (i.e. base and pruned versions) we demonstrate pruned ensemble learner's lightweight characteristic through reduced number of parameters. On average, we eliminate 17,694 parameters for each house from our base ensemble learner network, corresponding to a reduction of 1.7M parameters over all houses. Since our network is not extremely large (4-layer network), we do not observe training or inference time difference between base and pruned ensemble models, where the training time in both learners is approximately 20 seconds per house. Thus, we are adding only a 10% execution time increase over the LSTM method, but with a huge performance gain, i.e. 72% average and up to 99% maximum improvement in prediction performance.

VI. CONCLUSION

We propose our readers to create a lightweight ensemble learner for individual house level electricity consumption prediction. To reach this goal, we first implement five distinctive prediction algorithms: ARIMA, Holt-Winters, TESLA, LSTM and Persistence. Among these algorithms, LSTM performs best. Afterwards, we create our intial ensemble learner by combining prediction values of single prediction algorithms using 4-layer dense neural network. Initial ensemble learner improves the LSTM's performance on average by 72.84% and by up to 99.13%. Finally, we apply pruning to the weights of our ensemble network to eliminate insignificant weights. Doing this leads to 10.9% less error and 27% less number of parameters. All in all, we show that pruned ensemble model reaches the minimum error value.

ACKNOWLEDGEMENT

This work has been funded in part by San Diego State University Sustainable Energy Center. It also has been funded in part by NSF, with award numbers #1911095, #1826967, #1730158, and #1527034. It was also partially supported by SRC task #2805.001.

TABLE III
SENSITIVITY ANALYSIS OF OUR ENSEMBLE METHOD WHEN INDIVIDUAL METHODS ARE REMOVED

		Removed method					Only
		LSTM	**ARIMA**	**TESLA**	**HW**	**Persistence**	**LSTM+TESLA**
Compared against	**LSTM**	35.8%	73.61%	65.7%	73.17%	75.99%	54.83%
	ARIMA	23.92%	94.56%	94.96%	96.25%	96.67%	90.62%
	TESLA	78.8%	88.4%	84.35%	88.35%	89.58%	82.03%
	HW	89.26%	95.37%	93.37%	92.95%	95.63%	87.21%
	Persistence	95.61%	97.95%	97.22%	97.97%	97.78%	95.79%
	Full ensemble	-49.66%	21.17%	-20.16%	36.36%	30.63%	-40.25%

REFERENCES

[1] IEA, "Electricity demand by sector and scenario." https://www.iea.org/data-and-statistics/charts/electricity-demand-by-sector-and-scenario-2018-2040, 2018.

[2] L. Hernández, C. Baladron, J. M. Aguiar, B. Carro, A. Sanchez-Esguevillas, J. Lloret, D. Chinarro, J. J. Gomez-Sanz, and D. Cook, "A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 106–113, 2013.

[3] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International journal of forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[4] T. Hong, "Crystal ball lessons in predictive analytics," *EnergyBiz Mag*, vol. 12, no. 2, pp. 35–37, 2015.

[5] SDGE, "Reduce your use rewards." https://webarchive.sdge.com/residential/reduce-your-use/reduce-your-use-rewards, 2019. Accessed: February 2019.

[6] A. Rahman, V. Srikumar, and A. D. Smith, "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks," *Applied energy*, vol. 212, pp. 372–385, 2018.

[7] O. Güngör, B. Akşanlı, and R. Aydoğan, "Algorithm selection and combining multiple learners for residential energy prediction," *Future Generation Computer Systems*, vol. 99, pp. 391–400, 2019.

[8] K. Kasiviswanathan, J. He, K. Sudheer, and J.-H. Tay, "Potential application of wavelet neural network ensemble to forecast streamflow for flood management," *Journal of Hydrology*, vol. 536, pp. 161–173, 2016.

[9] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4820–4828, 2016.

[10] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, pp. 1135–1143, 2015.

[11] Eric Wilson, "Commercial and residential hourly load profiles for all tmy3 locations in the united states." https://openei.org/doe-opendata/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states, 2013.

[12] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[13] C. Chatfield, "The holt-winters forecasting procedure," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 27, no. 3, pp. 264–279, 1978.

[14] B. O. Akyurek, A. S. Akyurek, J. Kleissl, and T. Š. Rosing, "Tesla: Taylor expanded solar analog forecasting," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 127–132, IEEE, 2014.

[15] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[16] A. Zendehboudi, M. Baseer, and R. Saidur, "Application of support vector machine models for forecasting solar and wind energy resources: A review," *Journal of cleaner production*, vol. 199, pp. 272–285, 2018.

[17] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by lstm," *Energy*, vol. 148, pp. 461–468, 2018.

[18] S. Li and R. Li, "Comparison of forecasting energy consumption in shandong, china using the arima model, gm model, and arima-gm model," *Sustainability*, vol. 9, no. 7, p. 1181, 2017.

[19] A. Rahman and A. S. Ahmar, "Forecasting of primary energy consumption data in the united states: A comparison between arima and holter-winters models," in *AIP Conference Proceedings*, vol. 1885, p. 020163, AIP Publishing LLC, 2017.

[20] S. Papadopoulos and I. Karakatsanis, "Short-term electricity load forecasting using time series and ensemble learning methods," in *2015 IEEE Power and Energy Conference at Illinois (PECI)*, pp. 1–6, IEEE, 2015.

[21] M. H. Alobaidi, F. Chebana, and M. A. Meguid, "Robust ensemble learning framework for day-ahead forecasting of household based energy consumption," *Applied energy*, vol. 212, pp. 997–1012, 2018.

[22] M. Khairalla, X. Ning, N. AL-Jallad, and M. El-Faroug, "Short-term forecasting for energy consumption through stacking heterogeneous ensemble learning model," *Energies*, vol. 11, no. 6, p. 1605, 2018.

[23] M. Avand, S. Janizadeh, D. Tien Bui, V. H. Pham, P. T. T. Ngo, and V.-H. Nhu, "A tree-based intelligence ensemble approach for spatial prediction of potential groundwater," *International Journal of Digital Earth*, pp. 1–22, 2020.

[24] S. Shan, B. Cao, and Z. Wu, "Forecasting the short-term electricity consumption of building using a novel ensemble model," *IEEE Access*, vol. 7, pp. 88093–88106, 2019.

[25] J. Lee, J. Kim, and W. Ko, "Day-ahead electric load forecasting for the residential building with a small-size dataset based on a self-organizing map and a stacking ensemble learning method," *Applied Sciences*, vol. 9, no. 6, p. 1231, 2019.

[26] Y. Huang, Y. Yuan, H. Chen, J. Wang, Y. Guo, and T. Ahmad, "A novel energy demand prediction strategy for residential buildings based on ensemble learning," *Energy Procedia*, vol. 158, pp. 3411–3416, 2019.

[27] M. Al-Rakhami, A. Gumaei, A. Alsanad, A. Alamri, and M. M. Hassan, "An ensemble learning approach for accurate energy load prediction in residential buildings," *IEEE Access*, vol. 7, pp. 48328–48338, 2019.

[28] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

[29] V. Radu, K. Kaszyk, Y. Wen, J. Turner, J. Cano, E. J. Crowley, B. Franke, A. Storkey, and M. O'Boyle, "Performance aware convolutional neural network channel pruning for embedded gpus," 2019.

[30] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks," in *2016 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 002858–002865, IEEE, 2016.

[31] H. Van den Dool, "Methods in short-term climate prediction," *Empirical methods in short-term climate prediction*, pp. 121–157, 2007.

[32] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.

[33] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," *arXiv preprint arXiv:2003.03033*, 2020.

[34] J. Heaton, *Introduction to neural networks with Java*. Heaton Research, Inc., 2008.

[35] G. Dhiman and T. S. Rosing, "Dynamic power management using machine learning," in *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pp. 747–754, 2006.

[36] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[37] P. S. Mashhadi, S. Nowaczyk, and S. Pashami, "Stacked ensemble of recurrent neural networks for predicting turbocharger remaining useful life," *Applied Sciences*, vol. 10, no. 1, p. 69, 2020.