Short-Term Demand Prediction Using an Ensemble of Linearly-Constrained Estimators

Md. Zulfiquar Ali Bhotto, Richard Jones, Stephen Makonin, *Senior Member, IEEE,* and Ivan V. Bajić, *Senior Member, IEEE,*

***Abstract*—The benefits of forecasting power demand can bring increased stability to any power grid. Between optimizing the production and control of grid resources and interacting with energy markets, there is a strong motivation for generation, transmission, and distribution grid stakeholders to obtain accurate power demand prediction, which requires more sophisticated prediction methods. We introduce an ensemble of linear predictive nodes called the Ensemble Prediction Network (EPN), which optimizes demand prediction motivated by various microgrid considerations. EPN outputs a nonlinear combination of the individual predictions whose mixing weights are optimized in the least-squares sense. Using a large number of publicly available datasets, we show that on-the-whole, EPN provides substantial improvement relative to each individual predictor. Furthermore, we compare our method with a Long Short-Term Memory (LSTM) neural network and a multi-layer perceptron, and demonstrate the advantages of the proposed method.**

***Index Terms*—energy prediction, demand forecasting, ensemble learning, optimization, smart grid, microgrid**

I. Introduction

Policy and technological developments are slowly deincentivizing the more traditional grid architecture of centralized generating plants, and instead reinforcing the progressive adoption of *distributed energy resources* (DER). These DER include renewable energy sources such as photovoltaic systems, wind, and fuel cells. Also included in DER are sources such as gas turbines, microturbines, and storage. As the grid increasingly incorporates DER rather than centralized generation, the electrical distances between generation and load are decreased. This naturally leads to improved voltage profiles, decreased line losses and waste heat, as well as potentially delaying investment in major grid infrastructure upgrades [1]. However, this development precipitates the need for robust control systems and transparency of grid resources to operators. These respectively bring with them many new challenges, not the least of which is the intermittent, volatile behaviour of renewables.

One important and increasingly wide-spread approach to managing distributed resources and control points is to break the distribution system into self-contained *microgrids*. Microgrids are defined as discrete, controllable entities with respect to the grid, comprised of interconnected loads and distributed energy resources. Microgrids are modular by design in the sense that they can either participate in the overall grid in a connected mode, or can operate independently from the grid in an island mode [2]. One of the main objectives of a DER-based distribution grid is the concept of dynamic islanding, allowing coordination of the increasing number of discrete microgrids with the overall distribution system so as to accommodate increased penetration of renewables, improve reliability and resilience to faults, voltage sags, black-/brown-outs, as well as to optimize the participation with transactive energy markets [1]. Distribution companies typically purchase energy through one hour forward bilateral contracts with generation companies or through day-ahead bidding based on estimated demand. Over-consumption by distribution companies is compensated via pool markets at a real-time price, incurring penalty costs. Sale of excess electricity back to the pool market due to under-consumption relative to forecasted demand is also sub-optimal [3].

There is a clear motivation for distribution companies to know the exact amount of electricity to be exchanged between substations and the overall grid. Forecasting techniques provide potential for distributors to opt for short-term contracts in day-ahead forward markets and optimize their profits. Aside from market participation, the ability of grid operators and control systems to reliably predict demand allows optimized scheduling of grid resources, grid maintenance, security assessments, and many other operational benefits.

In this work, we approach short-term load forecasting (STLF) in microgrids using an ensemble of nine nodes we call the Ensemble Prediction Network (EPN). Each node, under unique linear constraints, provides an optimal estimate of the predicted demand in the least-squares sense. The final predicted output is then produced by a nonlinear combination of the individual node outputs. Section II provides a brief summary of existing approaches to STLF, while Section III describes the proposed method in greater detail. Finally, Section IV provides a summary of examined datasets and quantifies the performance of our proposed network relative to two benchmark approaches.

II. Background

Due to the small and modular nature of microgrids, as well as the generally large penetration of intermittent renewables, demand forecasting is significantly more challenging than for typical grids. Several dimensions of data are often used to estimate the production of renewables and to compensate for the increased volatility of the time-series data. For example, a total of 9 feature variables are used in [4], including

Research funded by NSERC Enagage Grant EGP543219-19 in Canada.
Special thanks to Elena Popovici (elena@awesense.com) and our industry partner Awesense Wireless Inc. for their valuable feedback.

meteorological and operational parameters. They approach STLF by an ensemble method, the output of which is a support vector regression of three model outputs: gradient boosted decision trees, an extreme learning machine, and a basic linear regression model. In [5], the short-term demand of a microgrid community was predicted by a long short-term memory (LSTM)-based recurrent neural network (RNN), which combined historical demand data with future weather conditions predicted by another LSTM network. By design, RNNs are adept at modelling dynamic temporal behaviour, and complex input dependencies can be captured by using LSTM units in RNNs, while avoiding the gradient stability issues of traditional RNNs. We use this method as a benchmark for comparison in Section IV.

Time-frequency domain approaches are also common in forecasting non-stationary time-series signals. For example, [6] combines the above two concepts in a self-recurrent wavelet neural network, in which a wavelet neural network (WNN) is modified with self-recurrent loops for short-term microgrid demand prediction. In traditional WNNs, the translations, b, and dilations, a, of a mother wavelet family (considered the activations of the wavelet node) are learned from the data, providing a set of wavelet basis functions. The output is then taken as the weighted sum of these basis functions and compared with the desired signal, usually in terms of mean squared error. In [6], a feedback term is added to the wavelet activations of the form $\psi_{i,j} = \psi\left(\frac{u_{i,j}-b_i}{a_i}\right)$, where ψ is the mother wavelet and $\psi_{i,j}$ is the ith basis function of dimensionality j. The feedback term is contained in $u_{i,j}$, given by $u_{i,j} = x_j + \psi_{i,j} z^{-1} \cdot \theta_{i,j}$ for input x_j, time delay z^{-1} and weight $\theta_{i,j}$. In this way, the basis functions retain some memory of previous activations determined by their weight, leveraging some of the desirable properties of recurrent neural networks. The authors applied their method to a university campus, showing improved forecasting relative to traditional wavelet networks (where $\theta_{i,j} = 0$ for all i, j).

Many other methods have been employed in STLF, including a hybrid approach combining empirical mode decomposition, particle swarm optimization, and adaptive network-based fuzzy inference systems [7]; minimum-variance-based combination of support vector machines and neural networks [8]; subsequencing the daily load profile via variational mode decomposition followed by prediction by summing over subsequence-specific support vector machines [9]; wavelet decomposition of the daily load as input to a second-order gray neural network [10]; principal component analysis for dimension reduction passed to an echo state network employing genetic algorithms [11]; convolutional neural networks trained on multi-channel 'images' consisting of converted load, temperature, and fuzzified load data [12]; and many others. A summary of additional recent work along with information on availability of source data and other useful details is assembled in Table I. Accuracy metrics used by the referenced works are listed in Table II. Note that MAPE is thought to be a less valid accuracy metric compared to RMSE due to its several undesirable properties: (1) it is nonsymmetric; (2) as $d_i \to \infty$ MAPE yields the same value for all $\widehat{d_i}$; (3) when d_i is small, MAPE may yield an unrealistically large value.

TABLE I: Summary of Power Demand Prediction Literature

Ref.	Method	Time	Node	Data/Code
[13]	DE+Filtering	1 DA	MG,G	NA/NA
[14]	Similar-day based input selection + WT + NN	1 DA	G	NA/NA
[15]	NARX + SVM + input selection	1 DA	G	A/NA
[16]	EMD + ELMK / EKF + PSO	7 DA	MG	NA/NA
[17]	WLSE+ARMA	1 DA	NG	NA/NA
[18]	SOM + k-means clustering + MLP	1-DA	MG	NA/NA
[19]	Similar-day based input selection + WT + BNN	1-7 DA	G	A/NA
[20]	WT + GA + FLNN	1-5 WDA	G	NA/NA
[21]	WT+ ELM + MABC	1 DA	G	A/NA
[22]	SSA + SVM + BPNN + CS	1 DA	G	NA/NA

Method: NN: Neural Network, DE: Differential Evolution, SOM: Self-Organizing Map, MLP: Multilayer Perceptron, FL: Fuzzy logic, CNN: Cascaded NN, BNN: Bayesian NN, NARX: Nonlinear Autoregressive Model Inputs, BP: Back propagation, SSA: Singular Spectral Analysis, WLSE: Weighted Least Square Error, G: Grid, SH: Single House, EMD: Empirical Mode Decomposition, ELM-K: Extreme Learning Machine Kernel, EKF: Extended Kalman Filter, PSO: Particle Swarm Optimization, CS: Cuckoo Search, GA: Genetic Algorithm, MABC: Modified Artificial Bee Colony; **Time:** DA: Day Ahead, WDA: Weekday Ahead; **Node:** MG: Microgrid, G: Grid, NG: Nanogrid; and, Links to **Data Link / Code:** NA: Not Available, A: Available.

Nomenclature

$\mathbf{a}_i$	Mixing vector
$\mathbf{D}_k$	Consumption matrix
$\mathbf{d}_k$	Consumption vector
$\mathbf{w}_k$	Combiner vector
$\mathcal{W}_2$	Modulator matrix
$\widehat{\mathbf{d}}_k$	Predicted consumption vector

III. Short-term Demand Prediction

Let $\mathbf{d}_k[i]$ denote the net energy consumed from a feeder at the i-th hour of the k-th day, for $i = 1, 2, \ldots, 24$ and $k = 1, 2, \ldots, N$. Our objective is to make day- or week-ahead prediction of the consumption $\mathbf{d}_k$ in that feeder. As described above, we propose a nonlinear combination of multiple variants of linear predictors, following the structure shown in Figure 1. The EPN learns by optimizing the linear combiner vector over the previous L days of known consumption, subject to unique optimization constraints. Each day, every node in the ENP network in Figure 1 generates a model $\mathbf{w}$ for prediction by uniquely minimizing the error in the training data $(\mathbf{d}_k, \mathbf{D}_k)$, where $\mathbf{D}_k$ is the consumption matrix of hourly data for the previous L days, and $\mathbf{d}_k$ is the most recent day of observed demand. This model $\mathbf{w}$ is then used to predict the next day of hourly consumption $\widehat{\mathbf{d}}_{k+1}$. Nodes $\mathbf{w}_1$ to $\mathbf{w}_9$ in the first hidden layer predict the net demand as per their own unique method detailed below. The node $\mathbf{a}$ of the second hidden layer combines all individual predictions made by the nodes of the first hidden layer. In the case of multichannel data (real power, reactive power, current, etc.), node $\mathbf{b}$ is intended to combine the predictions of the individual channels, where each channel will be respectively predicted before being fused by the output node. Since this work used single channel data

TABLE II: Metrics From Referenced Works

References	Metric	Inputs	Mean accuracy
[14], [15], [19]	MAPE=$\frac{1}{N}\sum_{i=1}^{N} \text{abs}\left(\frac{d_i - \widehat{d}_i}{d_i}\right)$	load, weather	[14], 0.013/m, [15], 0.023/d, [19], 0.438/m
[13], [18], [21]	MAPE	load	[13], 0.024/w, [18], 0.03/d, [21], 0.56/m
[16]	RMSE $=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \widehat{d}_i)^2}$	load, day, hr, week	15-190
[22], [20]	RMSE	load	[22], 123, [20], 49
[17]	RMSE	load, weather	

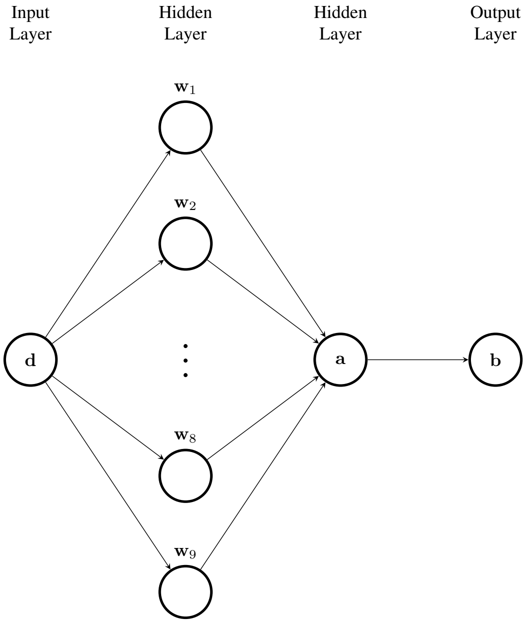


Fig. 1: Ensemble Prediction Network (EPN).

(apparent power), $\mathbf{b}$ is simply 1 in this case. In each of these nodes we will use steepest-descent recursions since they are simpler to analyze, have lower computational complexity, and higher robustness against data anomalies and ill-conditioning compared to Newton recursions [23]. These are important factors in real-time optimization. Each of these nodes will now be described in greater detail.

A. Unconstrained Prediction Node

This node is shown in Figure 1 as node w_1 in the first hidden layer. Let the consumption matrix $\mathbf{D}_k$ be obtained by stacking the consumption of the L most recent previous days as $\mathbf{D}_k = [\mathbf{d}_{k-1}, \mathbf{d}_{k-2}, \cdots, \mathbf{d}_{k-L}]$, where $\mathbf{d}_{k-1}^{\top} = [d_1 \; d_2 \; \cdots \; d_{24}]$ denotes the hourly consumption in the $(k-1)$-th day with $\top$ representing the transpose of a vector. In this node, we predict $\mathbf{d}_k$ as a linear combination of the columns of $\mathbf{D}_k$ in the minimum mean-square error sense. We consider the problem $\min_{\mathbf{w}_1} \; \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{D}_k\mathbf{w}_1\|^2\right]$, where $\mathbf{w}_1$ is the linear combiner vector. The solution is obtained by the recursive formula [24]:

$$\begin{aligned} \widehat{\mathbf{d}}_{1,k} &= \mathbf{D}_k\mathbf{w}_{1,k-1} && (1)\\ \mathbf{e}_k &= \mathbf{d}_k - \widehat{\mathbf{d}}_{1,k} \\ \mu &= \frac{\mathbf{e}_k^{\top}\mathbf{D}_k\mathbf{D}_k^{\top}\mathbf{e}_k}{\mathbf{e}_k^{\top}\mathbf{D}_k\mathbf{D}_k^{\top}\mathbf{D}_k\mathbf{D}_k^{\top}\mathbf{e}_k} \\ \mathbf{w}_{1,k} &= \mathbf{w}_{1,k-1} + \mu\mathbf{D}_k^{\top}\mathbf{e}_k && (2) \end{aligned}$$

for $k = 1, 2, \ldots, N$. The assumption of linear dependence is especially reasonable in the Macrogrid setting since consumption data from a feeder can be temporally correlated, as reported in [25].

B. Linearly Constrained Prediction Node

In this node (shown in Figure 1 as w_2 in the first hidden layer), we assume that in some selected time slots of the day, the net consumption from the feeder is day-invariant. This assumption might be reasonable for settings such as schools and universities, offices, hospitals, grocers, hotels, etc., where the energy consumption in the early morning or late at night remains day-invariant. In this node we consider the following day-invariant time slots: Late night, early morning, mid-day, and evening. The cost function in this node can be written as $\mathbb{E}\left[\|\mathbf{d}_k - \widehat{\mathbf{d}}_{2,k}\|^2\right]$ subject to:

$$\sum_{i=i_1}^{i_2} \widehat{\mathbf{d}}_{2,k}[i] = \delta_1$$

$$\sum_{i=i_3}^{i_4} \widehat{\mathbf{d}}_{2,k}[i] = \delta_2$$

$$\sum_{i=i_5}^{i_6} \widehat{\mathbf{d}}_{2,k}[i] = \delta_3$$

$$\sum_{i=i_7}^{i_8} \widehat{\mathbf{d}}_{2,k}[i] = \delta_4$$

where δ_1, δ_2, δ_3, and δ_4 are the day-invariant net consumptions in the selected time slots. The number of time slots and the length of each time slot vary as per the type of consumers. In the matrix-vector product format we can write the problem as

$$\min_{\mathbf{w}_2} \; \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{D}_k\mathbf{w}_2\|^2\right] \qquad (3)$$

subject to $\mathbf{B}_k\mathbf{w}_2 = \boldsymbol{\delta}$ where

$$\mathbf{B}_k = \begin{bmatrix}\mathbf{u}_1^{\top}\\ \mathbf{u}_2^{\top}\\ \mathbf{u}_3^{\top}\\ \mathbf{u}_4^{\top}\end{bmatrix}\mathbf{D}_k, \quad \boldsymbol{\delta} = \begin{bmatrix}\delta_1\\ \delta_2\\ \delta_3\\ \delta_4\end{bmatrix}$$

with $\mathbf{u}$ a zero vector of size $[24, 1]$, but with unity elements in the selected time slots. The solution of (3) yields the recursive formula [26]:

$$\widehat{\mathbf{d}}_{2,k} = \mathbf{D}_k\mathbf{w}_{2,k-1} \qquad (4)$$

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{d}_k - \widehat{\mathbf{d}}_{2,k} \\
\mathbf{e}_z &= \mathbf{d}_k - \mathbf{D}_k[\mathbf{Z}_k\mathbf{w}_{2,k-1} + \mathbf{c}_k] \\
\mu &= \frac{\mathbf{e}_k^\top \mathbf{D}_k \mathbf{Z}_k \mathbf{D}_k^\top \mathbf{e}_z}{\mathbf{e}_k^\top \mathbf{D}_k \mathbf{Z}_k \mathbf{D}_k^\top \mathbf{D}_k \mathbf{Z}_k \mathbf{D}_k^\top \mathbf{e}_k} \\
\mathbf{w}_{2,k} &= \mathbf{Z}_k \left[\mathbf{w}_{2,k-1} + \mu \mathbf{D}_k^\top \mathbf{e}_k\right] + \mathbf{c}_k
\end{aligned} \tag{5}$$

where

$$\begin{aligned}
\mathbf{Z}_k &= \mathbf{I} - \mathbf{B}_k^\top (\mathbf{B}_k \mathbf{B}_k^\top)^{-1} \mathbf{B}_k \\
\mathbf{c}_k &= \mathbf{B}_k^\top (\mathbf{B}_k \mathbf{B}_k^\top)^{-1} \boldsymbol{\delta}.
\end{aligned}$$

C. Hourly Load Variation Constrained Prediction Node

In this node (shown in Figure 1 as w_3 in the first hidden layer), we assume that the average hourly load variation of the demand in a feeder is a known constant. This assumption might reasonably hold, for example, in industrial plants where the production line remains running throughout the day. Hence we consider the problem

$$\min_{\mathbf{w}_3} \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{D}_k \mathbf{w}_3\|^2 + \gamma \|\mathbf{F}\mathbf{D}_k \mathbf{w}_3\|_1\right] \tag{6}$$

to make the day-ahead prediction, where $\gamma = \mathbb{E}[\|\mathbf{F}\mathbf{d}_k\|_1]$ is the known constant of the variation of $\mathbf{d}_k$ and

$$\mathbf{F} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

The solution of (6) can be obtained as

$$\widehat{\mathbf{d}}_{3,k} = \mathbf{D}_k \mathbf{w}_{3,k-1} \tag{7}$$

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{d}_{3,k} - \widehat{\mathbf{d}}_{3,k} \\
\mathbf{f}_d &= \mathbf{F}\widehat{\mathbf{d}}_{3,k} \\
\mathbf{y}_k &= \mathbf{F}^{-1} \left[\mathbf{diag}(\text{abs}(\mathbf{f}_d)) - \gamma\right]_+ \text{sign}(\mathbf{f}_d) \\
\mathbf{e}_z &= \mathbf{d}_{3,k} - \mathbf{y}_k \\
\mathbf{q}_k &= \mathbf{D}_k^\top \mathbf{e}_z \\
\mu &= \frac{\mathbf{e}_k^\top \mathbf{D}_k \mathbf{q}_k}{\|\mathbf{D}_k \mathbf{q}_k\|^2} \\
\mathbf{w}_{3,k} &= \mathbf{w}_{3,k-1} + \mu \mathbf{q}_k
\end{aligned} \tag{8}$$

Parameter γ becomes the known average hour-wise load variation in the feeder. In this case it is called hour-wise load variation constrained prediction (HLP).

D. Day-wise Load Variation Based Prediction Node

In this node (shown in Figure 1 as w_4 in the first hidden layer), we assume that the daily consumption has common low frequency components, which do not need to be predicted. This assumption reasonably holds in cases of consistent operation, where energy consumption patterns differ only in details from day to day. Such situations include hospitals, hotels, malls, restaurants, etc. For this purpose we let the daily difference in the consumption data to be regressed in the matrix $\mathbf{D}_{c,k}$ as

$$\mathbf{D}_{c,k} = \left[\mathbf{d}_{k-1} - \mathbf{d}_{k-2},\ \mathbf{d}_{k-2} - \mathbf{d}_{k-3},\ \cdots,\ \mathbf{d}_{k-L} - \mathbf{d}_{k-L+1}\right]$$

Our objective is to predict $\mathbf{d}_k$ by first predicting the details $\mathbf{d}_k - \mathbf{d}_{k-1}$ by using $\mathbf{D}_{c,k}$ and then adding $\mathbf{d}_{k-1}$. For this purpose we consider the following problem

$$\min_{\mathbf{w}_4} \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{d}_{k-1} - \mathbf{D}_{c,k}\mathbf{w}_4\|^2\right]. \tag{9}$$

One of the advantages of the cost function in this problem is that it removes the low frequency components from matrix $\mathbf{D}_{c,k}$, which improves the performance of adaptive filters. Its solution is obtained by the recursive formula

$$\begin{aligned}
\widehat{\mathbf{d}}_{c,k} &= \mathbf{D}_{c,k} \mathbf{w}_{4,k-1} \\
\widehat{\mathbf{d}}_{4,k} &= \widehat{\mathbf{d}}_{c,k} + \mathbf{d}_{k-1}
\end{aligned} \tag{10}$$

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{d}_k - \mathbf{d}_{k-1} - \widehat{\mathbf{d}}_{c,k} \\
\mu &= \frac{\mathbf{e}_k^\top \mathbf{D}_{c,k} \mathbf{D}_{c,k}^\top \mathbf{e}_k}{\mathbf{e}_k^\top \mathbf{D}_{c,k} \mathbf{D}_{c,k}^\top \mathbf{D}_{c,k} \mathbf{D}_{c,k}^\top \mathbf{e}_k} \\
\mathbf{w}_{4,k} &= \mathbf{w}_{4,k-1} + \mu \mathbf{D}_{c,k}^\top \mathbf{e}_k
\end{aligned} \tag{11}$$

for $k = 1, 2, \ldots, N$.

E. Robust Linear Prediction Node

The discrepancy between the available demands and the demand to be predicted can often be too high. In such cases, it would be better to consider those demands to be predicted as outliers and suppress their predictions so that they cannot influence the future predictions. The node handling such cases is shown in Figure 1 as w_5 in the first hidden layer. Situations like renovations, maintenance of power lines, fire safety training simulation in industrial plants, schools, offices, hospitals, and many other conceivable situations all give rise to outliers in the load consumption data. Since the L_1 norm-based cost functions are robust to outliers, the resulting problem becomes

$$\min_{\mathbf{w}_5} \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{D}_k \mathbf{w}_5\|_1\right] \tag{12}$$

and its solution results in the recursive formula

$$\widehat{\mathbf{d}}_{5,k} = \mathbf{D}_k \mathbf{w}_{5,k-1} \tag{13}$$

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{d}_k - \widehat{\mathbf{d}}_{5,k} \\
\mu &= \alpha \frac{\mathbf{e}_k^\top \mathbf{D}_k \mathbf{D}_k^\top \text{sign}(\mathbf{e}_k)}{\text{sign}(\mathbf{e}_k)^\top \mathbf{D}_k \mathbf{D}_k^\top \mathbf{D}_k \mathbf{D}_k^\top \text{sign}(\mathbf{e}_k)} \\
\mathbf{w}_{5,k} &= \mathbf{w}_{5,k-1} + \mu \mathbf{D}_k^\top \text{sign}(\mathbf{e}_k)
\end{aligned}$$

for $k = 1, 2, \ldots,$ and $0 < \alpha \ll 1$.

F. Robust DLV Linear Prediction Node

In this node (shown in Figure 1 as w_6 in the first hidden layer), we use the same error given in equation (9) but with the L_1 norm for the same reason given in the previous node. In addition, this cost function suppresses the variability between weekdays and weekends/holidays. The problem in this case becomes

$$\min_{\mathbf{w}_6} \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{d}_{k-1} - \mathbf{D}_{c,k}\mathbf{w}_6\|_1\right] \tag{14}$$

whose solution yields the recursive formula

$$\widehat{\mathbf{d}}_{c,k} = \mathbf{D}_{c,k} \mathbf{w}_{6,k-1}$$

$$\begin{aligned}\widehat{\mathbf{d}}_{6,k} &= \widehat{\mathbf{d}}_{c,k} + \mathbf{d}_{k-1} \\ \mathbf{e}_k &= \mathbf{d}_k - \mathbf{d}_{k-1} - \widehat{\mathbf{d}}_{c,k} \\ \mu &= \alpha \frac{\mathbf{e}_k^\top \mathbf{D}_{c,k} \mathbf{D}_{c,k}^\top \mathrm{sign}(\mathbf{e}_k)}{\mathrm{sign}(\mathbf{e}_k)^\top \mathbf{D}_{c,k} \mathbf{D}_{c,k}^\top \mathbf{D}_{c,k} \mathbf{D}_{c,k}^\top \mathrm{sign}(\mathbf{e}_k)} \\ \mathbf{w}_{6,k} &= \mathbf{w}_{6,k-1} + \mu \mathbf{D}_{c,k}^\top \mathrm{sign}(\mathbf{e}_k)\end{aligned} \tag{15}$$

for $k = 1, 2, \ldots,$ and $0 < \alpha \ll 1$.

G. MinMax Linear Prediction Node

In stochastic optimization, continuing to minimize errors that are already smaller than the optimal mean-square error introduces noise in the solution. This node is shown in Figure 1 as w_7 in the first hidden layer. This has been addressed in the literature by developing set-membership adaptive filter [27] that prevents updates whenever the *a priori* error becomes smaller than a pre-specified threshold. Based on these considerations, in this node we preferentially address the larger predictive errors and weight smaller errors less heavily. The problem in this case becomes

$$\min_{\mathbf{w}_7} \; \max_{\mathbf{c}} \; \mathbb{E}\left[(\mathbf{c}^\top \mathbf{d}_k - \mathbf{c}^\top \mathbf{D}_k \mathbf{w}_7)^2\right] \tag{16}$$

subject to

$$\mathbf{c} > \mathbf{0} \quad \text{and} \quad \mathbf{1}^\top \mathbf{c} = 1.$$

Its solution yields the recursive formula

$$\begin{aligned}\widehat{\mathbf{d}}_k &= \mathbf{D}_k \mathbf{w}_{7,k-1} \\ \mathbf{e}_k &= \mathbf{d}_k - \widehat{\mathbf{d}}_k \\ \varepsilon_a &= \mathbf{c}_{k-1}^\top \mathbf{e}_k \\ \mathbf{Z} &= \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{24} \\ \mathbf{q}_k &= \mathbf{Z}\mathbf{e}_k \varepsilon_a \\ \mu &= \frac{\min(\mathbf{c}_{k-1})}{\max(|\mathbf{q}_k|)} \\ \mathbf{c}_k &= \mathbf{c}_{k-1} + \mu \mathbf{q}_k \\ \mathbf{w}_{7,k} &= \mathbf{w}_{7,k-1} + \frac{1}{\mathbf{c}_{k-1}^\top \mathbf{D}_k \mathbf{D}_k^\top \mathbf{c}_{k-1}} \mathbf{D}_k^\top \mathbf{c}_{k-1} \varepsilon_a\end{aligned} \tag{17}$$

H. Kurtosis Based Prediction Node

The MSE cost function is optimal when the modelling error residual noise signal is Gaussian. However, when the noise signal is periodic, uniform, or uncertain, other cost functions should be considered. For these situations, the least mean Kurtosis cost function-based algorithm in [28] offers improved performance relative to the MSE. We reformulate the method in [28] and propose the following Kurtosis-based cost function for microgrid prediction:

$$\min_{\mathbf{w}_8} \; \max_{\mathcal{W}_1} \; \mathbb{E}\left[\mathbf{e}^\top (\mathcal{W}_1 - \mathbf{e}\mathbf{e}^\top)\mathbf{e}\right] \tag{18}$$

where $\mathbf{e} = \mathbf{d}_k - \mathbf{D}_k \mathbf{w}_8$. The recursion formula for $\mathbf{w}_8$ becomes

$$\begin{aligned}\widehat{\mathbf{d}}_{8,k} &= \mathbf{D}_k \mathbf{w}_{8,k-1} \\ \mathbf{e}_k &= \mathbf{d}_k - \widehat{\mathbf{d}}_{8,k} \\ \mathcal{W}_e &= 3\mathcal{W}_{1,k-1} - \mathbf{e}_k \mathbf{e}_k^\top \\ \mu &= \frac{\mathbf{e}_k^\top \mathbf{D}_k \mathbf{D}_k^\top \mathcal{W}_e \mathbf{e}_k}{\mathbf{e}_k^\top \mathcal{W}_e^\top \mathbf{D}_k \mathbf{D}_k^\top \mathbf{D}_k \mathbf{D}_k^\top \mathcal{W}_e \mathbf{e}_k} \\ \mathcal{W}_{1,k} &= \lambda \mathcal{W}_{1,k-1} + \beta \mathbf{e}_k \mathbf{e}_k^\top \\ \mathbf{w}_{8,k} &= \mathbf{w}_{8,k-1} + \mu \mathbf{D}_k^\top \mathcal{W}_e \mathbf{e}_k\end{aligned} \tag{19}$$

where the update formula of $\mathcal{W}_{1,k}$ computes the autocorrelation of the error signal with a leakage factor $\lambda \in (0,1)$ and step size β for each $k = 1, 2, \ldots, N$. Since $\mathcal{W}_{1,k}$ is the autocorrelation matrix, cost function (18) computes the Kurtosis of the error signal. As can be seen the recursion formula with a prespecified μ and with scalar error e becomes identical to the recursion formula in [28].

I. Alternating Minimization Based Linear Prediction Node

In this final node (shown in Figure 1 as w_9 in the first hidden layer), we attempt to modulate the linear predictor output in order to bring improvement in the straightforward linear predictions. We consider the problem

$$\min_{\mathbf{w}_9, \mathcal{W}_2} \; \mathbb{E}\left[\|\mathbf{d}_k - \mathcal{W}_2 \mathbf{D}_k \mathbf{w}_9\|^2\right] \tag{20}$$

where the filtering matrix $\mathcal{W}_2$ is the optimal modulator of the linear prediction $\mathbf{D}_k \mathbf{w}_9$, and $\mathbf{w}_9$ is the optimal linear combiner of the filtered data matrix $\mathcal{W}_2 \mathbf{D}_k$. If matrix $\mathcal{W}_2$ is a doubly stochastic matrix with binary elements, then it would only reshuffle the linear prediction $\mathbf{D}_k \mathbf{w}_9$ to move closer to $\mathbf{d}_k$. This optimization problem can be solved by the alternating minimization method:

$$\begin{aligned}\mathbf{y}_k &= \mathbf{D}_k \mathbf{w}_{9,k-1} \\ \widehat{\mathbf{d}}_{9,k} &= \mathcal{W}_{2,k-1} \mathbf{y}_k \\ \mathbf{e}_k &= \mathbf{d}_k - \widehat{\mathbf{d}}_{9,k} \\ \mathbf{q}_{1,k} &= \mathbf{D}_k^\top \mathcal{W}_{2,k-1}^\top \mathbf{e}_k \\ \mathbf{q}_{2,k} &= \mathbf{e}_k \mathbf{y}_k^\top \\ \mathbf{w}_{9,k} &= \mathbf{w}_{9,k-1} + \mu_1 \mathbf{q}_{1,k} \\ \mathcal{W}_{2,k} &= \mathcal{W}_{2,k-1} + \mu_2 \mathbf{q}_{2,k}\end{aligned} \tag{21}$$

J. Geometric Mean Combiner Node

The prediction accuracy of the individual predictors in (1), (4), (7), (10), (13), (15), (17), (19), and (21) clearly depend on how well respective assumptions are met by the microgrid demand (node $\mathbf{a}$ in the second hidden layer of Figure 1). Because of the inherent volatility of microgrid load consumption data, we should expect rogue patterns and hence we use all proposed variants of the linear predictors and combine them in the second hidden layer. In this layer we attempt to find parameter $\mathbf{a}_i$ for mixing the predictions in each layer 1 node by considering the cost function $\mathbb{E}\left(\mathbf{d}_k[i] - \prod_{l=1}^{9} \widehat{\mathbf{d}}_{l,k}[i]^{\mathbf{a}_i[l]}\right)^2$ where $\sum_{l=1}^{9} \mathbf{a}_i[l] = 1, \quad \mathbf{a}_i > \mathbf{0}$

for $i = 1, 2, \ldots, 24$. Minimizing this cost function with respect to $\mathbf{a}_i$ finds the weighted geometric mean of the first layer's predictions producing the lowest mean square error. Its solution with the initialization $\mathbf{a}_{i,0} = 1/9 \cdot \mathbf{1}$ can be obtained as

$$
\begin{aligned}
\mathbf{g}_i[l] &= \log\left(\widehat{\mathbf{d}}_{l,k}[i]\right) \quad \text{for} \quad l = 1,\ldots,9 \\
\mathbf{Z} &= \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{9} \\
\widehat{\mathbf{d}}_{g,k}[i] &= \prod_{l=1}^{9} \widehat{\mathbf{d}}_{l,k}[i]^{\mathbf{a}_{i,k-1}[l]} && (22) \\
e_i &= \log\left(\frac{\mathbf{d}_k[i]}{\widehat{\mathbf{d}}_{g,k}[i]}\right) \\
\mathbf{q}_i &= \mathbf{Z}\mathbf{g}_i \\
\mu &= \frac{\min(\mathbf{a}_{i,k-1})}{\max(|\mathbf{q}_i e_i|)} \\
\mathbf{a}_{i,k} &= \mathbf{a}_{i,k-1} + \mu \mathbf{q}_i e_i && (23)
\end{aligned}
$$

for $i = 1, 2, \ldots, 24$ and $k = 1, 2, \ldots, N$.

K. Output Layer

For a single data channel (e.g., apparent power), the second hidden layer in Figure 1 would be the output layer. Upon the availability of multitype data (e.g., real power, reactive power, etc.), the output layer in Figure 1 will perform fusion in a way similar to a multitasking adaptive filter [29] — i.e., an ensemble learner.

IV. Numerical Results

We consider the hourly power consumption data given in [31], [32], [12], [33], [34], [35], and [30] to test the proposed network and we compare it with the long short-term memory (LSTM), multilayer perceptron (MLP), support vector regression (SVR), and ensemble tree regression (ETR) networks since they are the most popular global structures for time series prediction problem using neural network in engineering and nonengineering applications. The basic properties of all the datasets used are given in Table IV (*see Supporting Document*). We denote the individual node prediction results from equations (1), (4), (7), (10), (13), (15), (17), (19), and (21) as follows: unconstrained prediction (UP), linearly constrained prediction (LCP), hourly load variation constrained prediction (HLP), day-wise load variation constrained prediction (DLP), Robust prediction (RbP), Robust DLV prediction (RdP), MinMax prediction (MMP), Kurtosis based prediction (KbP), and alternating minimization prediction (ALM), respectively. The ground truth demand is denoted as Gt. The combined prediction in equation (22) is denoted as the geometric-mean adaptive combiner (GMC) prediction. We used $L = 30$ previous days in all the networks. Both the LSTM and MLP network have two hidden layers with 50 neurons in each layer to perform day-ahead prediction. All weights were initialized as $\mathbf{w}_{\cdot,0} = \mathbf{0}$, $\mathbf{c}_0 = 1/24 \cdot \mathbf{1}$, $\mathcal{W}_{1,0} = 10^{-16}\mathbf{I}$, and $\mathcal{W}_{2,0} = \mathbf{I}$. We plot the normalized square error curve defined as $\mathrm{SE}_t = 10\log_{10}\left[\frac{(d_t-\widehat{d}_t)^2}{d_{max}^2}\right]$ for $t = 1, 2, \ldots, 24 \times N$ in order to illustrate the performance obtained by the UP, LCP, HLP, DLP, RbP, RdP, MMP, KbP, ALM, and GMC predictors while skipping the plots of the LSTM, MLP, SVR, ETR predictors in Figure 3 for clear visibility for the dataset in [12]. It can be seen from Figure 3 that the GMC predictor achieves the best predictive error quite often. Out of the 8760 predictions shown, the relative percentage of predictions for which each method performed the best are as follows: UP=5.2%, LCP=10.6%, HLP=4.6%, DLP=11.7%, RbP=3.3%, RdP=12.1%, MMP=3.0%, KbP=5.6%, ALM=6.5%, GMC=9.3%, LSTM=5.0%, MLP=9.0%, SVR = 4.2%, ETR = 10.0%.

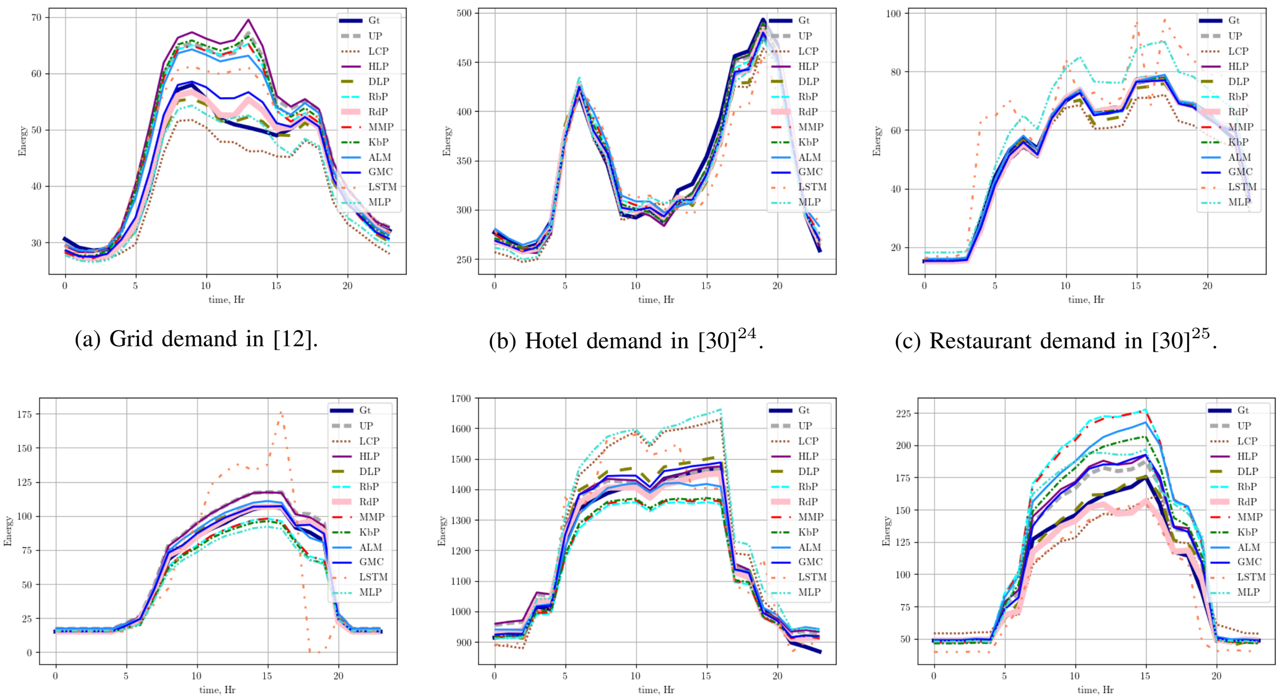
In addition we use the root mean square error (RMSE) defined as

$$
\mathrm{RMSE} = \sqrt{\frac{1}{24 \times (N-L-1)+1} \sum_{t=24\times(L+1)}^{24\times N} (d_t - \widehat{d}_t)^2}
$$

to compare the prediction accuracy obtained by the UP, LCP, HLP, DLP, RbP, RdP, MMP, KbP, ALM, GMC, and the LSTM, MLP, SVR, ETR predictors for all the datasets as given in Table V (*see Supporting Document*). Row 3 of Table V confirms the GMC predictor as a useful contribution to STLF, yielding an RMSE of 4.52 kW, which performs just slightly worse overall than the best RMSE (4.27 kW) achieved by the RdP predictor alone. The evolution of the mixing parameter (23) in the GMC prediction of Figure 3 is illustrated Figure 4. It can be seen from Figure 4 that the mixing parameter corresponding to the RdP and DLP predictions reached the highest and second highest values, respectively, since the RdP and DLP predictions yield the smallest and second smallest RMSE, respectively in Row 3 of Table V.

Examples of selected ground truth (Gt) demand from some feeders of a randomly selected day and the corresponding predicted demand obtained by the UP, LCP, HLP, DLP, RbP, RdP, MMP, KbP, ALM, GMC, and the benchmark LSTM/MLP predictors are illustrated in Figure 2. The RMSE and MAPE values of the plots in Figure 2 are given in Table III. As can be seen from Figures 2b, 2d, 2f, load demand in early morning, noon, and late night is constant. As a consequence we can see from Table V (*see Supporting Document*) with row index $[30]^{24}$, $[30]^{26}$, $[30]^{30}$, the LCP predictor yields lower RMSE than the UP predictor. The same can also be seen from Table III. Due to abrupt step changes in the consumption data, the HLP predictor could not yield better RMSE than the UP predictor in the respective examples as seen in Table V. It can also be seen from Figures 2a, 2c, 2d, 2e that some predictors overestimate while some others underestimate the Gt curve, whereas the GMC predictor by design takes their geometric mean, making its prediction closer to the Gt. Since the DLP predictor whitens the input data and the adaptive filter yields lower MSE with white input, the DLP predictor yields lower RMSE than the UP predictor in most cases as seen in Table V. The KbP predictor in Table V on the other hand yields slightly better RMSE as compared to the UP predictor. For developing countries where power generation capacity is much lower than the demand, such slight improvement can produce a positive impact in the operation of the power grid.

The RbP predictor yields much lower RMSE than the UP predictor in Table V (*see Supporting Document*), since a single



(a) Grid demand in [12].

(b) Hotel demand in [30][24].

(c) Restaurant demand in [30][25].

(d) Mall demand in [30][26].

(e) Hospital demand in [30][29].

(f) School demand in [30][30].

Fig. 2: Predicted demand for a randomly chosen day.

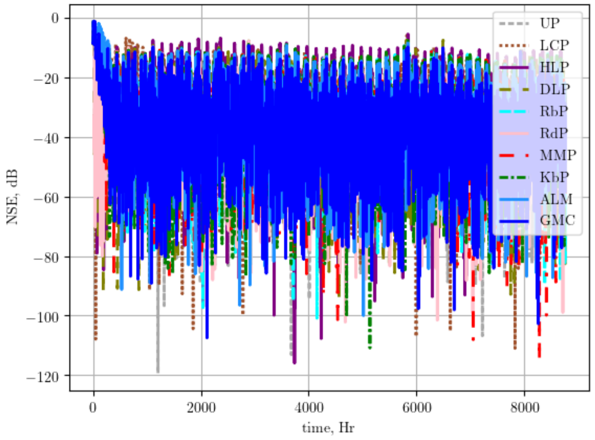


Fig. 3: An illustration of test performance by different predictors using normalized square error.

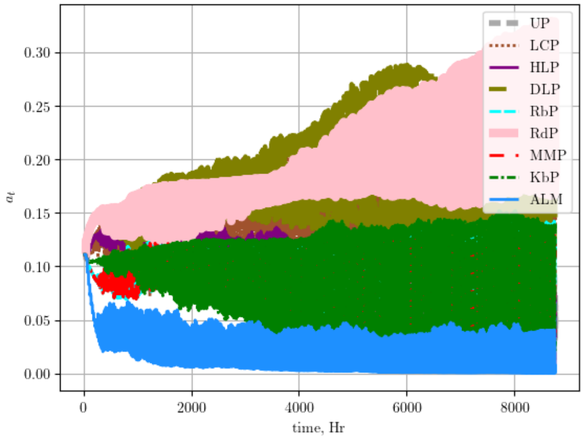


Fig. 4: Mixing parameter $\mathbf{a}$ in the GMC prediction of Figure 3.

outlier in the data would deviate the UP predictor causing larger error and would also affect all subsequent predictions. This would further aggregate the situation, which has been prevented by RbP predictor due to its L_1 norm based cost function. Likewise, as seen in Table V, the RdP predictor yields even better prediction than the RbP predictor since it whitens the input data by removing its correlated part. The MMP and ALM predictors offer similar RMSE in Table V. The MMP predictor puts less weight on the smaller errors (instead of zero weight seen in the set-membership filters), which causes it to offer improved performance over the UP predictor as seen in Table V. The KbP predictor offers similar performance as the MMP predictor since the cost functions of both predictors have a similar MinMax term. In most cases, the ALM predictor offers the second smallest and in some cases the smallest RMSE in Table V by virtue of its optimized $\mathcal{W}_2$, which attempts to reshuffle the linear prediction. The purpose of the GMC predictor was to follow the best performing predictors, which in this case were the RbP, RdP, and ALM predictors. It can be seen from Table V that the GMC predictor has been able to follow them successfully. In some cases

TABLE III: RMSE of Fig. 2

Figs.	2a	2b	2c	2d	2e	2f
UP	7.20	8.49	1.99	8.52	31.60	13.62
LCP	3.88	17.48	4.20	3.66	105.98	11.97
HLP	8.45	8.54	2.01	8.04	35.54	16.66
DLP	1.38	14.33	1.90	3.06	38.01	6.22
RbP	6.72	11.36	1.48	6.54	56.85	38.67
RdP	1.59	10.65	1.64	2.52	19.75	8.06
MMP	6.67	12.50	1.17	6.54	51.94	37.99
KbP	7.29	8.00	1.17	7.81	48.54	23.88
ALM	5.96	15.13	1.02	4.27	30.58	30.84
GMC	2.46	10.74	1.31	2.57	27.18	15.52
LSTM	5.15	28.28	15.47	32.93	90.50	14.03
MLP	2.60	15.59	8.66	9.64	125.35	25.64
SVR	11.66	71.78	12.63	25.64	226.27	53.04
ETR	8.17	16.31	3.12	5.85	66.98	21.78

MAPE of Fig. 2

Figs.	2a	2b	2c	2d	2e	2f
UP	0.114	0.023	0.030	0.141	0.024	0.095
LCP	0.086	0.042	0.058	0.041	0.067	0.112
HLP	0.133	0.023	0.030	0.136	0.027	0.109
DLP	0.031	0.030	0.026	0.032	0.025	0.041
RbP	0.107	0.028	0.027	0.062	0.037	0.229
RdP	0.027	0.026	0.024	0.022	0.016	0.038
MMP	0.106	0.031	0.021	0.062	0.033	0.231
KbP	0.114	0.021	0.021	0.074	0.031	0.155
ALM	0.093	0.038	0.022	0.101	0.021	0.191
GMC	0.044	0.027	0.020	0.037	0.018	0.094
LSTM	0.087	0.062	0.246	0.233	0.056	0.138
MLP	0.057	0.039	0.151	0.095	0.083	0.159
SVR	0.272	0.178	0.221	0.687	0.189	0.733
ETR	0.138	0.038	0.045	0.065	0.039	0.130

the GMC offers the smallest RMSE in Table V. All of these discussions also hold true for the estimation accuracy (EAC) [36] and MAPE performance measure metrics shown in Tables-VI and VII, respectively and also for the Table III.

In short, the RMSE in Table V (*see Supporting Document*) shows that RdP, ALM, and GMC are always among the best performing predictors. All the variants of the UP in Table V can offer better performance than the basic UP whenever the associated assumptions hold true. In all cases in Table V, GMC is better than the MLP, LSTM, SVR, and ETR networks, while the MLP network offers better prediction over the LSTM network. We now investigate the impact of the length of the time slots in one type of grid by increasing and decreasing the length of the time slots by 2. The RMSE results are shown in Table VIII. As can be seen in the table, the performance of the LCP node, as well as the EPN network, varies only slightly with slight changes in the length of time slots.

In order to further assess the effect of LCP node (i.e., time slot information) on the EPN network, we performed an experiment where the LCP node (which is the only node using time slot information) is removed. When tested on the cases from Table V, the EPN network without the LCP node performed equal to the EPN network with the LCP node in 11 cases, better in 6 cases, and worse in 33 cases. However, in all cases where performance degraded, RMSE was still lower than the RMSE produced by the LSTM, MLP, SVR, and ETR networks in Table V. Hence, excluding the LCP node (i.e., the time slot information) from the EPN network does not affect its superiority with respect to the other networks.

The CPU runtime per iteration (day) for a network is computed by first taking total CPU time of Table V and then dividing it by the number of datasets in Table V and the number of iterations (days) in one dataset. The CPU runtime per sample obtained in Python 3.6 run on an Intel(R) Core(TM) i7-4770 CPU @ 3.40GHZ 3.40 GHZ processor with 16.0 GB RAM and 64 bit operating system for the EPN, LSTM, MLP, SVR, and ETR networks are found to be 0.7 μs, 10.98 ms, 0.44 ms, 0.15 μs, and 1.5 ms respectively. As can be seen, the EPN network has much less computational load.

TABLE IV: Dataset Metadata Summary (in Watts)

Data	Place	Maximum	Minimum	Average
[31]	Vancouver (CDN)	40.38k	6.33k	17.10k
[32]	Zone 1 (USA)	44.87k	8.69k	17.73k
[12]	Johor (Malaysia)	75.45k	19.01k	43.93k
[33]	Building (USA)	1852.00k	1.00	267.23k
[34]	London (UK)	23.72G	9.02G	13.85G
[35]	Europe	9.36G	3.84G	6.57G
[37]^1	Ontario (CDN)	23.24G	10.54G	15.70G
[37]^2	Ontario (CDN)	1.69G	0.27G	0.91G
[37]^3	Toronto (CDN)	9.43G	3.63G	5.80G
[37]^4	Essa	1.94G	0.50G	0.98G
[37]^5	Southwest	7.34G	2.00G	3.18G
[30]^1	Restaurant (AK)	48.39k	12.85k	33.44k
[30]^2	Hospital (AK)	1324.78k	445.26k	904.91k
[30]^3	Hotel (AK)	408.78k	99.14k	248.31k
[30]^4	Apartment (AK)	44.47k	14.40k	23.98k
[30]^5	Supermarket (AK)	308.42k	72.35k	181.65k
[30]^6	StripMall (AK)	61.40k	3.17k	31.06k
[30]^7	RetailStore (AK)	117.97k	3.75k	40.57k
[30]^8	Office (CA)	1449.07k	211.11k	643.82k
[30]^9	School (CA)	283.30k	39.94k	94.86k
[30]^{10}	School (CA)	930.44k	86.93k	291.63k
[30]^{11}	Restaurant (CA)	71.39k	14.62k	35.67k
[30]^{12}	Hospital (CA)	1373.88k	447.39k	960.76k
[30]^{13}	Hotel (CA)	418.77k	106.16k	263.62k
[30]^{14}	Hotel (CA)	126.74k	32.13k	63.70k
[30]^{15}	Supermarket (CA)	349.88k	76.46k	178.63k
[30]^{16}	Apartment (CA)	66.08k	14.72k	25.56k
[30]^{17}	StripMall (CA)	87.57k	3.17k	31.78k
[30]^{18}	RetailStore (CA)	98.26k	3.71k	34.97k
[30]^{19}	School (AZ)	361.09k	40.00k	106.34k
[30]^{20}	School (AZ)	1383.57k	87.41k	531.71k
[30]^{21}	Office (AZ)	1960.61k	211.11k	837.67k
[30]^{22}	Apartment (AZ)	84.13k	14.71k	33.53k
[30]^{23}	Hospital (AZ)	1582.78k	745.45k	1149.81k
[30]^{24}	Hotel (AZ)	495.22k	110.36k	304.50k
[30]^{25}	Restaurant (AZ)	77.71k	14.65k	41.16k
[30]^{26}	StripMall (AZ)	115.65k	3.17k	39.30k
[30]^{27}	SuperMarket (AZ)	423.67k	75.10k	202.73k
[30]^{28}	RetailStore (AZ)	135.81k	3.71k	45.26k
[30]^{29}	Hospital (AR)	1558.71k	474.94k	1138.97k
[30]^{30}	School (AR)	361.09k	40.00k	106.34k
[38]^1	AEP (US)	24.02G	9.82G	15.52G
[38]^2	COMED (US)	23.75G	7.5G	11.65G
[38]^3	DAYTON (US)	3.72G	1.20G	2.10G
[38]^4	DEOK (US)	5.45G	1.87G	3.07G
[38]^5	DOM (US)	18.92G	6.24G	10.50G
[38]^6	DUQ (US)	2.89G	1.05G	1.69G
[38]^7	EKPC (US)	2.51G	0.51G	1.42G
[38]^8	FE (US)	14.03G	0.0G	7.75G
[38]^9	NI (US)	20.12G	7.19G	11.33G

TABLE V: Testing Results Using RMSE Where EPN is Compared to LSTM, MLP, SVR, and ETR

Dataset	Linear Predictors Used in EPN (Nodes $\mathbf{w}_1 \ldots \mathbf{w}_9$)									**EPN**	Compared To:			
	UP	**LCP**	**HLP**	**DLP**	**RbP**	**RdP**	**MMP**	**KbP**	**ALM**	**(GMC)**	LSTM	MLP	SVR	ETR
[31]	3.33	3.44	4.02	3.34	2.92	3.08	2.93	2.92	2.91	2.91	3.13	3.17	5.37	3.86
[32]	3.14	4.13	3.21	3.04	2.65	2.62	2.67	2.67	2.65	2.52	2.74	3.49	4.12	2.67
[12]	7.52	5.64	8.05	4.79	6.48	4.27	6.47	6.48	5.96	4.52	7.30	6.19	14.42	7.12
[33]	84.92	130.18	90.12	90.38	82.38	84.92	85.81	86.32	81.60	86.94	155.70	111.02	101.09	100.26
[34]	1.45	2.30	1.47	1.28	1.34	1.12	1.32	1.35	1.26	1.08	1.22	2.07	1.71	1.32
[35]	0.93	1.38	0.96	0.49	0.70	0.42	0.72	0.70	0.69	0.48	0.73	0.68	0.86	0.92
[37]^{1}	1.38	1.95	1.39	1.13	1.21	1.01	1.19	1.21	1.17	0.98	1.25	1.55	1.78	1.29
[37]^{2}	0.14	0.17	0.14	0.13	0.11	0.11	0.11	0.11	0.11	0.11	0.12	0.13	0.11	0.71
[37]^{3}	0.61	0.84	0.61	0.49	0.54	0.41	0.52	0.54	0.51	0.42	0.52	0.97	0.60	0.76
[37]^{4}	0.12	0.20	0.12	0.11	0.10	0.10	0.10	0.12	0.10	0.09	0.11	0.18	0.10	0.68
[37]^{5}	0.32	0.40	0.32	0.26	0.27	0.23	0.27	0.27	0.27	0.22	0.28	0.38	0.28	0.63
[30]^{1}	0.65	1.16	0.65	2.48	0.62	0.60	0.63	0.64	0.78	0.83	389.17	1.71	3.50	0.96
[30]^{2}	160.18	179.31	166.87	93.28	125.22	82.93	124.02	122.07	117.59	91.16	208.33	114.84	219.51	143.05
[30]^{3}	14.73	22.06	16.19	13.59	14.77	12.08	14.87	14.85	14.95	12.36	42.84	21.63	78.00	17.18
[30]^{4}	1.09	1.42	1.09	1.10	0.93	0.80	0.93	1.09	0.97	0.83	1.13	1.99	5.07	1.21
[30]^{5}	29.53	22.27	31.01	12.24	20.87	12.01	19.91	18.71	18.78	13.75	26.10	22.21	68.98	23.60
[30]^{6}	7.56	6.61	8.01	4.65	6.60	4.18	6.54	6.49	5.84	4.80	9.10	5.83	14.21	8.66
[30]^{7}	11.65	8.29	12.16	8.15	10.04	7.45	10.12	10.05	9.42	7.80	63.75	8.21	21.36	14.08
[30]^{8}	231.21	196.07	249.69	160.87	203.03	154.38	194.44	213.25	167.38	172.70	596.54	173.05	414.74	275.06
[30]^{9}	34.62	27.47	36.90	26.59	30.39	24.34	33.48	34.51	25.22	26.52	141.50	26.63	53.61	50.63
[30]^{10}	123.32	110.04	129.46	104.21	109.60	97.71	105.11	131.38	99.15	100.55	501.50	102.28	219.03	179.74
[30]^{11}	1.94	3.28	1.95	2.80	1.74	1.64	1.81	1.90	1.86	1.73	308.21	3.76	5.50	2.01
[30]^{12}	159.70	122.22	165.79	86.48	125.13	79.41	124.63	125.15	121.42	81.27	604.34	106.30	230.53	146.92
[30]^{13}	16.73	17.40	17.25	15.70	15.61	14.16	15.73	15.51	15.81	13.76	174.84	22.89	76.97	18.38
[30]^{14}	4.70	5.64	5.02	4.39	4.48	3.69	4.80	4.63	4.64	3.78	20.06	7.79	20.48	5.50
[30]^{15}	30.72	21.62	32.50	14.90	20.36	13.83	20.28	24.42	20.42	15.37	47.29	25.86	72.43	25.22
[30]^{16}	1.97	2.05	1.98	2.11	1.89	1.66	1.75	1.89	1.89	1.63	203.65	3.77	7.82	2.07
[30]^{17}	7.92	5.22	8.37	5.18	7.07	4.73	7.07	7.04	6.40	5.12	20.51	6.58	14.42	9.96
[30]^{18}	8.93	5.97	9.40	5.92	8.02	5.41	8.03	7.94	7.16	5.84	159.50	7.57	17.22	11.44
[30]^{19}	44.40	38.26	47.03	37.42	38.29	33.11	39.18	44.23	35.03	36.17	607.37	38.09	70.23	62.85
[30]^{20}	193.06	338.23	209.45	159.63	177.41	154.32	164.13	188.35	158.78	159.79	629.35	172.39	360.54	259.27
[30]^{21}	268.89	562.27	295.65	184.00	234.89	179.62	234.18	237.52	198.63	200.08	532.29	202.50	460.77	305.45
[30]^{22}	4.20	7.16	4.22	4.09	3.28	3.17	3.23	3.44	3.36	3.14	422.44	5.03	12.49	3.72
[30]^{23}	168.90	112.86	174.89	89.34	123.25	78.62	123.40	124.68	122.55	85.80	439.61	114.04	220.00	143.79
[30]^{24}	21.64	20.83	22.05	20.76	18.81	17.68	18.76	19.56	19.20	17.58	53.48	29.22	76.47	22.72
[30]^{25}	3.31	3.58	3.32	3.14	2.68	2.55	2.70	2.69	2.78	2.54	246.62	5.44	9.26	3.12
[30]^{26}	10.41	7.45	11.03	7.29	9.45	6.48	9.46	9.48	8.62	7.00	333.17	10.26	19.75	13.02
[30]^{27}	35.36	34.73	36.91	18.04	24.19	16.71	23.00	28.95	23.38	17.79	613.07	24.15	83.45	27.85
[30]^{28}	12.10	8.77	12.88	8.21	11.08	7.53	11.42	11.28	10.21	8.24	110.77	10.82	24.12	15.48
[30]^{29}	168.57	139.52	174.35	108.27	134.62	95.42	134.09	135.20	132.53	96.46	323.66	131.78	227.04	149.10
[30]^{30}	44.40	38.26	47.03	37.42	38.29	32.93	40.05	42.11	35.03	36.17	145.66	40.05	70.23	62.85
[38]^{1}	1.56	2.44	1.57	1.10	1.33	0.97	1.34	1.34	1.30	0.97	1.40	1.88	2.01	1.43
[38]^{2}	1.60	2.61	1.62	1.38	1.40	1.18	1.42	1.42	1.36	1.12	1.43	2.00	1.81	1.53
[38]^{3}	0.29	0.46	0.29	0.21	0.25	0.18	0.25	0.26	0.24	0.18	0.27	0.33	0.21	0.62
[38]^{4}	0.35	0.57	0.36	0.30	0.31	0.27	0.31	0.39	0.29	0.25	0.30	0.49	0.30	0.64
[38]^{5}	1.08	2.13	1.09	1.00	0.98	0.86	0.99	1.30	0.93	0.84	0.97	1.50	1.61	1.13
[38]^{6}	0.18	0.32	0.17	0.14	0.15	0.12	0.15	0.20	0.14	0.12	0.15	0.24	0.15	0.61
[38]^{7}	0.19	0. 30	0.19	0.22	0.17	0.17	0.18	0.36	0.17	0.16	0.18	0.29	0.16	0.54
[38]^{8}	0.91	1.40	0.92	0.70	0.81	0.64	0.82	1.20	0.78	0.62	0.83	1.03	1.03	1.00
[38]^{9}	1.62	2.03	1.65	1.26	1.45	1.08	1.48	1.55	1.38	1.10	1.38	1.84	1.84	1.53

Note: EPN has lower error scores in all tests as compared to LSTM, MLP, SVR, and ETR. Yellow highlighted values represent output with minimal error of algorithms (nodes $\mathbf{w}_1 \ldots \mathbf{w}_9$) in the first hidden layer. Green highlighted values show that the error score in the final EPN node is less than or equal to the error scores of the nine algorithms in the first hidden layer.

V. Conclusions

We have presented a novel method called the Ensemble Prediction Network (EPN) to perform accurate short-term prediction in microgrids composed of an ensemble of nine linear predictive nodes creating an ensemble learner. Each node is a unique linearly-constrained estimator, which provides for an optimal estimate of the predicted demand in the least-squares sense under certain constraints. The error measurement on a wide variety of data shows that our EPN, using a sensor fusion method, outperforms (in the sense of RMSE) LSTM, MLP, SVR, and ETR networks that are used as benchmarks for prediction using hourly data. Moreover, being composed of linear predictors with explicit constraints, the proposed EPN is more interpretable than LSTM and MLP, and is able to offer deeper insight into demand prediction. Future work could improve on the capability of the EPN to adapt quickly to abrupt changes in microgrid demand signatures. As discussed in Section II, volatile and rapidly changing load profiles

TABLE VI: Testing Results Using EAC Where EPN is Compared to LSTM, MLP, SVR, and ETR

Dataset	Linear Predictors Used in EPN (Nodes $\mathbf{w}_1 \ldots \mathbf{w}_9$)									**EPN**	Compared To:			
	UP	**LCP**	**HLP**	**DLP**	**RbP**	**RdP**	**MMP**	**KbP**	**ALM**	**(GMC)**	LSTM	MLP	SVR	ETR
[31]	0.927	0.925	0.912	0.927	0.937	0.933	0.936	0.937	0.937	0.937	0.931	0.930	0.872	0.915
[32]	0.937	0.914	0.936	0.938	0.945	0.947	0.945	0.945	0.945	0.951	0.941	0.925	0.905	0.944
[12]	0.942	0.956	0.938	0.966	0.951	0.970	0.951	0.952	0.954	0.968	0.942	0.953	0.860	0.953
[33]	0.895	0.828	0.887	0.893	0.900	0.905	0.896	0.899	0.902	0.902	0.818	0.870	0.862	0.884
[34]	0.961	0.939	0.961	0.967	0.965	0.970	0.966	0.964	0.967	0.971	0.967	0.943	0.952	0.964
[35]	0.950	0.914	0.949	0.977	0.957	0.980	0.959	0.957	0.961	0.975	0.957	0.961	0.950	0.946
[37]^{1}	0.968	0.953	0.968	0.974	0.971	0.977	0.972	0.971	0.973	0.978	0.971	0.964	0.955	0.969
[37]^{2}	0.944	0.927	0.943	0.948	0.955	0.956	0.956	0.955	0.954	0.957	0.947	0.942	0.950	0.648
[37]^{3}	0.963	0.948	0.962	0.971	0.968	0.975	0.967	0.965	0.969	0.975	0.967	0.943	0.963	0.946
[37]^{4}	0.954	0.924	0.952	0.957	0.960	0.961	0.961	0.955	0.960	0.964	0.956	0.932	0.956	0.686
[37]^{5}	0.963	0.953	0.962	0.972	0.967	0.974	0.969	0.967	0.970	0.975	0.967	0.955	0.971	0.917
[30]^{1}	0.997	0.990	0.997	0.989	0.997	0.998	0.997	0.997	0.996	0.996	0.000	0.988	0.976	0.991
[30]^{2}	0.938	0.932	0.936	0.968	0.951	0.972	0.951	0.950	0.953	0.966	0.921	0.957	0.898	0.951
[30]^{3}	0.981	0.967	0.979	0.982	0.980	0.984	0.980	0.980	0.979	0.984	0.958	0.969	0.866	0.978
[30]^{4}	0.985	0.980	0.985	0.986	0.987	0.989	0.987	0.984	0.986	0.989	0.984	0.970	0.916	0.982
[30]^{5}	0.948	0.956	0.946	0.980	0.965	0.981	0.965	0.966	0.965	0.976	0.950	0.957	0.827	0.962
[30]^{6}	0.930	0.937	0.925	0.967	0.936	0.971	0.935	0.936	0.944	0.962	0.895	0.945	0.816	0.929
[30]^{7}	0.914	0.941	0.911	0.944	0.928	0.951	0.925	0.926	0.930	0.947	0.676	0.938	0.790	0.905
[30]^{8}	0.888	0.904	0.876	0.931	0.907	0.936	0.911	0.889	0.920	0.930	0.695	0.923	0.733	0.880
[30]^{9}	0.888	0.918	0.878	0.931	0.912	0.936	0.896	0.883	0.919	0.927	0.602	0.917	0.803	0.870
[30]^{10}	0.871	0.886	0.864	0.900	0.893	0.909	0.895	0.851	0.894	0.905	0.596	0.893	0.713	0.845
[30]^{11}	0.987	0.968	0.987	0.982	0.989	0.990	0.988	0.987	0.986	0.989	0.000	0.965	0.955	0.983
[30]^{12}	0.941	0.954	0.939	0.969	0.951	0.972	0.951	0.951	0.954	0.971	0.843	0.960	0.899	0.951
[30]^{13}	0.980	0.976	0.980	0.980	0.981	0.983	0.980	0.980	0.980	0.983	0.911	0.969	0.878	0.978
[30]^{14}	0.977	0.970	0.976	0.977	0.977	0.982	0.975	0.977	0.976	0.981	0.896	0.956	0.865	0.974
[30]^{15}	0.944	0.958	0.941	0.972	0.961	0.975	0.961	0.957	0.961	0.972	0.913	0.948	0.816	0.956
[30]^{16}	0.976	0.975	0.976	0.975	0.978	0.981	0.980	0.978	0.976	0.982	0.302	0.954	0.884	0.974
[30]^{17}	0.928	0.957	0.923	0.960	0.933	0.965	0.932	0.935	0.940	0.959	0.804	0.936	0.825	0.921
[30]^{18}	0.927	0.956	0.923	0.958	0.932	0.963	0.931	0.932	0.940	0.958	0.194	0.935	0.798	0.915
[30]^{19}	0.872	0.898	0.862	0.907	0.901	0.919	0.894	0.869	0.898	0.909	0.000	0.887	0.771	0.851
[30]^{20}	0.884	0.783	0.874	0.907	0.897	0.912	0.904	0.884	0.903	0.908	0.679	0.894	0.738	0.865
[30]^{21}	0.893	0.756	0.880	0.933	0.913	0.938	0.913	0.903	0.921	0.929	0.774	0.919	0.773	0.892
[30]^{22}	0.962	0.924	0.961	0.961	0.969	0.970	0.969	0.966	0.967	0.971	0.000	0.949	0.855	0.964
[30]^{23}	0.948	0.963	0.947	0.975	0.959	0.979	0.959	0.958	0.959	0.974	0.869	0.963	0.919	0.961
[30]^{24}	0.978	0.977	0.977	0.978	0.980	0.982	0.980	0.980	0.979	0.982	0.945	0.966	0.897	0.977
[30]^{25}	0.975	0.970	0.975	0.977	0.980	0.982	0.980	0.980	0.978	0.982	0.000	0.953	0.922	0.976
[30]^{26}	0.922	0.947	0.917	0.950	0.930	0.956	0.930	0.928	0.934	0.952	0.000	0.917	0.813	0.915
[30]^{27}	0.943	0.939	0.941	0.970	0.960	0.973	0.961	0.955	0.960	0.971	0.605	0.957	0.816	0.956
[30]^{28}	0.923	0.946	0.918	0.951	0.933	0.956	0.928	0.929	0.934	0.951	0.353	0.924	0.789	0.911
[30]^{29}	0.947	0.955	0.945	0.969	0.955	0.974	0.955	0.955	0.958	0.972	0.910	0.961	0.917	0.959
[30]^{30}	0.872	0.898	0.862	0.907	0.901	0.920	0.891	0.878	0.898	0.908	0.564	0.878	0.771	0.851
[38]^{1}	0.961	0.940	0.961	0.973	0.968	0.977	0.967	0.966	0.969	0.977	0.965	0.950	0.946	0.964
[38]^{2}	0.952	0.918	0.951	0.963	0.959	0.968	0.958	0.959	0.959	0.969	0.955	0.943	0.942	0.952
[38]^{3}	0.947	0.918	0.946	0.964	0.955	0.970	0.955	0.950	0.959	0.970	0.951	0.940	0.961	0.877
[38]^{4}	0.957	0.929	0.957	0.964	0.963	0.969	0.962	0.952	0.964	0.970	0.963	0.938	0.964	0.913
[38]^{5}	0.962	0.924	0.961	0.964	0.965	0.970	0.965	0.952	0.966	0.970	0.964	0.946	0.940	0.958
[38]^{6}	0.959	0.928	0.962	0.971	0.967	0.975	0.967	0.954	0.969	0.974	0.965	0.949	0.967	0.836
[38]^{7}	0.953	0.919	0.952	0.945	0.957	0.958	0.954	0.906	0.958	0.960	0.952	0.922	0.957	0.839
[38]^{8}	0.955	0.932	0.955	0.969	0.961	0.972	0.960	0.942	0.962	0.972	0.960	0.950	0.949	0.949
[38]^{9}	0.946	0.930	0.945	0.961	0.955	0.967	0.954	0.948	0.956	0.966	0.955	0.939	0.935	0.949

Note: EPN has higher accuracy scores in all tests as compared to LSTM, MLP, SVR, and ETR. Yellow highlighted values represent output with minimal error of algorithms (nodes $\mathbf{w}_1 \ldots \mathbf{w}_9$) in the first hidden layer. Green highlighted values show that the error score in the final EPN node is less than or equal to the error scores of the nine algorithms in the first hidden layer.

are more likely in microgrid networks than in typical grids, and even more so for nanogrid networks. Another research direction could include extending the EPN to compensate for missing data, a common issue for grid operators.

Supplemental documents and experimental source code can be viewed/downloaded from GitHub at https://github.com/compsust/HourlyPredictionEnsemble.

References

[1] R. Lasseter, “Smart distribution: Coupled microgrids,” *Proceedings of the IEEE*, vol. 99, no. 6, pp. 1074–1082, 2011.

[2] US Department of Energy, “DOE Microgrid Workshop Report,” https://www.energy.gov/sites/prod/files/MicrogridWorkshopReportAugust2011.pdf, US Department of Energy.

[3] Y. Zheng, Z. Dong, F. Luo, K. Meng, J. Qiu, and K. Wong, “Optimal allocation of energy storage system for risk mitigation of discos with high renewable penetrations,” *Power Systems, IEEE Transactions on*, vol. 29, pp. 212–220, 01 2014.

TABLE VII: Testing Results Using MAPE Where EPN is Compared to LSTM, MLP, SVR, and ETR

Dataset	Linear Predictors Used in EPN (Nodes $\mathbf{w}_1 \ldots \mathbf{w}_9$)									**EPN**	Compared To:			
	UP	**LCP**	**HLP**	**DLP**	**RbP**	**RdP**	**MMP**	**KbP**	**ALM**	**(GMC)**	LSTM	MLP	SVR	ETR
[31]	0.147	0.154	0.176	0.147	0.128	0.135	0.128	0.128	0.128	0.127	0.140	0.142	0.292	0.166
[32]	0.120	0.173	0.123	0.120	0.108	0.102	0.109	0.108	0.108	0.095	0.116	0.148	0.201	0.108
[12]	0.112	0.085	0.120	0.067	0.096	0.058	0.096	0.092	0.091	0.062	0.115	0.096	0.315	0.091
[33]	0.201	0.334	0.218	0.204	0.190	0.176	0.197	0.191	0.187	0.181	0.335	0.248	0.279	0.215
[34]	0.076	0.122	0.077	0.065	0.069	0.058	0.067	0.070	0.064	0.056	0.064	0.111	0.093	0.069
[35]	0.098	0.171	0.100	0.047	0.089	0.040	0.085	0.088	0.081	0.049	0.089	0.078	0.097	0.105
[37]^{1}	0.064	0.096	0.064	0.052	0.059	0.046	0.057	0.059	0.053	0.044	0.058	0.072	0.089	0.062
[37]^{2}	0.123	0.165	0.125	0.119	0.100	0.099	0.097	0.100	0.102	0.095	0.119	0.129	0.114	0.759
[37]^{3}	0.074	0.105	0.075	0.057	0.064	0.048	0.066	0.070	0.061	0.049	0.066	0.113	0.071	0.106
[37]^{4}	0.094	0.158	0.096	0.088	0.081	0.078	0.080	0.091	0.081	0.073	0.090	0.133	0.092	0.689
[37]^{5}	0.074	0.095	0.074	0.056	0.066	0.051	0.062	0.066	0.060	0.049	0.066	0.090	0.057	0.169
[30]^{1}	0.006	0.020	0.006	0.025	0.006	0.005	0.006	0.006	0.009	0.008	4.891	0.026	0.062	0.021
[30]^{2}	0.121	0.140	0.126	0.067	0.100	0.058	0.100	0.102	0.097	0.070	0.164	0.093	0.213	0.095
[30]^{3}	0.042	0.071	0.044	0.040	0.043	0.035	0.043	0.044	0.044	0.036	0.090	0.065	0.304	0.048
[30]^{4}	0.031	0.041	0.031	0.029	0.027	0.021	0.027	0.032	0.029	0.023	0.034	0.062	0.176	0.039
[30]^{5}	0.101	0.090	0.106	0.043	0.072	0.039	0.071	0.069	0.072	0.048	0.103	0.090	0.440	0.076
[30]^{6}	0.192	0.174	0.210	0.082	0.213	0.070	0.214	0.213	0.184	0.101	0.305	0.177	0.523	0.214
[30]^{7}	0.233	0.135	0.246	0.126	0.228	0.111	0.241	0.240	0.225	0.133	0.897	0.185	0.790	0.274
[30]^{8}	0.246	0.227	0.269	0.162	0.209	0.145	0.216	0.281	0.206	0.161	0.679	0.188	0.775	0.257
[30]^{9}	0.241	0.171	0.265	0.140	0.207	0.125	0.257	0.292	0.188	0.158	0.944	0.194	0.467	0.292
[30]^{10}	0.328	0.266	0.343	0.235	0.283	0.212	0.270	0.435	0.283	0.239	1.150	0.273	0.867	0.417
[30]^{11}	0.024	0.064	0.024	0.033	0.021	0.016	0.021	0.023	0.026	0.020	3.714	0.068	0.097	0.034
[30]^{12}	0.116	0.095	0.120	0.064	0.099	0.057	0.099	0.098	0.095	0.060	0.309	0.084	0.211	0.095
[30]^{13}	0.043	0.051	0.044	0.044	0.042	0.039	0.043	0.043	0.043	0.037	0.188	0.065	0.279	0.049
[30]^{14}	0.045	0.060	0.046	0.045	0.045	0.036	0.048	0.045	0.047	0.037	0.206	0.086	0.314	0.050
[30]^{15}	0.109	0.085	0.115	0.057	0.078	0.051	0.079	0.086	0.080	0.057	0.175	0.104	0.472	0.087
[30]^{16}	0.046	0.047	0.046	0.046	0.043	0.036	0.040	0.043	0.046	0.035	1.531	0.088	0.238	0.052
[30]^{17}	0.184	0.099	0.198	0.088	0.192	0.077	0.195	0.185	0.171	0.099	0.448	0.159	0.513	0.200
[30]^{18}	0.193	0.108	0.208	0.099	0.208	0.086	0.211	0.206	0.184	0.105	1.788	0.173	0.658	0.235
[30]^{19}	0.286	0.213	0.311	0.188	0.230	0.165	0.262	0.340	0.247	0.200	3.179	0.269	0.578	0.342
[30]^{20}	0.370	0.759	0.395	0.309	0.340	0.289	0.331	0.433	0.356	0.322	1.094	0.387	1.022	0.438
[30]^{21}	0.246	0.604	0.271	0.179	0.210	0.164	0.221	0.255	0.216	0.188	0.539	0.227	0.711	0.247
[30]^{22}	0.075	0.157	0.075	0.077	0.062	0.060	0.062	0.068	0.067	0.058	5.079	0.101	0.305	0.074
[30]^{23}	0.101	0.074	0.105	0.050	0.080	0.043	0.081	0.082	0.082	0.051	0.260	0.076	0.165	0.076
[30]^{24}	0.049	0.050	0.049	0.050	0.044	0.041	0.044	0.043	0.046	0.040	0.115	0.073	0.242	0.051
[30]^{25}	0.047	0.055	0.047	0.042	0.036	0.032	0.037	0.037	0.043	0.033	2.510	0.093	0.175	0.046
[30]^{26}	0.218	0.120	0.234	0.117	0.228	0.101	0.229	0.234	0.209	0.120	3.513	0.239	0.617	0.246
[30]^{27}	0.111	0.123	0.115	0.062	0.081	0.055	0.080	0.091	0.082	0.060	0.837	0.090	0.473	0.085
[30]^{28}	0.237	0.134	0.259	0.127	0.235	0.113	0.268	0.262	0.240	0.135	1.480	0.233	0.872	0.297
[30]^{29}	0.105	0.092	0.109	0.064	0.092	0.055	0.092	0.092	0.088	0.058	0.185	0.082	0.172	0.080
[30]^{30}	0.286	0.213	0.311	0.188	0.230	0.164	0.272	0.314	0.247	0.206	1.150	0.297	0.578	0.342
[38]^{1}	0.080	0.123	0.081	0.055	0.065	0.046	0.067	0.069	0.065	0.048	0.072	0.101	0.108	0.073
[38]^{2}	0.097	0.161	0.099	0.072	0.080	0.061	0.082	0.082	0.082	0.062	0.091	0.113	0.114	0.093
[38]^{3}	0.112	0.168	0.114	0.072	0.094	0.059	0.093	0.102	0.086	0.062	0.104	0.123	0.079	0.254
[38]^{4}	0.086	0.148	0.087	0.071	0.075	0.061	0.076	0.096	0.073	0.060	0.076	0.125	0.071	0.174
[38]^{5}	0.078	0.156	0.078	0.072	0.070	0.060	0.071	0.096	0.070	0.059	0.073	0.110	0.119	0.085
[38]^{6}	0.083	0.147	0.077	0.057	0.066	0.049	0.067	0.092	0.064	0.053	0.071	0.104	0.066	0.329
[38]^{7}	0.095	0.169	0.097	0.111	0.088	0.086	0.093	0.183	0.086	0.082	0.100	0.155	0.088	0.318
[38]^{8}	0.092	0.136	0.093	0.061	0.080	0.055	0.081	0.117	0.077	0.056	0.083	0.101	0.100	0.101
[38]^{9}	0.110	0.142	0.112	0.077	0.091	0.063	0.093	0.102	0.091	0.069	0.092	0.122	0.129	0.101

Note: EPN has higher accuracy scores in all tests as compared to LSTM, MLP, SVR, and ETR. Yellow highlighted values represent output with minimal error of algorithms (nodes $\mathbf{w}_1 \ldots \mathbf{w}_9$) in the first hidden layer. Green highlighted values show that the error score in the final EPN node is less than or equal to the error scores of the nine algorithms in the first hidden layer.

TABLE VIII: RMSE: Length of timeslots

	Decreased by 2		Increased by 2		Original	
	LCP	EPN	LCP	EPN	LCP	EPN
[31]	3.43	2.91	3.51	2.90	3.44	2.91
[32]	4.90	2.95	4.20	2.60	4.13	2.52
[12]	5.65	4.56	6.80	4.42	5.64	4.52
[34]	2.27	1.08	2.28	1.09	2.30	1.08

[4] 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.

[5] M. Cenek, R. Haro, B. Sayers, and J. Peng, "Climate change and power security: Power load prediction for rural electrical microgrids using long short term memory and artificial neural networks," *Applied Sciences*, vol. 8, no. 749, pp. 1–16, May 2018.

[6] H. Chitsaz, H. Shaker, H. Zareipour, D. Wood, and N. Amjady, "Short-term electricity load forecasting of buildings in microgrids," *Energy and Buildings*, vol. 99, pp. 50–60, Jul. 2015.

[7] Y. K. Semero, J. Zhang, and D. Zheng, "EMD-PSO-ANFIS-based hybrid approach for short-term load forecasting in microgrids," *IET Generation,*

Transmission and Distribution, vol. 14, no. 3, pp. 470–475, Feb. 2020.

[8] Z. Ming-guang and L. Lin-rong, "Short-term load combined forecasting method based on BPNN and LS-SVM," in *Proc. IEEE Power Engineering and Automation Conference*. Wuhan, China: IEEE, Sep. 2011.

[9] Y. Liu, T. Wei, W. Zhang, M. Zhang, Y. Li, X. Li, Y. Zhang, S. Hao, and F. Zhao, "Short-term daily load forecasting in distribution power system based on vmd and ba-lssvm combined method," in *Proc. IEEE PES Innovative Smart Grid Technologies Asia*, Chengdu, China, Oct. 2019, pp. 364–368.

[10] B. Li, J. Zhang, Y. He, and Y. Wang, "Short-term load-forecasting method based on wavelet decomposition with second-order gray neural network model combined with ADF test," *IEEE Access*, vol. 5, pp. 16 324–16 331, Sep. 2017.

[11] F. M. Bianchi, E. de Santis, A. Rizzi, and A. Sadeghian, "Short-term electric load forecasting using echo state networks and PCA decomposition," *IEEE Access*, vol. 3, pp. 1931–1943, Oct. 2015.

[12] H. J. Sadaei, P. C. L. e Silva, F. G. Guimaraes, and M. H. Lee, "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series," *Energy*, vol. 175, pp. 365–377, May 2019.

[13] N. Amjady, F. Keynia, and H. Zareipour, "Short-term load forecast of microgrids by a newbilevel prediction strategy," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 286–294, Dec. 2010.

[14] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland, and S. J. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 322–330, Feb. 2010.

[15] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4356–4364, Nov. 2013.

[16] N. Liu, Q. Tang, J. Zhang, W. Fan, and J. Liu, "A hybrid forecasting model with parameter optimization for short-term load forecasting of micro-grids," *Applied Energy*, vol. 129, pp. 336–345, Jun. 2014.

[17] Y. Guo, E. Nazarian, J. Ko, and K. Rajurkare, "Hourly cooling load forecasting using time-indexed arx models with two-stage weighted least squares regression," *Energy Conversion and Management*, vol. 80, pp. 46–53, Apr. 2014.

[18] L. Harnandez, C. Baladron, J. M. Aguiar, B. Carro, A. S. Esguevillas, and J. Lioret, "Artificial neural networks for short-term load forecasting in microgrids environment," *Energy*, vol. 75, pp. 252–264, Aug. 2014.

[19] M. Ghofrania, M. Ghayekhloob, A. Arabalic, and A. Ghayekhloo, "A hybrid short-term load forecasting with a new input selection framework," *Energy*, vol. 81, pp. 777–786, Jan. 2015.

[20] D. Chaturvedi, A. Sinha, and O. P. Malik, "Short term load forecast using fuzzy logic and wavelet transform integrated generalized neural network," *Electrical Power and Energy Systems*, vol. 67, pp. 230–237, 2015.

[21] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electric Power Systems Research*, vol. 122, pp. 96–103, Jan. 2015.

[22] X. Zhang, J. Wang, and K. Zhang, "Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by cuckoo search algorithm," *Electric Power System Research*, vol. 146, pp. 270–285, Feb. 2017.

[23] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd ed. New York, NY, USA: Springer, 2008.

[24] M. Z. A. Bhotto and A. Antoniou, "Affine-projection-like adaptive-filtering algorithms using gradient-based step size," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 2048–2056, Jul. 2014.

[25] G. M. D. Mijolla, S. Konstantinopoulos, P. Gao, J. H. Chow, and M. Wang, "An evaluation of algorithms for synchrophasor missing data recovery," in *Proc. IEEE Power System Computation Conference*. Dublin, Ireland: IEEE, Aug. 2018.

[26] M. Z. A. Bhotto and A. Antoniou, "New constrained affine-projection adaptive-filtering algorithm," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2013, pp. 517–520.

[27] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, May 1998.

[28] O. Tanrikulu and A. G. Constantinides, "Least-mean kurtosis: a novel higher-order statistics based adaptive filtering algorithm," *Electronics Letters*, vol. 30, no. 3, pp. 189–190, 1994.

[29] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[30] Office of Energy Efficiency and Renewable Energy, "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-s

[31] S. Makonin, "HUE: The hourly usage of energy dataset for buildings in British Columbia," *Data in Brief*, vol. 23, no. 103744, pp. 1–4, 2019.

[32] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *International Journal of Forecasting*, vol. 30, no. 2, pp. 357–363, Apr. 2014.

[33] National Renewable Energy Laboratory, "NREL RSF measured date 2011," https://openei.org/datasets/dataset/nrel-rsf-measured-data-2011, 2011, US Department of Energy.

[34] ISO New England Inc., "Energy, load, and demand reports," https://www.iso-ne.com/isoexpress/web/reports/load-and-demand.

[35] European load data, "Entso-e data portal," https://www.entsoe.eu/data/data-portal/.

[36] S. Makonin and F. Popowich, "Nonintrusive load monitoring (NILM) performance evaluation," *Energy Efficiency*, pp. 8:809–814, 2015, doi.org/10.1007/s12053-014-9306-2.

[37] IESO, "IESO, connecting today powering tomorrow," http://www.ieso.ca/en/Power-Data/Data-Directory.

[38] PJM Interconnection LLC, "PJM Hourly Energy Consumption Data," https://www.kaggle.com/robikscube/hourly-energy-consumption, eastern US grid.