Superiority of the Neural Network Dynamic Regression Models for Ontario Electricity Demand Forecasting

***Abstract*—Electricity demand forecasting is an essential step in balancing the electricity grid. A balanced electricity grid helps make efficient decisions in production planning and scheduling. The study of uncertainty in electricity demand requires more advanced hybrid forecasting approaches, which combine statistical methods and machine learning techniques. In electricity demand forecasting, the features such as temperature and day type are suggested as the most influential features in the literature. The novelty of this paper is to use Canadian electricity demand data and explore the effect of temperature and day type on electricity demand using hybrid models. The hybrid neural network dynamic regression (NNDR) model is used to obtain the point forecasts/predictions for Ontario electricity demand data. The experimental results show the superiority of the proposed NNDR model over the commonly used dynamic regression models with the Seasonal Autoregressive Integrated Moving Average errors (DRSARIMA) model and the Facebook Prophet model. Moreover, long-term point forecasts and innovations are used to obtain two classes of prediction intervals (PIs) using data-driven probabilistic innovation distribution and bootstrapping for NNDR, DRSARIMA and Prophet.**

***Index Terms*—Innovations, Prediction intervals, Neural Network Dynamic Regression, Prophet Forecasting Package, Ontario Electricity Demand**

I. Introduction

Electricity demand forecasting is an important step in balancing the electricity grid, and an accurate electricity demand forecast is a primary input to optimize production planning and scheduling. Depending on the industrial requirements, different time horizons may impose on electricity demand forecasting, and in some cases, forecasts may be computed for a few years.

Among the different approaches used in electricity demand forecasting, Neural Nets (NN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), dynamic regression with Seasonal Autoregressive Integrated Moving Average (SARIMA) errors are popular approaches among scholars [1]–[6]. Univariate time series forecasting methods exploit daily and weekly patterns in electricity demand data relying on SARIMA models and they are very efficient for short-term forecasts with small data. Studies show that when NN algorithms augmented with deep-learning are used in forecasting time series data (e.g., LSTM), they perform better than traditional algorithms (e.g., ARIMA) [7].

Machine learning forecasting models are also more appropriate to incorporate calendar characteristics and meteorological effects as inputs when models are trained on a large set of historical data. Even though machine learning models lead to accurate forecasting, they are weak in interpretability. Models that combine machine learning methods and statistical methods are known as hybrid models. Popular forecasting algorithms include hybrid models such as gradient boosting machines and LSTM combined with triple exponential smoothing [4], neural networks (NN) [5], or statistical models like Generalized Additive Models (GAM) combined with NN. Obst et al. [5] and Hewamalage et al. [9] discuss the superiority of such models.

A recent study by Liang et al. [11] and Thavaneswaran et al. [10] discuss improving risk forecasts using data-driven volatility models, and the study emphasizes that the distribution of innovations can be determined using sign correlation. In general, innovation distribution is a heavy-tailed t distribution, and appropriate degrees of freedom (d.f.) are derived from sign correlation. Thavaneswaran et al. [12] explores the possibility of applying neuro volatility models to financial data to forecast Value-at-Risk (VaR). Thavaneswaran et al. [13] discussed fuzzy option pricing using the data-driven feedforward NN variance model.

A type of confidence interval used with predictions is known as prediction intervals (PIs). PIs provide the measure of uncertainty of the point predictions/forecasts, and PIs are used to compare the point predictions/forecasts. Recently, Liang et al. [11] proposed a method for calculating PIs for deep learning neural network predictions and hybrid model predictions. Also, there is a lack of studies on obtaining data-driven PIs for deep learning neural network, which are reliable and interpretable. In this study, PIs are constructed using model innovations and the point forecasts based on dynamic regression models with Seasonal Autoregressive Integrated Moving Average errors (DRSARIMA), neural networks dynamic regression (NNDR) model, and Prophet model. SARIMA model-based forecasts are obtained by putting restrictions on the model parameters such as stationarity, whereas NNDR fits a semiparametric nonlinear model for the nonstationary response variable. This study explores two methods to construct data-driven innovation distribution-based PIs and bootstrap PIs based on innovations of the models. The remainder of the paper consists of two sections and conclusions. Section II discusses models and methods, and section III discusses a detailed summary of the numerical experiment(s) for electricity demand forecasting.

Preprint

II. Models, Forecasts and Prediction Intervals

A. Forecasting Models

1) DRSARIMA: The dynamic regression model with SARIMA $(p,d,q)(P,D,Q)_m$ errors is given by, for $t = 1,\ldots,n$,

$$y_t = f(x_{1,t},\ldots x_{L,t}) + \eta_t$$
$$\phi(B)\Phi(B^m)(1-B)^d(1-B^m)^D\eta_t = \theta(B)\Theta(B^m)\epsilon_t$$
$$\epsilon_t \sim IID(0,\sigma^2)$$

(p,d,q) is the non-seasonal part of the SARIMA errors η_t where p is non-seasonal AR order, d is the degree of non-seasonal differencing involved and q is the non-seasonal MA order. $(P,D,Q)_m$ is the seasonal part of the SARIMA errors where P,D,Q are the seasonal AR order, the degree of seasonal differencing and the seasonal MA order, respectively. m is the seasonal period. $y_1,\ldots,y_n$ is the target series. ϵ_t are independent and identically distributed (IID) errors with mean 0 and variance σ^2. f is a function of the k features $\mathbf{x}_1,\ldots,\mathbf{x}_L$, where $\mathbf{x}_i = x_{i1},\ldots,x_{in}$ for each $i = 1,\ldots,L$. B is the differencing operator, for example $(1-B)\eta_t = \eta_t - \eta_{t-1}$. $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ is the non-seasonal AR component. $\theta(B) = 1 + \theta_1 B + \cdots + \phi_q B^q$ is the non-seasonal MA component. $\Phi(B^m) = 1 - \Phi_1 B^m - \cdots - \Phi_P B^{Pm}$ is the seasonal AR component. $\Theta(B^m) = 1 + \Theta_1 B^m + \cdots + \Theta_Q B^{Qm}$ is the seasonal MA component. $\phi_1,\ldots,\phi_p,\theta_1,\ldots,\theta_q,\Phi_1,\ldots,\Phi_P,\Theta_1,\ldots,\Theta_Q$ are the model parameters.

2) NNDR: A popular and common approach to approximate multivariate nonlinear function is feedforward NN [14]. It consists of input variables, hidden layer(s), and an output node. A hidden layer consists of small individual units called neurons, and hidden layer(s) represents a very efficient parallel distributed nonlinear regression model. Neurons in the network automatically learn new features from the data through a process called supervised learning. Thus, a feedforward NN can approximate any continuous function uniformly for a sufficient number of neurons and training samples. The functions that translate input signals to output signals are known as transfer functions, and popular choices are logistic sigmoid or the hyperbolic tangent for regression problems.

As lagged valued are used in an autoregression model they can be used as inputs in NN model For time series data. Suppose have p lagged values of the target variable $(y_{t-1}, y_{t-2},\ldots,y_{t-p})$ and selected features $(x_{1,t},\ldots,x_{L,t})$. Here lags and features are inputs of the NN model and interested outcome will be the value of y_t. Note that categorical feature are available they need to be encorded to before included in the model. Fig. 1 visualize a NN with $k-1$ hidden layers with $p+L$ inputs. The final layer has an output function h and corresponding weights. For quantitative regression h is typically the identity function. Since each layer of the NN is functions of previous layer(s) and network represents a complex function $f(y_{t-1},\ldots,y_{t-p},x_{1,t},\ldots,x_{L,t};W)$, where W represents the entire collection of weights. Note here that complex network given in Fig. 1 depicted with parameters following [8].

From the first to the second layer we have

$$z_l^{(2)} = \omega_{l0}^{(1)} + \sum_{j=1}^{p}\omega_{lj}^{(1)}y_{t-j} + \sum_{j=p+1}^{p+L}\omega_{lj}^{(1)}x_{j-p,t} \tag{1}$$

and

$$a_l^{(2)} = g^{(2)}(z_l^{(2)}) \tag{2}$$

We have separated the linear transformations $z_l^{(2)}$ of the y_{t-j} and $x_{j-p,t}$ from their non-linear transformations, and we allow for layer-specific non-linear transformations $g^{(k)}$. The transition from layer $k-1$ to k is modelled as:

$$z_l^{(k)} = \omega_{l0}^{(k-1)} + \sum_{j=1}^{p_{k-1}}\omega_{lj}^{(k-1)}a_j^{(k-1)} \tag{3}$$

and

$$a_l^{(k)} = g^{(k)}(z_l^{(k)}) \tag{4}$$

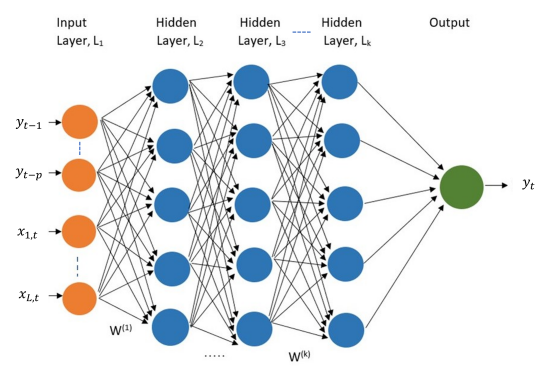

Fig. 1. NNDR

3) Prophet: Prophet is a time-series forecasting package developed at Facebook [18]. Prophet has three additive components in its forecast model. (1) Trend (growth g over time, linear or nonlinear), (2) seasonality (s, within year), cyclicality (across years), and (3) holidays (h, irregular breaks). It is written as, for time series y (More extensive details are in the paper [18]):

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{5}$$

where ϵ_t is the error term accounts for any unusual changes not accommodated by the model. Prophet re-frames the forecasting problem as a curve-fitting exercise.

1) Growth is modeled as $g(t) = \frac{C}{1+\exp(-k(t-m))}$, where C is the saturation level, k is the slope, base growth rate, $\frac{\partial g}{\partial k} > 0$, m is the time offset, and $C(t)$ can be made a function of time with exogenous analyst forecasts.
2) Seasonality is modeled as a Fourier series. Here, P: period, equal to 365 for annual, 7 for weekly. N: components of the Fourier series. $s(t) = \sum_{n=1}^{N}\left[a_n\cos\left(\frac{2\pi nt}{P}\right) + b_n\sin\left(\frac{2\pi nt}{P}\right)\right]$.

Preprint

3) Holidays are modeled as follows. κ_i is change in forecast at time i, written into a vector κ. $I(t)$ is a indicator vector of holiday dummies. Then $h(t) = I(t) \cdot \kappa$.

B. Prediction Intervals

1) Data-driven probabilistic PIs: Calculating PIs for NNs is challenging, and there are no standard techniques available in the literature. In the following architecture, we illustrate how to train a DRNN to forecast future observations and construct data-driven PIs using model innovations. The method is applicable in a wider context for any statistical or NN models. Each training sample consists of a window of n past observations that has L features $\mathbf{x}_1, \ldots, \mathbf{x}_L$, where $\mathbf{x}_i = x_{i1}, \ldots, x_{in}$ for each $i = 1, \ldots, L$. Therefore, each feature series has n past observations for training. The target series is $y_t, t = 1, \ldots, n$. The model inputs $\mathbf{x}_1, \ldots, \mathbf{x}_L$ and $y_1, \ldots, y_n$ to build a forecasting model and produces n innovations $e_1, \ldots, e_n$, where $e_i = y_i - \hat{y}_i$. Then by using the test sample consisting of l observations of L features $\mathbf{x}_1, \ldots, \mathbf{x}_L$, the model produces l forecasts $\hat{y}_{n+1}, \ldots, \hat{y}_{n+l}$. The standard deviation of the distribution of forecasts can be estimated using the standard deviation of the innovations given by $\hat{\sigma}$.

This paper describes a data-driven approach to computing PIs when the normality assumption for the innovations is not always satisfied. Using the definition of sign correlation given in [10], the sample sign correlation of the residuals $e_1, \ldots, e_n$ with sample mean $\bar{e}$ is defined as

$$\hat{\rho}_e = \text{Corr}(e_t - \bar{e}, \text{sign}(e_t - \bar{e}))$$

If e_t follows a normal distribution, $\hat{\rho}_e = \sqrt{2/\pi}$. If e_t follows a Laplace distribution, $\hat{\rho}_e = 1/\sqrt{2}$. If e_t follows a Student's t distribution, $\hat{\rho}_e$ is less than $\sqrt{2/\pi}$. The corresponding estimated d.f. ν can be computed by solving the following nonlinear equation

$$2\sqrt{\nu - 2} = (\nu - 1)\hat{\rho}_e \text{Beta}\left[\frac{\nu}{2}, \frac{1}{2}\right]$$

where $\text{Beta}(x, y)$ is the beta function. The sign correlation plays an important role in finding the probabilistic forecasts of the demand by determining an appropriate data-driven distribution of the innovations. If the innovations follow a t distribution with sign correlation $\hat{\rho}_e$ and estimated d.f. ν, then the demand PI with $100(1-p)\%$ significance level is calculated as

$$\hat{y}_{n+h} \pm t_{p/2,\nu}\hat{\sigma} \tag{6}$$

2) Bootstrap PIs: When the normality assumption for the innovations is not valid, bootstrapping can be used as an alternative approach. This approach assumes that the innovations are uncorrelated with almost constant variance. Once we have the l forecasts $\hat{y}_{n+1}, \ldots, \hat{y}_{n+l}$, assuming future errors and past errors are similar, l errors $e_{n+1}, \ldots, e_{n+l}$ are sampled from the collection of errors from the training data (i.e., the innovations or residuals). Point forecasts added with sampled errors consists of a sequence of simulated future observations $\hat{y}_{n+1} + e_{n+1}, \ldots, \hat{y}_{n+l} + e_{n+l}$. By doing this repeatedly for s times, s realizations of the future observations of the target y_t are obtained. Then percentiles (2.5th percentile and 97.5th percentile for 95% PI) of the simulated future observations at each prediction horizon are calculated to compute the PIs of the target. The resulting PIs are called bootstrapped PIs. The bootstrapping process allows us to measure the future uncertainty of the forecasts by using historical data only.

III. Experiments with Real Data

The study considers daily electricity demand (GW) data of Ontario province, Canada, from January 2017 to December 2021. Data also contain daily temperature and day type. Day type categorizes each day as a weekday, day of a weekend, or holiday. Fig. 2 visualizes daily electricity demand and temperature pattern throughout the study period, and it also visualizes daily electricity demand for Ontario against the maximum daily temperature. Fig. 2 also suggests a non-linear dynamic model for the data. In the experimental section of the study, a cubic regression model with SARIMA errors and a feed-forward networks model are investigated along with a Prophet model. All the statistical analyses in the study are carried forward in RStudio, and ARIMA() function from the R package fable and `nnetar` function from the R package forecast [15] are used.

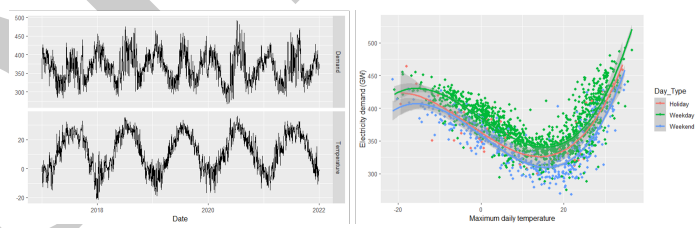

Fig. 2. Daily electricity demand versus maximum daily temperature for the province of Ontario in Canada from 2017 to 2021

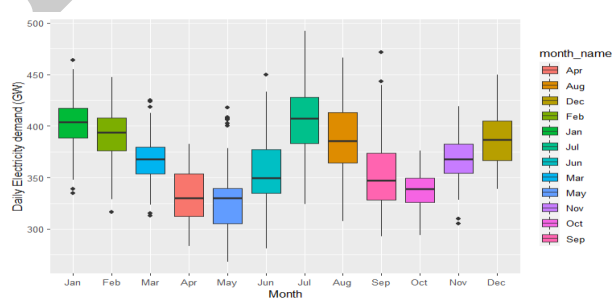

Fig. 3. Dispersion of Daily Electricity Demand from 2017 to 2021

Fig. 3 summarizes electricity demand each month for four years starting from January 2017. During the first five months of the year, a gradual decrease in electricity demand is observed, and from May to July gradual increase in electricity demand is observed. From July to October, electricity demand gradually decreases, whereas, from October to December gradual increase in electricity demand is observed. This observation coincides with season changes in general and the need for

Preprint

heating and cooling depending on the climate. Furthermore, from May to September variability of electricity demand is relatively high compared to the remaining months of the year. Thus, it is important to forecast and investigate the electricity demand further during the summer months when it is highly volatile. Data from July 2017 to June 2021 is considered as training data, and July 2021, August, and September data are used to verify the suggested models as testing data in the numerical analysis.

A. Forecasting Electricity Demand using ARIMA

The target variable for the study is electricity demand, and it is denoted by y_t. Change in electricity demand is observed against temperature (x_{1t}) and day type (x_{2t}). When regressors are included, ARIMA() produces a dynamic regression model with SARIMA $(p,d,q)(P,D,Q)_m$ errors. For the selected data, the fitted model has SARIMA $(0,1,4)(2,0,0)_7$ errors.

$$y_t = -2.0647x_{1t} + 0.0155x_{1t}^2 + 0.0023x_{1t}^3 + 22.4707x_{2t} + \eta_t$$
$$(1 - 0.0149B^7 - 0.0712B^{14})(1-B)\eta_t \quad =$$
$$(1 + 0.2052B + 0.1999B^21 + 0.1472B^3 + 0.1362B^4)\epsilon_t$$
$$\epsilon_t \sim NID(0, 180.9)$$

If the fitted regression model is adequate, the estimated errors are expected to be a white noise series. Fig. 4 visualizes innovation residuals and notes that there is an indication of heteroscedasticity in the residuals with higher volatility during the summer months and lower volatility during the rest of the months of the year. The residual histogram indicates long tails, and the ACF plot provides evidence that residuals are autocorrelated. Furthermore, these heavy tails distributions may affect the coverage probability of PIs.

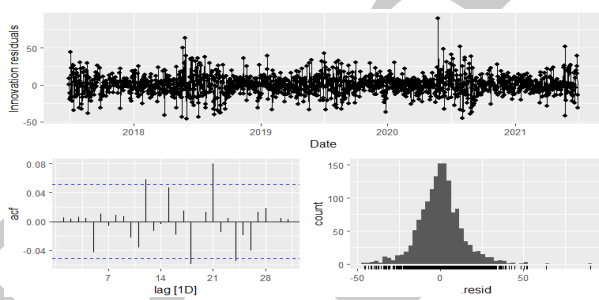

Fig. 4. Residuals diagnostics for the DRSARIMA model trained using data from 2017-07-01 to 2021-06-30

Fig.s 5 represent point forecasts for the testing period along with the training data, and blue curves denote forecasts of electricity demands for July 2021 to September 2021. Fig. 6 provides a more detailed representation of point forecasts for the testing period against the observed electricity demand. For the fitted model, test root mean square error (TRMSE) and test mean absolute deviation (TMAE) for testing data are 30.7623 and 26.3705, respectively.

As the normality assumption of the innovations is not satisfied, it is important to investigate how to obtain more accurate PIs compared to normal PIs. Non-normality is also confirmed

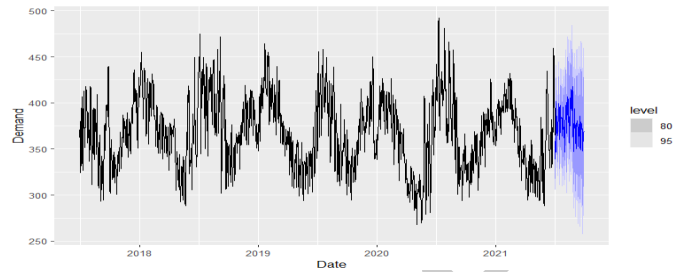

Fig. 5. Point forecasts and normal PIs from DRSARIMA for daily electricity demand from 2021-07-01 to 2021-09-30

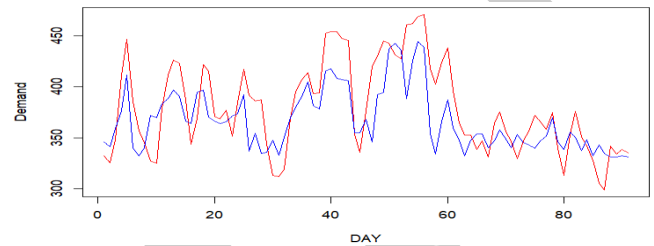

Fig. 6. Comparison of point forecasts from the DRSARIMA model (blue) and demand observations (red) from 2021-07-01 to 2021-09-30

as the appropriate distribution of the residuals is determined to be t distribution with the estimated d.f. $\nu = 4.7585$ from the sign correlation. 95% probabilistic Student t PIs are calculated using (6) with $\hat{\sigma} = 13.4499, p = 0.05, \nu = 4.7585$. 95% bootstrap PIs are calculated based on $s = 1000$ repetitions of resampling from the innovations. Fig. 7 compares 95% Student t PIs and bootstrap PIs, which is identified by green curves and dashed purple curves, respectively. The coverage of probability defines as the percentage of the time that future values were within the PIs. Student t PIs have a coverage probability of 70.33%, whereas, for bootstrap PIs, it is 58.24%. Note that Student t PIs are more accurate for given data with a larger coverage probability than bootstrap PIs.

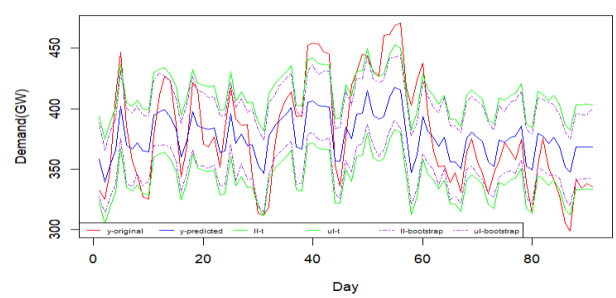

Fig. 7. 95% Student t PIs (green curves) and 95% bootstrap PIs (dashed purple curves) from the DRSARIMA model for demand from 2021-07-01 to 2021-09-30

B. Forecasting Electricity Demand using NNDR

The study investigates the performance of a feed-forward network with a hidden layer for given data improving the work of DRSARIMA. As mentioned in the previous section,

the feature day type has three levels, and the first step is to convert the day-type feature to a single numeric variable. This is achieved by designTreatmentsZ() and prepare() functions available in R package `vtreat` [17]. The next step is to fit an NNDR $(p, P, k)_m$ model to obtain demand forecasts for electricity, and the function `nnetar()` is fed with temperature and level-coded day type features. In the NNDR model, p and P are the AR orders of the non-seasonal and seasonal parts, and k is the number of nodes in the hidden layer. The fitted NNDR model contains 16 neurons in the hidden layer with the last 29 observations $y_{t-1}, \ldots, y_{t-29}$ used as inputs for forecasting the target y_t. The decay rate is set to 0.1 for regularization in order to compensate for model complexity. Note that histogram and ACF in Fig. 8 indicate residuals are autocorrelated with heavy tails. Fig. 9 visualizes point forecasts of electricity demand using NNDR models for July 2021 to September 2021. TRMSE and TMAE are calculated as 24.0211 and 17.4338 under the NNDR model, and note that they are lower than the DRSARIMA test errors.

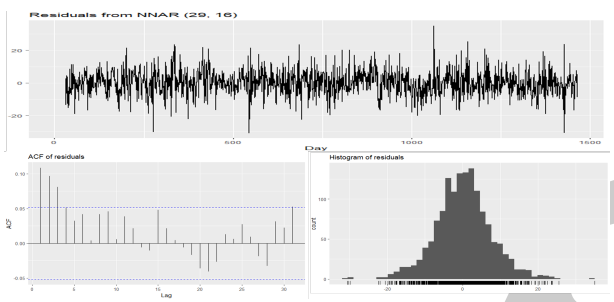

Fig. 8. Residuals diagnostics for the NNDR model trained using data from 2021-07-01 to 2021-09-30

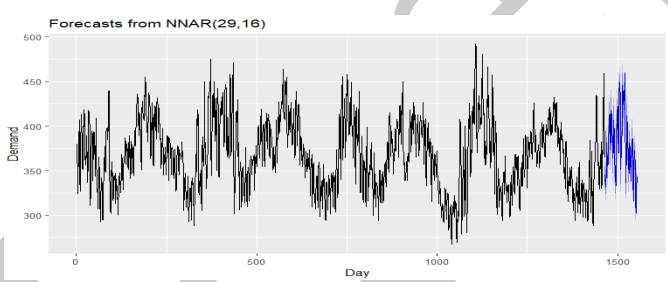

Fig. 9. Point forecasts and normal PIs from the NNDR model for daily electricity demand from 2021-07-01 to 2021-09-30

By default, `nnetar()` uses information from the in-sample residuals for the innovations used in PIs. The residuals can be arbitrarily small depending on the complexity of the model. As a reference, the standard deviation of the `nnetar()` innovations is roughly 4 times smaller than the residuals from ARIMA(). Therefore, we obtain the innovations from cross-validation (R CVar() function). The innovations from the cross-validation follow a heavy-tailed Student t distribution with the estimated d.f. equal to 5.5610. We will compare the data-driven probabilistic Student t PIs and bootstrap PIs. 95% probabilistic Student t PIs are calculated using (6) with

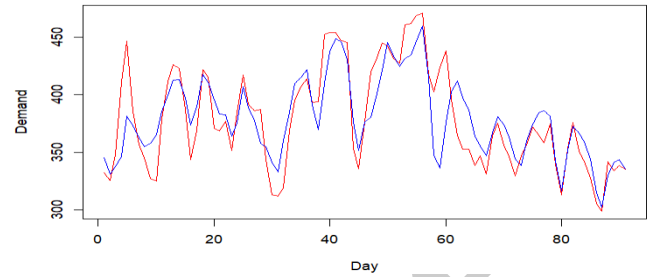

Fig. 10. Comparison of point forecasts from the NNDR model (blue) and demand observations (red) from 2021-07-01 to 2021-09-30

$\hat{\sigma} = 0.1188, p = 0.05, \nu = 5.5610$. 95% bootstrap PIs are calculated based on $s = 1000$ repetitions of resampling from the innovations. The coverage probabilities are 86.81% using Student t PIs (green curves) and 76.92% using bootstrap PIs (dashed purple curves) and they are visualized in Fig. 11. The results show that innovations follow a heavy-tailed t distribution with estimated d.f. equal to 3.6293. Hence the probabilistic Student t PIs are more accurate than bootstrap PIs.

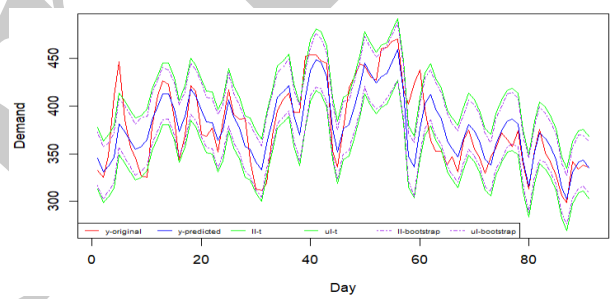

Fig. 11. 95% Student t PIs (green curves) and 95% bootstrap PIs (dashed purple curves) from the NNDR model for demand from 2021-07-01 to 2021-09-30

C. Forecasting Electricity Demand using Prophet

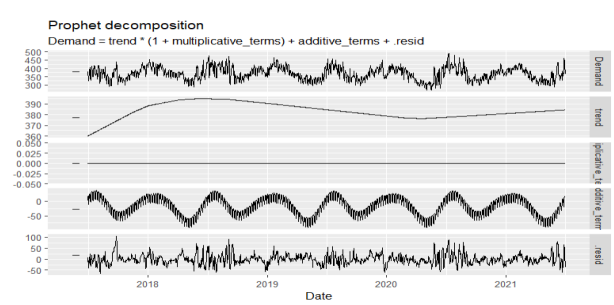

Fig. 12. Prophet Decomposition

The Prophet model has an interface in `fable` Package in R software which is used in this study to forecast daily electricity demand. Temperature and day-type are the features of the model. In Fig. 12, the decomposition of the electricity demand from Prophet is visualized and illustrated. Fig. 12 shows

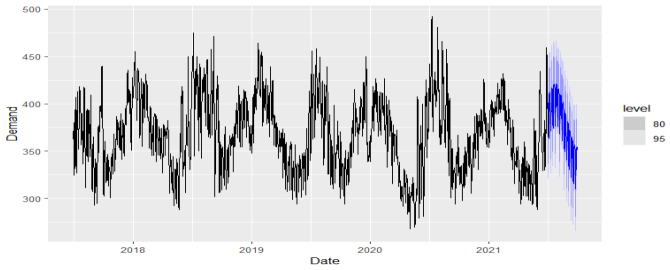

Fig. 13. Point forecasts and normal PIs from the Prophet model for daily electricity demand from 2021-07-01 to 2021-09-30

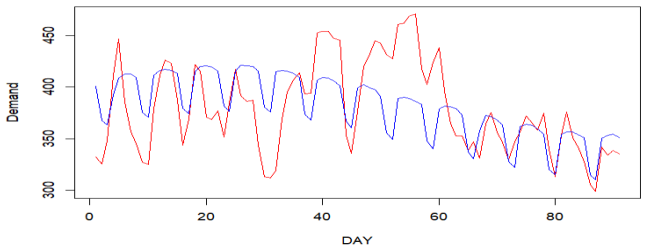

Fig. 14. Comparison of point forecasts from the Prophet model (blue) and demand observations (red) from 2021-07-01 to 2021-09-30

that the Prophet model has a piecewise linear component, a seasonal additive component, and a residuals term.

Prophet's forecasting performance is not as good as the NNDR model. TRMSE and TMAE are, respectively, 38.5343 and 30.3035, weaker than both DRSARIMA and NNDR models. Moreover, as a consequence of significant autocorrelation in the residuals, PIs are much narrower than PIs using DRSARIMA AND NNDR (Fig. 13). However, running the Prophet model takes substantially less time than the NNDR model. It is worth mentioning that the forecasting performance of the Prophet model is worse in highly-volatile months of the year, like July and August (Fig. 14).

IV. Conclusions

As electricity demand becomes volatile, improved forecasting techniques should be considered for better decision-making. The driving idea, unlike the existing work, is that the novel hybrid NNDR model is used to forecast Canadian electricity demand with multiple features (including categorical). Moreover, two different classes of PIs are obtained using data-driven innovation distribution and bootstrapping. As a comparison, we also considered DRSARIMA, which requires the co-integration of the demand and the other features. Moreover, Facebook industrial Prophet additive model is also used to obtain the forecasts. The numerical experiments show that NNDR is more accurate and resilient (see [6]) than DRSARIMA and Prophet in forecasting demand during the volatile period (summer). Moreover, recently proposed data-driven probabilistic Student t PIs have a higher coverage probability for the test sample than the bootstrap PIs for all the three models (NNDR, DRSARIMA, and Prophet).

References

[1] R. J. Hyndman and G. Athanasopoulos, Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3, 2021.

[2] B. Banitalebi, S. S. Appadoo and A. Thavaneswaran, "Data Driven Approach for Reduced Value at Risk Forecasts in Renewable Power Supply Systems," 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-6, 2020.

[3] B. Banitalebi, M. E. Hoque, S. S. Appadoo and A. Thavaneswaran, "Regularized Probabilistic Forecasting of Electricity Wholesale Price and Demand," 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 28-35, 2020.

[4] B. Banitalebi, S. S. Appadoo, Y. Gajpal and A. Thavaneswaran, "Intelligent Probabilistic Forecasts of Day-Ahead Electricity Prices in a Highly Volatile Power Market," 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), DSAT, pp. 166-175, 2021.

[5] D. Obst, J. D. Vilmarest and Y. Goude, "Adaptive Methods for Short-Term Electricity Load Forecasting During COVID-19 Lockdown in France", IEEE Transactions and Power Systems, vol. 36, pp. 4754-4763, 2021.

[6] M. E. Hoque, A. Thavaneswaran, S. S. Appadoo, R. K. Thulasiram and B. Banitalebi, "A Novel Dynamic Demand Forecasting Model for Resilient Supply Chains using Machine Learning," 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), DSAT, pp. 218-227, 2021.

[7] S. Siami-Namini, N. Tavakoli and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394-1401, 2018.

[8] B. Efron and T. J. Hastie, Computer age statistical inference: Algorithms, evidence, and data science. Cambridge University Press, New York, 2017.

[9] H. Hewamalage, C. Bergmeir and K. Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current status and future directions", International Journal of Forecasting, vol. 37, pp. 388-427, 2021.

[10] A. Thavaneswaran, A. Paseka and J. Frank, "Generalized Value at Risk Forecasting", Communications in Statistics Theory and Methods, vol. 49, pp. 4988-4995, 2020.

[11] Y. Liang and A. Thavaneswaran, "Long Term Interval Forecasts of Demand using Data-Driven Dynamic Regression Models", 2022 IEEE 46th Annual Computer Software and Applications Conference (COMPSAC 2022). Accepted.

[12] A. Thavaneswaran, R. K. Thulasiram, J. Frank, Z. Zhu, M. E. Hoque and N. Ravishanker, "Fuzzy Value-at-Risk Forecasts Using a Novel Data-Driven Neuro Volatility Predictive Model", In proceedings of 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC 2019), vol. 2, pp. 221-226, 2019.

[13] A. Thavaneswaran, R. K. Thulasiram, J. Frank, Z. Zhu and M. Singh, "Fuzzy Option Pricing Using a Novel Data-Driven Feed Forward Neural Network Volatility Model", IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, pp. 491-496, June 2019.

[14] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators", Neural Networks, vol. 2, pp. 359-366, 1989.

[15] R. J. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres G, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang and F. Yasmeen, forecast: Forecasting functions for time series and linear models. R package version 8.15, https://pkg.robjhyndman.com/forecast, 2021.

[16] M. O'Hara-Wild, R. J. Hyndman and E. Wang, fable: Forecasting Models for Tidy Time Series. R package version 0.3.1, https://CRAN.R-project.org/package=fable, 2021.

[17] J. Mount and N. Zumel, vtreat: A Statistically Sound 'data.frame' Processor/Conditioner. R package version 1.6.3., https://CRAN.R-project.org/package=vtreat, 2021.

[18] S. J. Taylor, B. Letham, Forecasting at scale. The American Statistician, 72(1), pp. 37-45, 2018.

Preprint