IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 30, 2021, accepted June 15, 2021, date of publication June 29, 2021, date of current version July 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093481

A Novel Technique for Short-Term Load Forecasting Using Sequential Models and Feature Engineering

ABDUL WAHAB[1], MUHAMMAD ANAS TAHIR[1], NAVEED IQBAL[1], ADNAN UL-HASAN[1,2], FAISAL SHAFAIT[1,2], (Member, IEEE), AND SYED MUHAMMAD RAZA KAZMI[3], (Member, IEEE)

[1]School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad 44000, Pakistan
[2]Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad 44000, Pakistan
[3]SkyElectric Private Ltd., Islamabad 44000, Pakistan

Corresponding author: Syed Muhammad Raza Kazmi (raza.kazmi@ieee.org)

This work was supported by the Grants by the Higher Education Commission of Pakistan for National Center of Artificial Intelligence (NCAI).

ABSTRACT With the advent of smart grid, load forecasting is emerging as an essential technology to implement optimal planning and control of grid assets. Ergo in recent years, a significant thrust can be witnessed for the research towards the improvement of the prediction of the energy demand. However thus far there has not been any one technique in the literature that is shown to give best forecasts for a variety of sites; almost all the papers published on load forecasting, report their best results on just one of the dataset. This problem accentuates further when the training data does not have enough data points to learn patterns over all the seasons. Hence to devise a load forecasting technique that can yield the best estimates on diverse datasets, especially when the training data is limited, is a big challenge, which is addressed in this paper. The paper presents a novel combination of deep learning with feature engineering for short-term load forecasting. The proposed architecture, named as Deep Derived Feature Fusion (DeepDeFF), is based on the sequential model in conjunction with the hand-crafted derived features in order to aid the model for better learning and predictions. The raw data and the hand-crafted features are trained at separate levels, then their respective outputs are combined to make the final prediction. The efficacy and robustness of the proposed methodology is evaluated on diverse datasets from five countries with completely different patterns. The extensive experiments and results demonstrate that the proposed technique is superior to the existing state of the art.

INDEX TERMS Load forecasting, smart grids, deep learning, feature engineering, sequential models.

I. INTRODUCTION

Smart grid, in simple terms, implies monitoring and control of the power system's assets in the generation, transmission, distribution, and utilization, to achieve high efficiency and reliability at low operational costs. Several cardinal aspects of smart grid planning and control, such as the aggregation of distributed energy resources, economic scheduling of generation units, and demand side management etc., require the estimation of the upcoming energy demand [1], [2]. The load forecasting however is quite a complex problem and the prediction error for one site can be drastically different for the other not only due to the size of the available data but also due to the difference in demand profiles over diurnal, seasonal and yearly scale.

Artificial intelligence is fast becoming an enabling technology for data analytics and enhanced control of modern power systems. One of its most sought after application in recent times is the load forecasting through machine learning for predicting the trends in energy demand. This can lead to proactive optimization of control decisions to achieve higher energy efficiency, longevity of assets lifetime, and lower operational cost.

Long-term [3], mid-term [4], and short-term [5] are the different types of load forecasting found in the literature based on their duration of prediction from years to minutes.

The associate editor coordinating the review of this manuscript and approving it for publication was Arash Asrari.

The advantage of Short-term load forecasting (SLF) is that it provides better insight into the electricity consumption patterns and a greater degree of freedom for demand-side management. Also, SLF can be aggregated to get mid-term and long-term forecasts. Therefore this paper focuses only on SLF.

Despite having SLF being a preferred method for electricity load management, it is more difficult than the mid and long term forecasting because of the greater variance in the respective energy consumption patterns [6]. The high variance in the respective energy consumption patterns is due to the fact that the individual household energy consumption patterns can vary depending on the daily routines of the customers, and the short monitoring period producing a high frequency energy consumption graph. Another reason why individual house hold energy consumption forecasting is challenging is because of the unique nature of the energy consumption pattern of each household. This makes it hard to create a single generalized model that can adjust to the consumption patterns of each household. In this work the individual models for each household are trained for forecasting.

The deep learning based models, esp. the sequence models, are highly effective models for time-series analysis and forecasting. Their ability to process long range context and to traverse the data in both forward and backward directions (bi-directional sequence models) make them a suitable candidate for this purpose. The main challenge of using deep learning based algorithms is the availability of large and diverse data. To be able to utilize them effectively for load consumption forecasting, they should be able to process the following variations in the data:

- The data should be available for a season in multiple variations (data spread over years) so that there are variations in monthly consumption.
- The data resolution also contribute significantly in uncovering the underlying patterns, that is, the data on hourly or minute scale should be available.
- If the data for individual house loads is considered, then a wide variety (large sample size) of household should be present in the data.

The availability of dataset with above-mentioned properties is challenging and extremely hard to gather. The limitation of required data with sufficient diversity make it challenging to train a deep learning based sequential model and to exploit its prowess. The data availability for SLF is of few months data to train the model and then to predict the energy load for months that are not seen during training. This type of data is referred to as a limited dataset, where the test data contains features that were not seen by the model during training, thus making it harder for the deep learning models to learn the underlying patterns [6].

In practice, deep learning community tackle these limitations in a variety of ways.

- **Data Augmentation:** The first and foremost technique is to employ data augmentation in increasing the training data size by adding artificial data points. This techniques has been quite successful in computer vision domain where images can be rotated, zoomed, scaled, etc., to increase the data size. This technique has not been explored in time-series analysis.
- **Transfer Learning:** The second successful techniques employed in computer vision is transfer learning, where a model trained on a very large dataset is further fine-tuned with small subset of a new domain to exploit the already learned weights. There is no such general purpose models proposed in load forecasting domain.
- **Feature Engineering:** The third method reported in the literature to circumvent the dataset limitations is the use of hand-crafted features along with deep learning models. The deep learning models when aided with hand-crafted features are reported to perform better with limited dataset. The main hypothesis behind this method is to augment the deep learning algorithm with powerful statistical features so that the model can learn the underlying complex features in a better way.

The third method is adopted here to circumvent the limited data availability in case of short term load forecasting.

This paper presents a novel deep learning architecture that combines the use of hand-crafted features with raw data, such that the deep learning model can work well for SLF of small datasets. It also proposes to use Mean Average Percentage Error (MAPE) as the preferred *loss function*. The results demonstrate significant improvements in the performance, especially for limited datasets by using the proposed architecture.

In summary, following are the contributions of this research work.

- **Derived Handcrafted Features:**It is proposed to fuse both basic and derived features (explained in Section III). The proposed fusion resulted in better results as reported in V. It should be noted that our work could be extended to include more features like temperature and humidity and stats derived from these basic features to further improve the sequential models.
- **Novel Sequential Model:** A novel Y-Shaped sequential model based on Recurrent Neural Networks (RNN) and its variants Long Short-Term Memory (LSTM) Networks and Gated Recurrent Units GRU) has been reported. The proposed architecture fuses both basic and derived features and could be extended to include more features. Please refer to 3 for further details.
- **Better Loss Function:** It is proposed to use Mean Average Percentage Error (MAPE) as the loss function instead of Mean Absolute Error (MAE) while training the sequential models. The advantage of using MAPE as the loss function is because MAPE helps in better generalization of the model. It is observed that using MAPE as loss function strongly penalizes the model for predicting higher than actual values, thus making it follow the general pattern of the energy consumption, and ignore the outliers. This setting is important to deal with individual households that have high variances and

peaks in their energy consumption data. Using MAPE as loss function helps in getting a smooth and a reliable prediction.

This article is further organized as follows: Section II provides an overview of research work that has been done in this domain, Section III provides all the necessary details of the proposed DeepDeFF method. Section IV provides an introduction to all the datasets that have been used to evaluate the proposed methodology and Section V reports the results obtained on individual dataset along with comparative analysis with other reported methodologies. Section VI discusses the edge cases where our method fails and where it can be improved further. Section VII concludes the article with a brief summary and future outlook.

II. LITERATURE REVIEW

A significant amount of research has been carried out to develop SLF as the enabling tool for efficient monitoring and control of power system.

Santos *et al.* [7] proposed the use of feature engineering to design a feature vector by performing entropy analysis with a specific tolerance band and auto-correlation function. The designed feature vector was then passed through an artificial neural network (ANN) for prediction. Ferreira and da Silva used a Bayesian based approach to solve the complexity of neural network and variable selection [8]. The approach has theoretical ground but relies on various assumptions regarding the network parameters distribution and requires three relevance thresholds. Phase-space embedding method was used for the selection of input variable which allowed to include the preference of the past values of prediction quantity in the input vector [9]. A neural networks based approach to forecast next 24 hour load on medium and low voltage substations was presented in [10]. The use of separate models each for daily average power and for intraday variation in power, improved the accuracy of prediction compared to the model based on time series.

Cao *et al.* [11] adopted autoregressive integrated moving average (ARIMA) model and similar day method for intraday load forecasting. The mechanism of their similar day method is to group the targeted day with meteorologically similar days in the history and predict the load based on the average demand of those days. It was demonstrated that in ordinary days, ARIMA performs better while similar day method wins in unordinary days. Li *et al.* [12] demonstrated the use of extreme learning machines (ELMs) for short term load forecasting. The pitfall of a single ELM is that the output is usually unstable due to the randomness in its training. Zang *et al.* proposed an ensemble of extreme learning machines (ELMs) to forecast the total load of the Australian national energy market. The proposed methodology not only made use of the supreme ELM learning efficiency for self-adaptive learning but also used the ensemble structure to mitigate the instability of the forecasts. Recently, k-nearest neighbour (KNN) algorithm had also seen some successful examples on load forecasting. Al-Qahtani and Crone [13] proposed multivariate k-NN regression method and Zang *et al.* [14] proposed an ensemble of KNN models for day ahead load forecasting. The dominant advantage of using KNN algorithm is its efficiency.

Recently recurrent neural networks (RNN) have become the popular choice for load forecasting. In [15] machine learning models were used for predicting the energy demand on publicly available RTE dataset [16]. The performances of RNN and support vector machine (SVM) models were compared using different input features. The models were evaluated on a test set of 10 days of year 2017. The results demonstrated that RNN performed better, with a MAPE of 3.52%, compared to SVM with a MAPE of 14.00%.

A recent study [6] demonstrated how the individual household level load forecasting can be challenging because of different patterns of energy consumption of individual consumers [17]. A two layer LSTM model was proposed and compared with other models based on back-propagation neural network (BPNN), *k*-nearest neighbour (KNN), extreme learning machine (ELM) and input scheme combined with a hybrid forecasting framework (IS-HF). Individual models for each household were trained and the best average MAPE of 44.06% was achieved through LSTM. Alhussein *et al.* [18] proposed a Hybrid CNN-LSTM for individual household level load forecasting in SGSC. In the proposed approach the CNN was used to extract the complex relationship between the input variables while LSTM was used for sequence learning. They achieved the MAPE of 40.38%.

Electricity demand is influenced by weather, holiday, time of day, etc. Time dependant convolution neural network (TD-CNN) and cycle based long short term memory (C-LSTM) for short- and medium-term load forecasting was presented in [19]. Electric load on weekly basis was arranged in image format and fed to TD-CNN model. C-LSTM helped to extract time dependencies between sequences. The models performed better than the traditional LSTM model while reducing the training time.

Another important application of SLF is in energy trading, which is a complex process due to non-periodic variations in energy consumption. Accurate forecasting for hourly spot price is the key to achieve the best trading decision, which is vital for investors and retailers in electricity market. A model based on a hybrid approach comprising of ARIMA, multiple linear regression (MLR), and Holt-Winter model was proposed in [20]. The hybrid model was tested for Iberian electricity market dataset to forecast hourly spot prices for various numbers of days. A hybrid model based on non-linear regression and SVM was proposed in [21], that was tested on ERCOT data [22]. This hybrid model achieved MAPE of 7.30% compared to the individual models with 8.99% and 8.63% MAPE respectively.

Improvement of forecasting accuracy using standard LSTM model by feeding it processed features rather than raw data was proposed in [23]. The power load sequence was decomposed by complementary ensemble empirical mode decomposition (CEEMD), then the approximate entropy (AE) values of the obtained subsequences

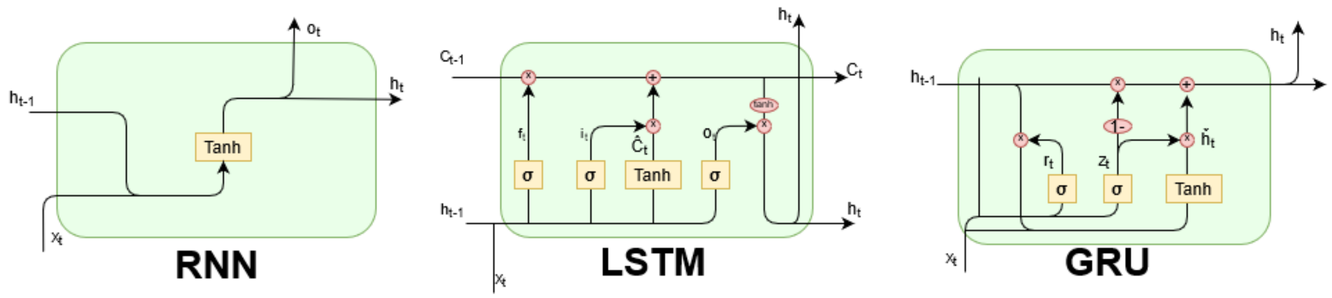


FIGURE 1. Internal architecture of individual cells in a sequence models. RNNs take input from current and previous timestamp and combine them using *tanh* activation function. An LSTM cell has three gates: *input, forget,* and *output*, while GRU cells have two gates: *reset* and *output*. Both LSTMs and GRUs were designed to overcome the so-called *Vanishing Gradient Problem*, that is the perseverance of long-term context. GRUs are relatively less complex and more efficient than LSTM Networks.

were calculated. The subsequences with similar AE values were merged into new sequence to form the inputs of the load forecasting model. This reduced the complexity of the power load sequence and improved the accuracy of load forecasting. The vanilla LSTM network was improved in [24] by cleaning and processing the raw load data using isolated forest algorithm.

Electric load forecasting requires training of large number of neurons in hidden layer, which increases the size of the network and slows overall training process. To reduce this overhead, a multi-column radial bias function (MCRF) with error correction algorithm designed to reduce the number of hidden neurons in a network, was proposed in [25]. It was shown that MCRF with only 50 neurons in hidden layer took only 10 minutes to train and achieved the MAPE of 4.59% compared to other models with more than 150 neurons that achieved better MAPE of 1.77% but took hours to train.

Accuracy of SLF can be improved through careful analysis of the load data to find the effectiveness of selected features. A technique was proposed in [26] for features selection where the bisecting K-means algorithm was used to cluster the load data with high similarity for a forecast date. The ensemble empirical mode decomposition (EEMD) helped to combine components with similar entropy. A bidirectional recurrent neural network (BRNN) model was proposed to forecast the load of the network. The model was verified on two datasets including a dataset from load forecasting competition. The results showed that BRNN model performed better even than the winner of the competition.

Recently introduced PRECON dataset [27] presents another phenomenon of power outages that is uncommon in developed countries. In [28], the authors explored the challenges of dealing with the problem of power outages while doing the short term load forecasting. The main challenge of including the power outage data is the presence of long range 0 KW in the data, thereby increasing the complexity of data. They achieved the Mean Percentage Error (MPE) of −496.70 using Support vector regression (SVR).

The use of sequence models and feature engineering has shown to increase the forecast accuracy. However both concepts have been explored separately. The proposed approach examines the effects of combining the concepts of feature engineering and sequence modeling in the context of Short Term Load Forecasting. This work, in essence, is aligned with that reported in Hybrid CNN-LSTM [18]; however, instead of asking a CNN to act as feature extractor that requires a lot of data, hand-crafted features have been used to train a novel sequence model architecture.

III. PROPOSED METHODOLOGY

Recently deep learning solutions [6], [19], [29], particularly sequential models such as RNN and LSTM models are becoming popular choices for load forecasting. Recurrent Neural Networks are a generalization of feedforward neural networks that have an internal memory. RNNs performs the same function for every input of data while the output of the current input depends on the previous inputs' computation. After producing the output, it is copied and sent back into the network. For making a decision, it considers the current input and the output from the previous inputs. RNNs take input from current and previous timestamp and combine them using *tanh* activation function. An LSTM cell has three gates: *input, forget,* and *output*, while GRU cells have two gates: *reset* and *output*. Simple RNN faces the problem of gradient vanishing and exploding and therefore cannot remember long sequences. Long Short-Term Memory (LSTM) [30] networks and Gated recurrent units (GRUs) [31] are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The GRU is similar LSTM with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. Fig. 1 describe the inner computations of each model in more detail.

A. BIDIRECTIONAL SEQUENTIAL MODELS

Bidirectional model trains forward and reverse nodes using respectively:

- input in positive time, i.e. the given input as it is,
- input in negative time, i.e. a time-reversed copy of the original input.

The advantage of bidirectional model compared to conventional ANN models is that it observes the input in both forward and reverse directions to extract more information from the input sequence (Fig. 2). This technique of negative time and bidirectional layer was first discussed in [32].

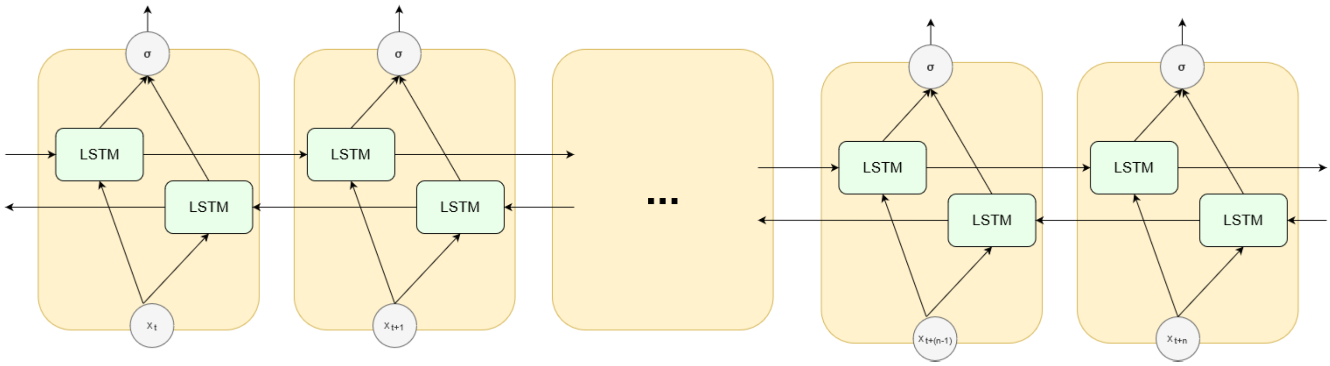


FIGURE 2. **Bi directional sequence models. The figure shows the example with LSTM cells, but these cells can be any of RNN or GRU types.**

This paper reports the results of using all three sequential models: LSTM, RNN, GRU as well as their bidirectional counterparts (BLSTM) [33], BRNN and BGRU on several datasets for a comprehensive comparison.

B. CONVERTING DATA INTO SEQUENCE

To understand how the model works, first it is important to understand how the input sequence is constructed. The input data can have the following attributes: time interval, date, month, energy load etc. The data has to be converted into a sequence so that it can be fed to the proposed architecture for the prediction of load at the next time interval. Sequential models like RNN, LSTM, BLSTM etc. require input data of past time-steps to extract their features and their temporal information to make prediction of the next time step. To make the data usable for training such a model, the data must be converted into input sequences. This conversion is done by concatenating the current input features f with the past $t-1$ input features, where f is the length of feature vector and K is the number of time-steps used for making the prediction.

C. HAND-CRAFTED FEATURE SELECTION

In this work, the input features are divided into two different types: basic and derived. The basic features include electricity load consumption for the residential customers, hour of the day, day of the week and holiday indicator. The derived features, as their names suggests, are calculated from the basic features. Both type of features are important in load forecasting. The basic features provides the absolute values of different parameters, while the derived features exploits the correlation between different values and help the deep learning models to learn the relationship or a pattern, esp. in situations where the data availability is limited (non-repetitive or sporadic data).

A deep learning model with enough computation time and data may extract derived features on its own, but this cannot be guaranteed within the constraints of time and resource. Thus providing these derived features explicitly as inputs can enable the model to learn more from the data and converge quickly. Generally the performance of deep learning models improve by increasing the number of relevant input features unless it starts to overfit.

The basic features used for generation of derived features and as input for the DeepDeFF model are:

- Energy load consumption E.
- Time-stamp of the day T, divided into 30 minutes interval each. The feature is converted into One-hot encoding.
- Current day of the week W, converted into one-hot encoding.
- Holidays represented by a binary label H. At the moment only weekends are marked as holidays, but in future work this can be expanded and synced with other public and national holidays.

There could be other basic features, for example, temperature and humidity values for each day, but that has not been considered in this work to remain consistent with the other referenced papers.

Derived features are calculated for each record in the input sequence $(1, K, f)$, where K represents the number of past records used for creating the input sequence and f represents the basic features. Only the derived features for the energy load consumption, E, have been considered in this work. Following are the derived features that are calculated and used as input to the DeepDeFF model.

- Average load consumption of K time-steps.
- Standard deviation of load consumption of K time-steps.
- Average load consumption of the time-stamp t that is to be predicted, for past K days.
- Standard deviation of load consumption of the time-stamp t that is to be predicted, for past K days.

D. PROPOSED ARCHITECTURE

This paper proposes a Y-Shaped sequential model architecture DeepDeFF (Fig. 3 shows the schematics of the DeepDeFF architecture), which inputs the raw and derived input features into separate layers to extract learned features. The idea behind using separate input layers for basic and derived sequences is to allow the sequential layers to learn from the two input sequences independently. The goal is to exploit the relevance of basic and derived sequences with the predictions individually. The learned representation from the individual sequential layers is then merged and fed to a dense layer. It is noted that both sequential layers in our Y-shaped sequential

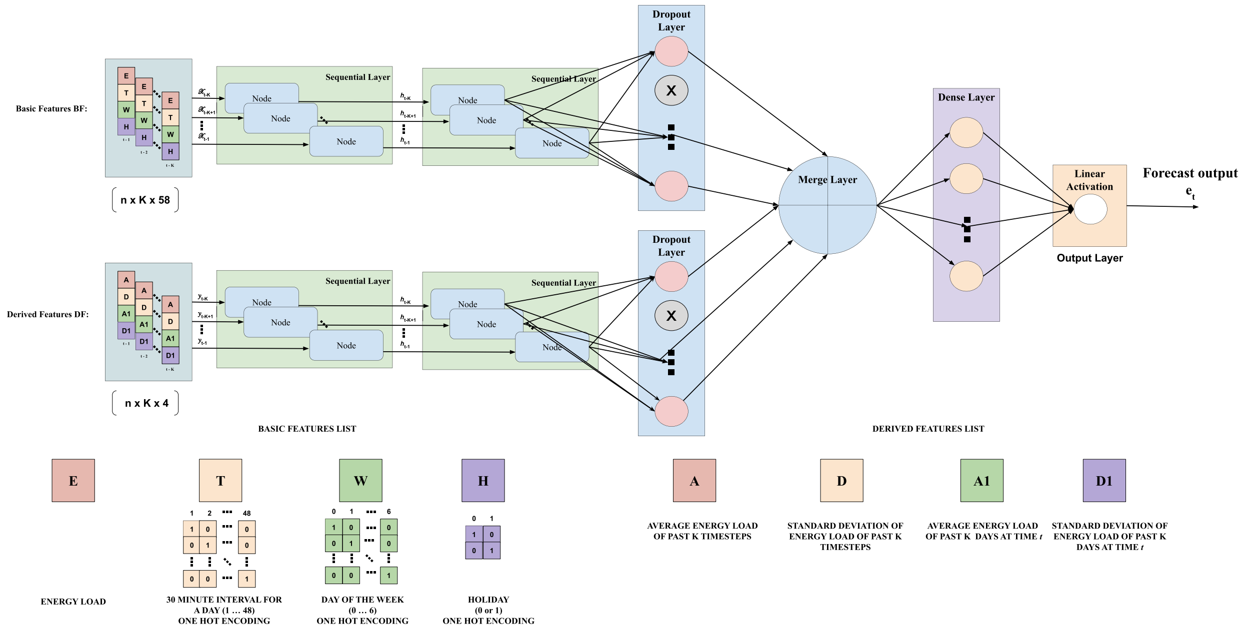


FIGURE 3. Proposed System Architecture: The input data is first pre-processed to achieve the derived input features. The raw input features T, W, H are converted to one-hot encoding. The raw and derived input features are then fed to two individual bi-directional sequential layers. The information extracted from these separate layers is then merged and used as input to a dense layer followed by a final feed forward layer with *Linear* activation function.

model are identical. The dense layer or fully connected layer consists of n nodes with Rectified Linear Unit (*ReLU*) activation, the best value of n is determined experimentally. The purpose of dense layer is to non-linearly merge the information extracted from basic and derived features. The output of a dense layer is processed through a linear activation output layer to make the final prediction of the load at the next time interval.

E. HYPER PARAMETER SELECTION

The focus of experiments and results presented in this paper is to demonstrate the efficacy of the proposed technique for various datasets, orthogonal to the hyper-parameters optimization. The tuning of hyper-parameters is not extensively explored here. The *Adam* optimizer, *No. of hidden layers* and *No. of nodes in a hidden layer* have been adopted from [6], which provides the rationale for the selection of these components. However the choice for the internal parameters of Adam optimizer, such as *epsilon*, *amsgrad*, β_1, is not mentioned in [6]. The *Epsilon* effects the numerical stability, this parameter is not considered for tuning here. For the rest of the parameters, the authors have performed experiments on SGSC dataset (the datasets are described in the next section) over a range of candidate values to find the optimal ones yielding best results. The *amsgrad* is a boolean - whether to use the AMSGrad variant of the Adam algorithm or not - so the experiments were performed with both *False (default value in Keras implementation)* and *True* values; the default value resulted in better *MAPE* score. The value of β_1 - exponential decay rate of the moment estimates - is swept from 0.5 to 0.99. Although the best *MAPE* was found at the value of 0.95, but here in this paper 0.9 is chosen which happens to be the default value in *Keras* and also sets the same benchmark for comparison with the reference paper [6].

Consequently the hyper-parameter settings in this manuscript is as follows:

(a) 20 nodes sequential layer
(b) a dropout of 0.2
(c) Adam optimizer
(d) MAPE as loss function
(e) Learning rate: 0.001

Same set of hyper-parameters is then kept for all the datasets in this manuscript for the very reason to prove that even without an extensive hyper-parameter tuning, the proposed technique can yield superior results on diverse datasets. So by keeping the same architecture for each dataset and yet achieving the results better than the published state-of-the-art, the supremacy of the proposed framework is truly established. Nonetheless, the hyper-parameters tuning can be carried out in future work.

IV. THE DATASETS

The proposed methodology for SLF has been evaluated on five energy load datasets from different sources. This section provides the salient parameters of the dataset and presents the pre-processing technique adopted for each.

A. SMART GRID SMART CITY (SGSC) DATASET

SGSC project was initiated by the Australian Government in year 2010 [17]. It gathered smart-meter data from around 78,000 customers for a period of 4-years. In [6], individual models for each customer was proposed. However since it is not feasible to train individual models for ~78,000 customers, therefore 69 customers having "hot water system" were selected. The same subset is extracted here to evaluate the DeepDeFF architecture.

B. THE ALMANAC OF MINUTELY POWER DATASET (AMPds)

AMPds [34] contains electricity, water and natural gas measurements of a single Canadian household with 19 appliances, recorded for 1 year with 1 minute resolution, which is down-sampled to 30 minutes resolution [29]. The variables for raw features used here are the same as for SGSC except that E here is assigned to the Ampere reading.

C. RÉSEAU DE TRANSPORT D'ÉLECTRICITÉ (RTE) FRANCE DATASET

RTE dataset [16] is also used here to evaluate the proposed technique. The dataset used spans from year 2013 to 2016 with the sampling interval of 30 minutes. The raw inputs are programmed with same variables as for SGSC above.

D. THE ELECTRIC RELIABILITY COUNCIL OF TEXAS (ERCOT) DATASET

ERCOT dataset [22] provides real time and historical statistics surrounding independent system operator (ISO) operations of the Texas region for a period of $\sim$5 years recorded every 1 hour. The raw features variables used here are the same as for SGSC except that the time T here ranges 1-24 since the resolution is 1 hour.

E. PAKISTAN RESIDENTIAL ELECTRICITY CONSUMPTION (PRECON)

PRECON dataset [27] records the electricity consumption patterns in a developing country for 42 households of varying financial status, daily routine and load profile. The data is collected with 1 minute interval from 01-06-2018 to 31-09-2019. The amount of data varies for each household due to different number and types of appliances that are selected for monitoring. This dataset also captures the problem of power outages rampant in developing countries. This is evident from several long 0KW data intervals. For raw features, same variables as in SGSC are used here except that E here refers to the KW usage.

V. RESULTS

The proposed framework for SLF is achieved through an evolutionary process after numerous rigorous experiments on all five datasets. This section discusses these experiments in sufficient detail and infers the results obtained. The results from the DeepDeFF architecture (all six variants) are compared with the results of other sequential models proposed in [6] and [18]. All six variants are same in overall architecture; the only difference between them is the use of different cell (RNN, LSTM, or GRU) in the sequential layer and their bi-directional counterparts (BRNN, BLSTM, or BGRU). The sequential layers, as mentioned previously, are identical in all experiments. Different cells can be used in both layers, but this work keeps all as same for simplicity.

The authors of this paper have re-implemented some other methods referred in the literature, such as, KNN, ELM, BPNN; the results have been reported on the same test data as used for DeepDeFF architecture. The authors also tested the methodology with other variants of sequential models -RNN, LSTM, GRU, BRNN, BLSTM and BGRU - similar to one proposed in [6] without using the hand-crafted features.

Following sub sections report the results and comparative analysis for different datasets individually. It is important to understand the difference between the reported results with respect to different timestamps. *2-timestamps* means that the future value is predicted using the previous 2 points only. Similarly, *6-timestamps* and *12-timestamps* mean that the future values are predicted using the previous 6 and 12 data points respectively.

For SGSC and AMPds datasets, the results have been reported for three different timestamps, 2, 6, and 12. However, the last three datasets, RTE, ERCOT, and PRECON are evaluated using only 2-timestamps because it was observed that the best predicted values corresponds to *2-timestamps* on first two datasets.

A. SGSC DATASET

1) TRAIN & TEST SETTING

The same settings provided in [6] are used to extract the subset of SGSC data for fair comparison on the same test set. The data spanning the whole winter season of New South Wales Australia is subdivided into a split ratio of 0.7/0.2/0.1 as:

(a) Training set (01-Jun-2013 to 05-Aug-2013)
(b) Validation set (06-Aug-2013 to 22-Aug-2013)
(c) Test set (23-Aug-2013 to 31-Aug-2013)

The first set is to train the DeepDeFF model, validation set is used to select the best model weights based on performance on validation set, while the test set is for the evaluation of the DeepDeFF model. The data is spaced between 30 minutes interval; so for 69 customers the 9 days of evaluation implies the forecasting of 29,808 time points.

2) RESULTS

Table-1 shows the comparison of results from rigorous experiments that are performed on SGSC dataset using the proposed DeepDeFF method in contrast with the implementation of the LSTM model proposed in [6], and its extended variants that use GRU, RNN, and their bi-directional counterparts. The addition of derived features in the proposed architecture along with MAPE as loss function, outperforms the state of the art on the SGSC dataset as evident from the average MAPE computed in Table-1.

B. AMPds

1) TRAIN & TEST SETTING

The AMPds data is converted from 1 minute resolution to 30 minutes, yielding 17,483 data points [29]. The data is subdivided with a split ratio of 0.7/0.2/0.1 into:

(a) Training set (01-Apr-2012 to 17-Dec-2012)

TABLE 1. Results achieved on SGSC dataset.

Method	Avg. MAPE %		
	2-timestamps	6-timestamps	12-timestamps
DeepDeff BGRU	**34.87**	36.02	36.48
DeepDeff BLSTM	35.40	36.41	37.74
DeepDeff BRNN	35.79	38.60	39.64
DeepDeff GRU	35.01	**35.72**	**36.34**
DeepDeff LSTM	36.88	37.28	38.09
DeepDeff RNN	35.94	38.97	40.57
BGRU	42.83	42.08	42.40
BLSTM	43.04	42.96	43.88
BRNN	42.74	42.61	43.20
GRU	42.85	41.78	41.54
LSTM [6]	44.39	44.31	44.06
RNN	42.04	42.28	43.29
CNN-LSTM [18]	40.38	41.07	42.85
KNN	74.83	71.19	81.13
ELM	122.90	136.49	123.45
BPNN	49.62	49.04	49.49

TABLE 2. Results achieved on AMP dataset.

Method	Avg. MAPE %		
	2-timestamps	6-timestamps	12-timestamps
DeepDeff BGRU	25.08	**24.64**	25.19
DeepDeff BLSTM	25.44	25.98	26.62
DeepDeff BRNN	**24.77**	25.47	25.35
DeepDeff GRU	25.48	25.33	**25.17**
DeepDeff LSTM	25.57	26.00	25.81
DeepDeff RNN	25.62	25.32	25.87
BGRU	28.17	31.58	32.85
BLSTM	28.14	32.32	36.08
BRNN	26.88	30.10	31.62
GRU	27.65	30.49	30.00
LSTM	28.57	33.47	30.65
RNN	26.26	27.97	29.54
KNN	30.61	30.38	30.66
ELM	43.88	44.05	44.16
BPNN	34.01	37.83	38.71

(b) Validation set (18-Dec-2012 to 23-Feb-2013)
(c) Test set (24-Feb-2013 to 01-Apr-2013)

2) RESULTS

Table-2 shows a comparison of the results produced by the simple two layer sequential model and the DeepDeFF architecture with derived features. The proposed architecture beats the benchmark of 26.23% achieved in [29] for 6 time-steps.

Fig. 4 shows that the DeepDeFF architecture performs well in predicting the general load and suffers in case of outliers. This is because the model was able to learn the underlying general pattern from the training data, and gave it more importance than to outliers. This problem occurred because the training data was not enough and does not cover all the months; so the test data is of a month that was never seen during training.

C. RTE DATASET

1) TRAIN & TEST SETTING

RTE data is subdivided into three subsets with a split ratio of 0.7/0.2/0.1 as:

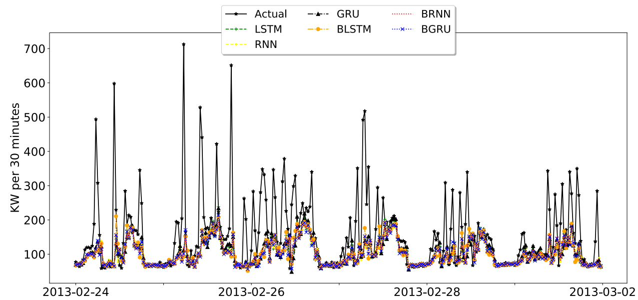


FIGURE 4. Actual load versus load predicted by DeepDeFF architecture utilizing different sequential layers for AMPds dataset.

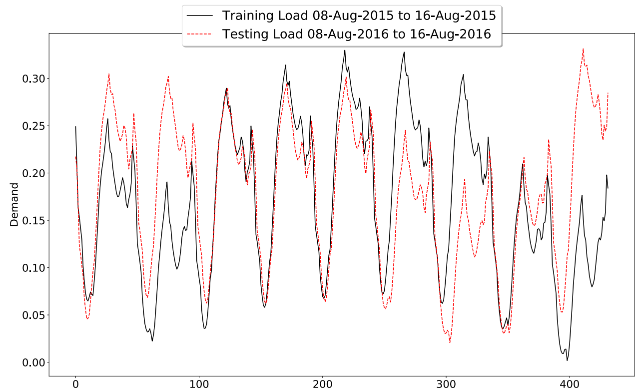


FIGURE 5. Training and testing data comparison, RTE dataset.

(a) Training set (01-Jan-2013 to 18-Nov-2015)
(b) Validation set (19-Nov-2015 to 07-Aug-2016)
(c) Test set (08-Aug-2016 to 31-Dec-2016)

Fig. 5 show the subsets of training and testing data for the dates mentioned in the figures' legends. Such close resemblance in the test and train data helps the model to make accurate predictions as evident from the results.

2) RESULTS

It is observed from the results for SGSC and AMPds datasets that the experiments with 2 time-steps mostly yield the best results. Henceforth 2 time steps is used for the experiments on other datasets. Table-3 shows the results for RTE dataset. The proposed model with GRU and derived features performed best with average MAPE of 0.81%. Fig. 6 shows the prediction results against the actual system load which further confirms the excellent performance of the DeepDeFF architecture.

D. ERCOT DATASET

1) TRAIN & TEST SETTING

ERCOT data is subdivided into three subsets with a split ratio of 0.5/0.1/0.4 as:

(a) Training set (01-Jan-2011 to 26-May-2013)
(b) Validation set (27-May-2013 to 31-Dec-2013)
(c) Test set (01-Jan-2014 to 31-Dec-2015)

Similar to RTE, ERCOT is also the accumulated load consumption data of Texas. The train and test data for ERCOT also has close resemblance similar to Fig. 5.

TABLE 3. Results achieved on RTE, ERCOT and PRECON datasets.

Method	Avg.MAPE % - 2 timestamp		
	RTE	ERCOT	PRECON
DeepDeff BGRU	0.84	**0.91**	21.87
DeepDeff BLSTM	1.63	0.98	22.00
DeepDeff BRNN	1.26	0.92	**21.67**
DeepDeff GRU	**0.81**	1.38	22.10
DeepDeff LSTM	1.17	1.17	22.18
DeepDeff RNN	1.39	1.38	21.89
BGRU	1.06	1.40	24.01
BLSTM	1.04	2.58	24.18
BRNN	1.13	1.37	23.90
GRU	1.09	1.76	24.04
LSTM	1.19	1.96	24.31
RNN	1.14	2.88	24.11
KNN	1.17	6.11	40.43
ELM	1.21	1.85	44.51
BPNN	1.20	6.01	30.02

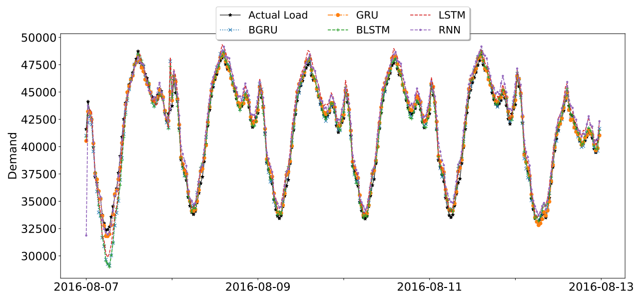


FIGURE 6. Actual load versus load predicted by DeepDeFF architecture utilizing different sequential layers on RTE dataset.

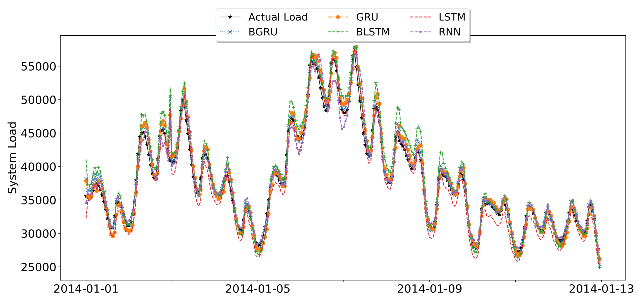


FIGURE 7. Actual load versus load predicted by DeepDeFF architecture utilizing different sequential layers on Ercot dataset.

2) RESULTS

Table 3 shows the results for ERCOT dataset where the DeepDeFF model with BGRU performed best with average MAPE of 0.91%. Figure-7 shows the results that establishes the effectiveness of the DeepDeFF architecture.

E. PRECON

1) TRAIN & TEST SETTING

Owing to the peculiar nature of the PRECON dataset, it is pre-processed in two steps in this research. First, the data is converted from 1-minute interval to 30-minute intervals by taking the average over the 30 consecutive load readings. The second step is to take care of close to zero values in the data that are mostly due to power outages. Otherwise these values cause divide-by-zero problem when using MAPE function for evaluation, resulting into unrealistically high MAPE and adversely affecting the performance of the machine learning algorithm. This is countered simply by adding a small offset of 0.1 KW to all the readings. The offset is small enough to makes no significant change in the nominal values of the load and takes effect only for the near zero data. This simplest pre-processing has shown remarkable impact on the performance of the DeepDeFF algorithm as evident from the results.

The data splitting is done in a unique way here due to the reason that it spanned over a period of only one year with no repeated data for any month. So instead of using an overall split of data, as done in previous datasets, a month-wise split is proposed. Here the training, validation, and testing data is taken from days 1 – 21, 22 – 26, 27 – 30/31 respectively for each month. This corresponds roughly to an overall split of 0.7/0.2/0.1.

2) RESULTS

Table 3 shows the comparison of results obtained for PRECON dataset. DeepDeFF models have consistently outperformed the basic models on all the houses, achieving on average 8.9% lower MAPE than the basic models.

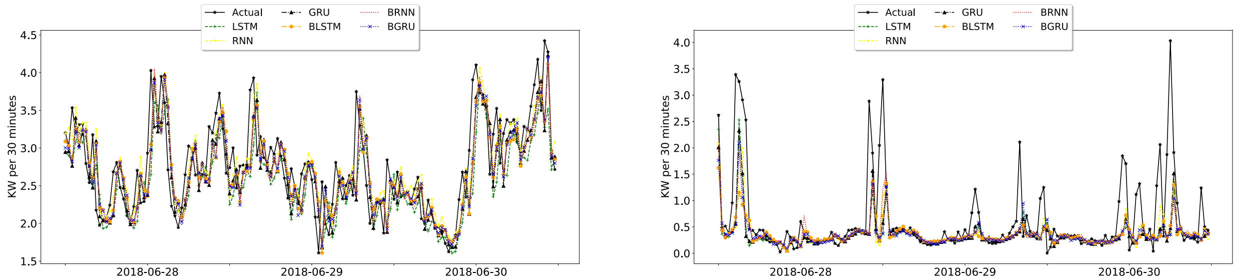
The value of the MAPE achieved by DeepDeFF models ranged from 7.67% on House 3 to 37.61% on House 29. The graphs of predicted versus actual load of these two houses are shown in Fig. 8a and Fig. 8b respectively.

VI. DISCUSSIONS

It can be observed from the results reported in Section V that the proposed methodology consistently outperforms previously reported results. It also proves our hypothesis that the fusion of hand-crafted features, both basic and derived, indeed increases the efficacy of deep learning models. Previous works usually report results on a single dataset; however, to the best of our knowledge, this is the first work that compares a single deep learning architecture across five (05) datasets from different countries under different energy consumption patterns and across different time-ranges.

The results also show that some datasets are difficult than others for SLF. This pertains to the fact that individual households (SGSC, APMds, and PRECON) have high variance in load consumption in comparison to those for country or a state (RTE and ERCOT). The effect of high variance in load consumption makes it difficult for any forecasting method to learn the underlying pattern. The effect of high variance can be overcome by using more data over wide range, such as data covering multiple seasons.

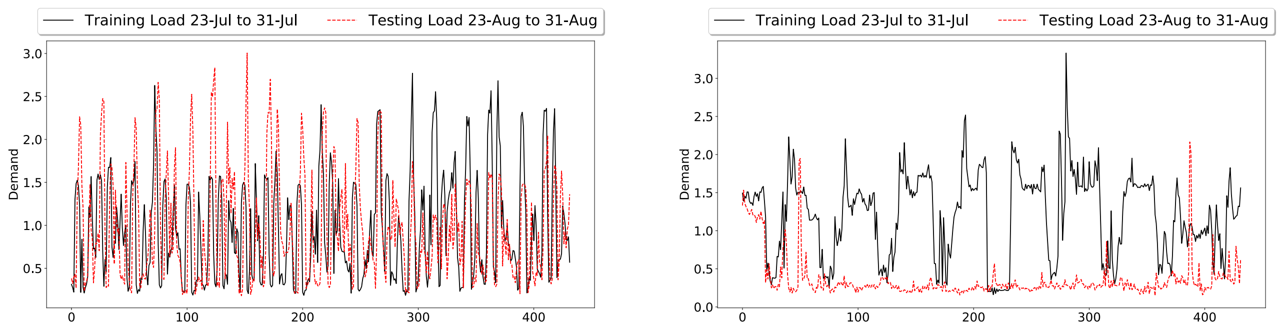
Despite better results across all five datasets than the previous state-of-the-art, the current methodology also fails in some scenarios of individual household energy forecasting. For SGSC, the best performing model is *DeepDeFF BGRU*



(a) Prediction of DeepDeFF models vs actual load for House 3

(b) Prediction of DeepDeFF models vs actual load for House 29

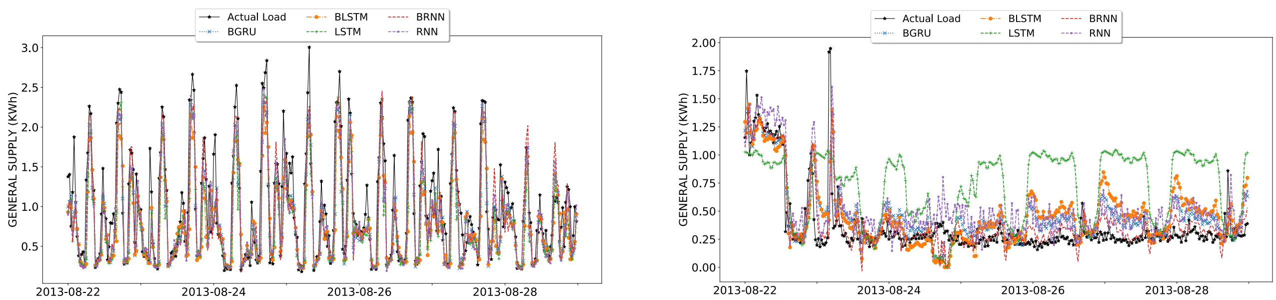
FIGURE 8. **Actual load versus load predicted by DeepDeFF architecture utilizing different sequential layers on PRECON dataset.**



(a) load pattern for customer 8804804

(b) load pattern for customer 8655993

FIGURE 9. **A comparison of train and test load data for two types of customers from SGSC dataset is shown, (a) shows a customer whose train and test data patterns have high correlation, (b) shows a customer who has very different load patterns for train and test data.**



(a) Predicted vs. actual data for customer 8804804

(b) Predicted vs. actual data for customer 8655993

FIGURE 10. **Actual load versus load predicted by DeepDeFF architecture utilizing different sequential layers for the two customers shown in Fig.9.**

with an average MAPE of 34.87%. However, upon close observation of failure cases, it can be witnessed that the proposed method did not perform better because of the anomalous customer behavior during test days, and because of the lack of the training data available for that month and dates. Fig. 9 provides some insight into the diversity of customers of SGSC data by juxtaposing the train and test data. Test data is plotted over training data with matching numeric dates. Fig. 9a shows the data of a customer with similarity between train and test data patterns, whereas Fig. 9b shows no similarity for another customer. This indeed effects the results of DeepDeFF architecture, which is reflected in their respective MAPE of 26.04% and 50.78% using BLSTM layer; thus the DeepDeFF model has been able to learn the underlying patterns and temporal relations for Fig. 9a but not as good for Fig. 9b. Disjoint customer behaviour during training and testing days along with the lack of training data for corresponding month and date also resulted in poor performance in some cases. Fig. 10a confirms that the DeepDeFF model indeed predicts the actual load very well for customer 8804804. However, Fig. 10b shows that the model under performs for customer 8655993 due to uncorrelated train and test data. These problems can be solved by including more data into training, such that the training set covers the entire year. More features such as temperature, humidity, climate, season, etc., can also help in better generalization and improve the performance of the model.

For *AMPds* dataset, DeepDeFF BGRU provides the best forecast with 24.64 MAPE value when taking 6 previous timestamps into consideration. However, it was observed that random availability of certain high wattage load makes the data spike randomly, thereby, resulting in more errors.

For *RTE* dataset, DeepDeFF GRU model (uni-directional) provided the best results with a MAPE of 0.81 and for *ERCOT* datatset, DeepDeFF BGRU outperform other methods with a MAPE of 0.91. Both of these dataset are highly smooth having low variance; therefore, sequential models have performed very well.

The *PRECON* is a special dataset because it provides an additional challenge of data with 0 *KW* entries due to excessive and sporadic power outages during network overloads, resulting into *divbyzero* problem during MAPE calculation. It is shown that such datasets can be dealt with a simple offset to avoid the *divbyzero* scenario. This simple modification helped in successful application of sequential model. The DeepDeFF BRNN model came out as winner with a MAPE of 21.67. The sudden random spikes in the load consumption resulted in outliers, as in the case of Fig. 8b, resulting in high MAPE for such houses as compared to rest of the houses.

Observing the results more closely, one can see that all sequential models enriched with hand-crafted features perform better than the simple application of sequential models. It has been successfully demonstrated that the fusion of hand-crafted derived features augmented with MAPE as a loss function can be utilized in short-term load forecasting across many type of datasets.

This work can be extended in multiple directions. One can consider hype-parameters optimization for each dataset in conjunction with other features such as temperature, humidity, climate, season, holidays etc., and related derived features to extend a more powerful DeepDeFF model especially for cases where random peaks are experienced in the data (Fig. 4 and Fig. 8b). Also, the issue of data limitation can be addressed further by data augmentation in time-series domain and transfer learning.

VII. CONCLUSION

Load forecasting is of critical importance to optimally schedule and reliably manage the operations of power systems. This manuscript presented a deep learning architecture based on sequential layers, and a pre-processing method for introducing hand-crafted features into the end-to-end learning pipeline of the deep learning model, for short-term load forecasting. It is demonstrated with rigorous experimentation that the inclusion of hand-crafted features has improved the learning and predictions of the model. The proposed DeepDeFF architecture has been comprehensively tested on five different datasets – two country/state wide datasets and three household datasets. The results achieved from the proposed methodology have shown to outperform the current benchmark of these datasets for short term load forecasting. These results can be verified from the data and the code pertaining to this paper, downloadable via: https://github.com/manastahir/Short-Term-Load-Forecasting.

ACKNOWLEDGMENT

(Abdul Wahab and Muhammad Anas Tahir contributed equally to this work.)

REFERENCES

[1] A. Ahmad, N. Javaid, A. Mateen, M. Awais, and Z. A. Khan, "Short-term load forecasting in smart grids: An intelligent modular approach," *Energies*, vol. 12, no. 1, p. 164, Jan. 2019.

[2] P. B. Luh, L. D. Michel, P. Friedland, C. Guan, and Y. Wang, "Load forecasting and demand response," in *Proc. IEEE PES Gen. Meeting*, Jul. 2010, pp. 1–3.

[3] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 456–462, Jan. 2014.

[4] H. M. Al-Hamadi and S. A. Soliman, "Long-term/mid-term electric load forecasting based on short-term correlation and annual growth," *Electr. Power Syst. Res.*, vol. 74, no. 3, pp. 353–361, Jun. 2005.

[5] B.-M. Hodge, D. Lew, and M. Milligan, "Short-term load forecast error distributions and implications for renewable integration studies," in *Proc. IEEE Green Technol. Conf. (GreenTech)*, Apr. 2013, pp. 435–442.

[6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[7] P. J. Santos, A. G. Martins, and A. J. Pires, "Designing the input vector to ANN-based models for short-term load forecast in electricity distribution systems," *Int. J. Electr. Power Energy Syst.*, vol. 29, no. 4, pp. 338–347, May 2007.

[8] V. H. Ferreira and A. P. A. D. Silva, "Toward estimating autonomous neural network-based electric load forecasters," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1554–1562, Nov. 2007.

[9] I. Drezga and S. Rahman, "Input variable selection for ANN-based short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1238–1244, Nov. 1998.

[10] N. Ding, C. Benoit, G. Foggia, Y. Bésanger, and F. Wurtz, "Neural network-based model design for short-term load forecast in distribution systems," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 72–81, Jan. 2016.

[11] X. Cao, S. Dong, Z. Wu, and Y. Jing, "A data-driven hybrid optimization model for short-term residential load forecasting," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Oct. 2015, pp. 283–287.

[12] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electr. Power Syst. Res.*, vol. 122, pp. 96–103, May 2015.

[13] F. H. Al-Qahtani and S. F. Crone, "Multivariate k-nearest neighbour regression for time series data—A novel algorithm for forecasting UK electricity demand," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–8.

[14] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite k-nearest neighbor model for day-ahead load forecasting with limited temperature forecasts," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Jul. 2016, pp. 1–5.

[15] P. Theile, A.-L. Towle, K. Karnataki, A. Crosara, K. Paridari, G. Turk, and L. Nordstrom, "Day-ahead electricity consumption prediction of a population of households: Analyzing different machine learning techniques based on real data from RTE in France," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2018, pp. 1–6.

[16] RTE. (2019). *Grid Data*. Accessed: Aug. 27, 2019. [Online]. Available: https://data.rte-france.com/

[17] (2019). *Smart-Grid Smart-City Customer Trial Data*. Accessed: Sep. 10, 2019. [Online]. Available: https://data.gov.au/data/dataset/4e21dea3-9b87-4610-94c7-15a8a77907ef

[18] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM model for short-term individual household load forecasting," *IEEE Access*, vol. 8, pp. 180544–180557, 2020.

[19] L. Han, Y. Peng, Y. Li, B. Yong, Q. Zhou, and L. Shu, "Enhanced deep networks for short-term and medium-term load forecasting," *IEEE Access*, vol. 7, pp. 4045–4055, Jan. 2019.

[20] D. Bissing, M. T. Klein, R. A. Chinnathambi, D. F. Selvaraj, and P. Ranganathan, "A hybrid regression model for day-ahead energy price forecasting," *IEEE Access*, vol. 7, pp. 36833–36842, Mar. 2019.

[21] Z. Ma, H. Zhong, L. Xie, Q. Xia, and C. Kang, "Month ahead average daily electricity price profile forecasting based on a hybrid nonlinear regression and SVM model: An ERCOT case study," *J. Mod. Power Syst. Clean Energy*, vol. 6, no. 2, pp. 281–291, Mar. 2018.

[22] ERCOT. (2019). *Grid Data*. Accessed: Aug. 27, 2019. [Online]. Available: https://ercot.com/

[23] J. Cui, Q. Gao, and D. Li, "Improved long short-term memory network based short term load forecasting," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 4428–4433.

[24] Y. Ma, Q. Zhang, J. Ding, Q. Wang, and J. Ma, "Short term load forecasting based on iForest-LSTM," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 2278–2282.

[25] A. O. Hoori, A. A. Kazzaz, R. Khimani, Y. Motai, and A. J. Aved, "Electric load forecasting model using a multicolumn deep neural networks," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6473–6482, Aug. 2020.

[26] X. Tang, Y. Dai, Q. Liu, X. Dang, and J. Xu, "Application of bidirectional recurrent neural network combined with deep belief network in short-term load forecasting," *IEEE Access*, vol. 7, pp. 160660–160670, 2019.

[27] A. Nadeem and N. Arshad, "PRECON: Pakistan residential electricity consumption dataset," in *Proc. 10th ACM Int. Conf. Future Energy Syst. (e-Energy)*, New York, NY, USA, Jun. 2019, pp. 52–57.

[28] A. Nadeem and N. Arshad, "Short term load forecasting on PRECON dataset," in *Proc. Int. Conf. Adv. Emerg. Comput. Technol. (AECT)*, Feb. 2020, pp. 1–6.

[29] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[31] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, pp. 1–15, Jun. 2014.

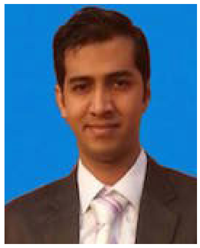
[32] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[33] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.

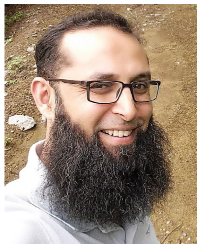
[34] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *Proc. IEEE Electr. Power Energy Conf.*, Aug. 2013, pp. 1–6.

ABDUL WAHAB was born in Pakistan, in 1998. He received the B.S. degree in computer science from the National University of Sciences and Technology (NUST), in 2020. His research interests include deep neural networks, sequence models, and forecasting.

MUHAMMAD ANAS TAHIR was born in Pakistan, in 2000. He is currently pursuing the B.S. degree in computer science with the National University of Sciences and Technology (NUST). His research interests include deep neural networks, sequence models, and forecasting.



NAVEED IQBAL received the B.S. degree in information technology from National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2011, where he is currently pursuing the master's degree in image processing and remote sensing domain. He has previously worked in industry as a senior software developer, led the team for android and iPhone application development. His research interests include deep learning, remote sensing, data analytics, and image processing for crop classification/health monitoring.



ADNAN UL-HASAN received the B.S. degree in electrical engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2002, the M.S. degree in systems engineering from the Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan, in 2004, and the Ph.D. degree in computer science from the University of Kaiserslautern, Germany, in 2016.

From 2014 to 2016, he was a Research Assistant with the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern. He is currently working as a Team Lead with the Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad. His research interests include deep neural networks, sequence models, information extraction, and forecasting.

FAISAL SHAFAIT (Member, IEEE) received the Ph.D. degree (Hons.) in computer engineering from the Technical University of Kaiserslautern (TUKL), Germany in 2008. He has more than 15 years of research and teaching experience in artificial intelligence and machine learning with the primary focus on computer vision and document image analysis. He is currently working as a Professor and is heading the Department of Computing, School of Electrical Engineering and Computer Science (SEECS), NUST, Pakistan, and the Deep Learning Laboratory (DLL) under the National Center of Artificial Intelligence (NCAI), NUST. He has made a significant international reputation in the field of artificial intelligence by earning more than 8000 citations with i10 and H-indices of 142 and 43 (Google Scholar), respectively. He received the prestigious IAPR/ICDAR Young Investigator Award by the International Association of Pattern Recognition (IAPR), in 2019. He has recently been included in the list of the World's Top 2% Scientists compiled by Stanford University.

SYED MUHAMMAD RAZA KAZMI (Member, IEEE) was born in Lahore, Pakistan, in 1977. He received the bachelor's degree from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, in 1999, the master's degree (Hons.) from the University of New South Wales, Australia, in 2001, and the Ph.D. degree from Tohoku University, Japan, in 2010. He is currently associated with SkyElectric Private Ltd., Pakistan, as an International Product Lead. His research interests involve efficient control and power electronics for smart grid and renewable energy conversion systems.