

# Hybrid Models for Adaptive Allocation of Electricity for Households

Midhush Manohar T.K., Naveen Suresh, Srikumar Subramanian, Gowri Srinivasa

**Abstract:** *In this paper, we analyze, model, predict and cluster Global Active Power, i.e., a time series data obtained at one minute intervals from electricity sensors of a household. We analyze changes in seasonality and trends to model the data. We then compare various forecasting methods such as SARIMA and LSTM to forecast sensor data for the household and combine them to achieve a hybrid model that captures nonlinear variations better than either SARIMA or LSTM used in isolation. Finally, we cluster slices of time series data effectively using a novel clustering algorithm that is a combination of density-based and centroid-based approaches, to discover relevant subtle clusters from sensor data. Our experiments have yielded meaningful insights from the data at both a micro, day-to-day granularity, as well as a macro, weekly to monthly granularity.*

**Keywords:** *Time series, Forecasting, SARIMA, LSTM, RNN, Clustering*

## I. INTRODUCTION

The integration of the internet and electronic devices has led to an abundance of data from the Internet of Things (IoT) industry. Data obtained from these sensors, if leveraged properly, can yield useful patterns that could help in making informed business decisions. Examining the electricity domain, notable reliance on this resource results in high power demand. This necessitates planning of electricity resources well in advance to ensure a continuous supply of power both in the present and the future. Modeling, forecasting and clustering of electricity consumption is hence of paramount importance in deregulated electricity markets for stakeholders across various aspects of the market, including energy wholesalers, traders, retailers, and consumers. The ability to accurately forecast the future electricity consumption will ensure effective planning and efficient operations, ultimately leading to increased financial profits. Moreover, energy-related time series forecasting and clustering plays an important role in the planning and working of various sensor based devices, as well as the entire power

grid system [1]. Sensory information is typically data points through time, or time series. A time series is a sequence of real-valued signals or observations that are measured at successive time intervals. The application of time series data include areas such as economics, finance, environment, and medicine and often manifests in the form of streaming in many real-world systems. Time series prediction and clustering has been successfully used in a wide range of domains including speech analysis, noise cancellation, and stock market analysis [2].

Previous works on similar datasets with respect to time series analyses have followed different forecasting approaches. An autoregression model, AR(1) with a high pass filter was used to forecast data on monthly electric energy consumption in Lebanon [3]. When comparing this with ARIMA (Auto Regressive Integrated Moving Average), it was found that the latter has a lower error rate when predicting monthly and quarterly consumption of electricity in a typical household [4].

In ARIMA, differences between the output values are taken instead of the absolute values to make the data stationary. Predictions of weekly and monthly usage are performed using ARIMA. However, for making daily predictions using ARIMA on IoT data, the seasonal component must also be considered. Thus SARIMA could be used to model the data instead. When considering even shorter time-frames, predictive models will be inaccurate due to the high amount of variations. Hence, an alternate approach to extract meaningful insights from hourly data needs to be explored. Clustering shorter time slices would yield useful motifs that could later be analyzed in collaboration with domain knowledge to better model the data [5].

With the advent of higher computational resources, it is now feasible to employ artificial neural networks for tasks such as classification and prediction [6]. Furthermore, it has been shown that neural networks are capable of approximating any continuous function [7] and have hence been used in forecasting of financial data series [8]. Neural networks are beneficial over traditional time series prediction methods such as ARIMA and Box Jenkins in that neural networks can learn non-linear relationships between the inputs and outputs.

Amongst Neural Networks themselves, and in particular, LSTMs, there are several approaches (architectures) which can be applied for Time Series data. These include (1) Sequence to Sequence Architecture and (2) Ensemble LSTM Network via Adaptive Weighting.

**Revised Manuscript Received on December 15, 2019.**

\* Correspondence Author

**Midhush Manohar T.K.\***, Department of Computer Science and Engineering, PES University RR Campus, Bangalore, India. Email: mimosk25@gmail.com

**Naveen Suresh**, Department of Computer Science and Engineering, PES University RR Campus, Bangalore, India. Email: naveen.213@gmail.com

**Srikumar Subramanian**, Department of Computer Science and Engineering, PES University RR Campus, Bangalore, India. Email: srik8552@gmail.com

**Gowri Srinivasa**, PES Center for Pattern Recognition, Department of Computer Science and Engineering, PES University EC Campus, Bangalore, India. Email: gsrinivasa@pes.edu

There is often an emergence of motifs discovered as an output of clustering or time series analysis of data sets. The key problem to solve is to find an optimal set of patterns or clusters from a large time series data and label them in order to differentiate good patterns from anomalies, which consequently helps in automating anomaly detection of noisy data sets.

Closest solution to the problem therefore is to use density-based clustering schemes like DBSCAN and OPTICS. However, absolute density algorithms without contextual input about the metric to cluster data often fail to be effective in clustering data sets which have recurring patterns that exhibit very little variation. An ideal approach to enhance such approach is to provide a multi-step algorithm which can firstly take context specific distance metric for clustering and then perform refinement of the clusters to improve the efficiency of clustering data into representative clusters assisting in a clear pattern driven outcomes.

The proposed novel clustering approach primarily uses the above two premises to solve outcome driven clustering problem demonstrated on real-time datasets. It is a novel technique to do a parameter light, time series motif clustering, helping to find clusters of similar waveforms effectively on real-time series data.

This paper attempts to successfully model the usage of electricity in a common household by combining the above mentioned forecasting and clustering methods.

## II. RELATED WORK

Our work comprises two main tasks: modeling and forecasting and clustering. Relevant work reported in literature for these tasks on time series data are described below.

### A. Forecasting

The current explorations towards accurately forecasting electric power consumption with time series data have used two main approaches: (1) Conditional Restricted Boltzmann Machines and Factored Conditional Restricted Boltzmann Machine (CRBM) that is a recently proposed model for time series that has a rich, distributed hidden state and permits simple, exact inference. It has an efficient, approximate learning algorithm called contrastive divergence [9]. However it performs poorly with null values, as the content must be clearly specified by the user and (2) Least Square Regression that computes the least squares approximation of linear functions to data [10]. Although being highly efficient, this method is unsuitable for datasets with non linear terms. This is because these terms curve relatively slowly, and for inherently non linear processes it becomes increasingly difficult to find a linear model that fits the data well as the range of the data increases.

### B. Clustering

There exists a multitude of clustering algorithms that are capable of handling time series data. Among the most popular include hierarchical clustering [11]. The main advantage with hierarchical clustering is the fact that it does not require the number of clusters to be formed beforehand. This parameter can be very difficult to determine, and hence the method is

used for various time series clustering [12] [13]. However, hierarchical clustering drastically drops its performance when dealing with large time series datasets [14]. This scalability issue leads us to other classes of clustering algorithms. Pure density based approaches, such as DBSCAN and OPTICS exist [15]. However, these methods tend to cluster IoT data very loosely, and clusters obtained are not very pure. But most of these algorithms either rely on hard parameters or are not robust enough for noisy IoT data.

Some measures of similarity between time series data include Euclidean distance, Pearson's correlation coefficient and Dynamic Time Warping (DTW) distance. Euclidean distance is a good and inexpensive measure for similarity between time series data provided that there is nearly zero phase difference between the motifs to be discovered. However, this in practice is a challenging task, and generally, arbitrary slices are to be clustered. Euclidean distance takes an adverse hit in such a scenario and does not capture the similarity between time series data accurately. Pearson's correlation coefficient suffers from the same limitation.

DTW, however, is much more forgiving of phase differences and minor variations in the signal due to similar regions of the series being matched even if they are shifted in time. Although DTW computation is significantly more expensive, there exist methods to make its computation faster, such as implementing a restricting window for computation or FastDTW [16]. Hence, DTW is chosen as the measure of similarity for time series clustering.

## III. OUR APPROACH

We analyze Global Active Power data, which is household global minute-averaged active power.

### A. Dataset

The dataset is from the UC Irvine Machine Learning Repository [17]. It consists of 20, 75, 259 measurements gathered in a house located in Sceaux (7km from the center of Paris, France) between December 2006 and November 2010 (47 months). A sample of the dataset is given below:

Date	Time	Global Active Power (kilowatt)
16/12/2006	18:37:00	4.408
16/12/2006	18:38:00	2.912
16/12/2006	18:39:00	2.326

The dataset had instances where the values could not be recorded, and, in order to overcome this, nearest neighbour interpolation [18] was performed. After comparing and contrasting with different approaches towards interpolation [19], [20], this particular method was chosen due to the uniform nature of electricity data.

### B. Forecasting

We have approached the problem of forecasting using (1) the traditional SARIMA model, (2) an LSTM-RNN model and (3) a hybrid model that combines both SARIMA and LSTM-RNN.

1) SARIMA: Chujai et. al. compared the performance of ARIMA and ARMA models in the forecasting of electricity [4]. However, we note that SARIMA models data more accurately because it takes account of the seasonal pattern of data. While ARIMA considers all the points up-to a certain lag SARIMA extends this concept to include the lag terms which correspond to the time period of the season. The equations using backshift notation are:

SARIMA

$$(1 - \phi_1 B)(1 - \Phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \theta_1 B^4)\epsilon_t$$

ARIMA

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \dots + \theta_q B^q)\epsilon_t$$

The global active power readings sampled in one minute intervals were downsampled to intervals of day, week and month. ACF for PACF were used to find the starting point of order and the seasonal order of the SARIMA model. Finer adjustments were made by applying a grid search on nearby values. The SARIMA model was used to forecast values in each of these intervals.

2) LSTM: Long short-term memory (LSTM), a class of recurrent neural networks (RNNs), is particularly designed for sequential data. It is an algorithm that is trained using back-propagation through time, and overcomes the vanishing gradients problem [21], [22]. Instead of neurons, which are predominant in traditional neural networks, LSTM networks have memory blocks that are connected through layers, as explained in [23]. This structure of LSTMs makes it relatively insensitive to gaps in data, which gives it an upper hand over traditional neural networks. The LSTM approach begins with a BDS test, which is a portmanteau test for time based dependence in a series [24]. As the p-value for this test comes out to be lesser than 0.05 for daily, weekly and monthly data, it indicates that the chosen data is non-linear in nature, and it can be modelled with Neural Networks. In traditional applied machine learning, the data is split into a train set and a test set, to prepare the model and evaluate it respectively. Additionally, various validation methods are used to assess the effectiveness of the model, particularly in cases where it is required to mitigate overfitting. One such commonly used validation process is k-fold cross validation that repeats the evaluation process by systematically splitting the data into k groups, each given a chance to be a held out model. However, commonly used method cannot be applied to time series data. This is because they assume that there is no relationship between the observations, that each observation is independent. This is not true of time series data, where the time dimension of observations means that we cannot randomly split them into groups. Instead, one must split the data up and respect the temporal order in which values were observed. Thus, walk forward validation is used. As this methodology involves moving along the time series one-time step at a time, it is often called Walk Forward Testing or Validation. Additionally, because a sliding or expanding window is used to train a model, this method is also referred to as Rolling Window Analysis or a Rolling Forecast.

For the chosen dataset two different variants of the LSTM model has been used:

- A simple Vanilla LSTM model
- A CNN Encoder-Decoder LSTM

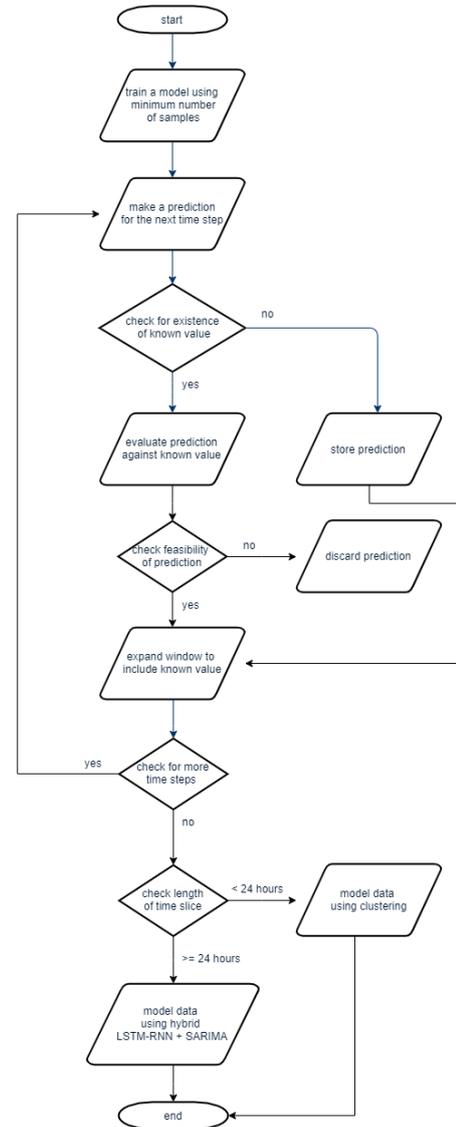


Fig 1. Walkforward Validation Pipeline

The encoder-decoder architecture uses two separate LSTMs, the encoder to read and encode the input sequence, and the decoder utilizes the encoded input sequence to make a one-step prediction for each element in the output sequence. The important difference is that an LSTM model is used in the decoder, allowing it to both know what was predicted for the prior day in the sequence and accumulate internal state while outputting the sequence. A convolutional neural network, or CNN, is used as the encoder in the CNN-LSTM architecture. The CNN does not directly support sequence input; instead, a 1D CNN is capable of reading across sequence input and automatically learning the salient features. These can then be interpreted by an LSTM decoder as per normal.

3) Hybrid Model: Regression models are used to model data that is linear. They cannot model non-linear data as effectively. On the other hand LSTM-RNNs are very good at modeling non-linear data. However LSTM-RNNs require large amounts of data in order to forecast accurately. Hence LSTM-RNNs are used here to complement the SARIMA model. Zhang et. al. used feed forward ANNs to complement ARIMA by modeling the residuals [25].

The LSTMs in this case are used to model the residuals to enhance the prediction accuracy of the existing SARIMA model. This is true of time slices of length 24 hours or greater. For readings at a scale of less than a day, clustering is employed. The entire pipeline is illustrated in 1.

### C. Clustering

Our clustering algorithm by itself requires the following parameters:

- Distance Matrix, containing the distances between different wavelets in the dataset ( $\Delta$ )
- Cluster Cut-off Size ( $\lambda$ )
- Distance Vicinity Threshold ( $\epsilon$ )

The cluster cut-off size is the minimum number of slices that a cluster should contain. It is dependent on the volume of data being clustered and can hence be heuristically determined for most purposes. The distance vicinity threshold,  $\epsilon$ , is used in the algorithm to determine if a point belongs to another's vicinity or not, i.e. for data points A and B,

$$A \in \mu_B, \Delta_{AB} \leq \epsilon$$

$$A \notin \mu_B, \Delta_{AB} > \epsilon$$

where  $\mu_X$  is the set of points in Xs vicinity.

This is used to calculate initial densities of points, as specified by the algorithm. A similar parameter is expected as an input to DBSCAN, which purely clusters points based on the density of points on the hyperplane.

1) Clustering Algorithm: Our clustering algorithm expects the distance matrix, the distance vicinity threshold and the cluster cut-off size (the minimum number of points in a group for the group to be considered a cluster). The algorithm can be divided into three parts:

- Pure cluster formation
- Removal of anomalies (points that cannot be clustered well)
- Reassignment of points from clusters that do not satisfy the cluster cut-off size

Pure density clusters are points which are objectively clustered based on the distance vicinity threshold. This is done by considering every point in the space as a centroid to its own cluster. Hence, there are as many clusters as points, and membership of points is decided by whether the point is within an  $\epsilon$  distance from the centroid. These clusters will have very similar points due to the clustering based purely on density, but points could potentially belong to multiple clusters.

Initially, fuzzy, but pure clusters are formed. Here, each point in the dataset acts as a cluster centre and all points within a distance vicinity threshold radius of a point is considered a part of the corresponding cluster. Hence, a point may belong to multiple clusters, but only N total clusters exist. If a point has less than the cluster cut off size number of points in its vicinity (as determined by the distance vicinity threshold), it is also marked as an anomaly. The clusters are

#### Algorithm 1 Pure Cluster Formation and Marking of Anomalies

**Input:** Distance Matrix  $\Delta[N]$ , Distance Vicinity Threshold  $\epsilon$ , Cluster cut-off size  $\lambda$   
**Output:** Pure Clusters C  
*Initialisation :*  
 1: C, densityBuffer This parameter can be difficult to heuristically estimate, and is key to the algorithms performance.  
*Initial Fuzzy Cluster Formation*  
 2: **for**  $i = 1$  to  $N$  **do**  
 3:   **for**  $j = 1$  to  $N$  **do**  
 4:     **if**  $(\Delta[i][j] \leq \epsilon)$  **then**  
 5:       Append Point j to densityBuffer  
 6:     **end if**  
 7:   **end for**  
 8:   **if** (Length of densityBuffer  $< \lambda$ ) **then**  
 9:     Mark Point i as Anomaly  
 10:   **end if**  
 11:   Append densityBuffer to C  
 12:   Clear densityBuffer  
 13: **end for**  
 14: Sort C in Descending Order of Length  
 15: **for**  $i = 1$  to  $N$  **do**  
 16:   **for**  $j = i$  to  $N$  **do**  
 17:      $C[j] = \text{set}(C[j]) - \text{set}(c[i])$  //set subtraction of larger cluster from smaller one  
 18:   **end for**  
 19: **end for**  
 20: Remove Empty Clusters from C  
 21: **return** C

then sorted in descending order by length. Now, an iterative set subtraction is applied on the clusters, wherein the smaller clusters are replaced with the set difference between the smaller cluster and the larger one. This removes points which exist in multiple clusters from the smaller of the clusters, leaving the larger clusters large while making the smaller clusters further smaller. This is demonstrated in Algorithm 1. This creates the premature, initial clusters.

The clusters formed by the anomalous points as the cluster centers are now removed from the list of pure clusters. These pure clusters are further refined by removing border points from each cluster and re-assigning them to the nearest cluster. A border point j belonging to cluster  $C_i$  is one which meets the following criterion:

$$\delta_{C_i} < \Delta_{C_{jk}},$$

where  $\delta_{C_i}$  is the mean intra-cluster distance of cluster  $C_i$  and  $\Delta_{C_{jk}}$  is the mean distance of point j to all points k in cluster  $C_j$ .

All the pure clusters obtained no longer contain any points that exist in more than one cluster. However, there could be clusters present which do not meet the cutoff-size constraint. For this, cluster reassignment is done. All points in clusters which do not meet the cluster cut-off size are reassigned to the closest cluster to the point. Here, the average distance between the point and the members of the cluster are used to determine the closest cluster to the point. This process is outlined in Algorithm 2.

2) Distance Vicinity Threshold Calculation: The distance vicinity threshold parameter can be specified as an input to the algorithm. However, an accurate and effective method to calculate the distance vicinity threshold was discovered by

**Algorithm 2** Point Reassignment

**Input:** Pure Clusters  $C$ , Cluster cut-off size  $\lambda$ , Distance Matrix  $\Delta[N]$

**Output:** Final Clusters  $\hat{C}$  Removal of Anomalies

```

1:  $\hat{C} = C - C.anomalies$ 
2: for  $i = 1$  to  $len(\hat{C})$  do
3:   pairwiseDist = array of all distance between all pairs of points in  $c_i$ 
4:   for  $j = 1$  to  $len(c_i)$  do
5:     if  $(\mu(\Delta[i, j]) > \mu(pairwiseDist))$  then
6:       Append Point  $j$  to  $\hat{C}$  and remove point from  $c_i$ 
7:     end if
8:   end for
9: end for
10: sort  $\hat{C}$  in descending order
11: remove empty clusters from  $\hat{C}$ 
12: for  $i = 0$  to  $len(\hat{C})$  do
13:   if  $len(c_i) < \lambda$  then
14:     for  $j = 0$  to  $len(c_i)$  do
15:       Point  $j$  is appended to closest cluster that is above cluster cut-off size, removed from  $\hat{C}$ 
16:     end for
17:   sort  $\hat{C}$  in descending order
18: end if
19: end for
20: return  $\hat{C}$ 

```

formulating a cost function to be minimized. Our cost function uses the mean of the mean pairwise distance within clusters and the ratio of total number motifs to the number of clusters.

Let  $\delta$  signify the mean within-cluster distance. This distance can be defined for each pure cluster as the average distance between every pair of points in the cluster. Hence the within cluster mean distance,  $\delta^*$  is the mean distance for all within cluster distances for all clusters, for a given distance vicinity threshold. Further, let  $\rho$  signify the grouped ratio. This is the ratio between the number of pure clusters to the number of data points, given an distance vicinity threshold. The within cluster mean distance,  $\delta^*$  varies in a non-decreasing manner with increasing distance vicinity threshold value. The grouped ratio,  $\rho$ , however, varies in a non-increasing manner with increasing distance vicinity threshold. After normalizing these parameters, a cost function can be formulated as follows:

$$Cost_{\epsilon} = A\delta^* + B\rho_{\epsilon}$$

where A and B are heuristic weights.

The nature of the relationship of  $\delta^*$  and  $\rho$  with the distance vicinity threshold is dependent on the dissimilarities between the data points. However, the cost function is seen to vary with the distance vicinity threshold as follows:

- Increasing, when  $\epsilon \in [0, \alpha_1)$
- Decreasing, when  $\epsilon \in [\alpha_1, \alpha_2)$
- Increasing, when  $\epsilon \in [\alpha_2, \infty)$

By optimizing this function and finding the global minima ( $\alpha_2$ ), we can obtain the optimum distance vicinity threshold to be passed to the clustering algorithm.

#### IV.

An initial analysis of the data was done using the BDS test, which when performed on the particular intervals gave the following results:

TABLE II  
P-VALUES

	p-values
p(daily)	$3.67 * 10^{-29}$
p(weekly)	$7.01 * 10^{-5}$
p(monthly)	$9.86 * 10^{-7}$

The results of this test indicate non-linearity in the data hence methods better than ARIMA are required. Our approach involves the support of LSTM to aid the prediction of residuals.

#### A. Forecasting using SARIMA

The parameters used for the daily, weekly and monthly models are given in III and IV. These parameters were obtained using the autocorrelation function (ACF) and the partial autocorrelation function (PACF) plots, which give the correlation of a stationary time series with its own lagged values [26].

TABLE III  
NORMAL PARAMETERS

	p	d	q
Daily	4	1	3
Weekly	2	0	1
Monthly	2	0	0

TABLE IV  
SEASONAL PARAMETERS

	P	D	Q	S
Daily	2	0	0	7
Weekly	1	0	0	52
Monthly	1	0	0	12

The root mean square error (RMSE) values from 06-12-2009 to 02-01-2010 are depicted in V, along with the Akaike information criterion (AIC) estimations [27].

#### B. Forecasting using LSTM

Fig. 2 shows the modeling of the actual Global Active Power against the vanilla and CNN encoder-decoder LSTM forecasting for the year 2009, followed by the predictions for a week in Fig. 3.

TABLE V  
PREDICTED RESULTS

	AIC	Train RMSE Values	Test RMSE Values
Daily	16357.181	459.899	424.600
Weekly	1888.728	2377.935	2112.522
Monthly	472.195	16634.531	5233.018

The RMSE values are shown in VI for both the vanilla LSTM model as well as the CNN encoder-decoder LSTM model.

- 1) Hybrid Model: The hybrid model achieved a testing

TABLE VI  
RMSE VALUES

Model	RMSE Values
LSTM	446.994
CNN encoder-decoder LSTM	432.906

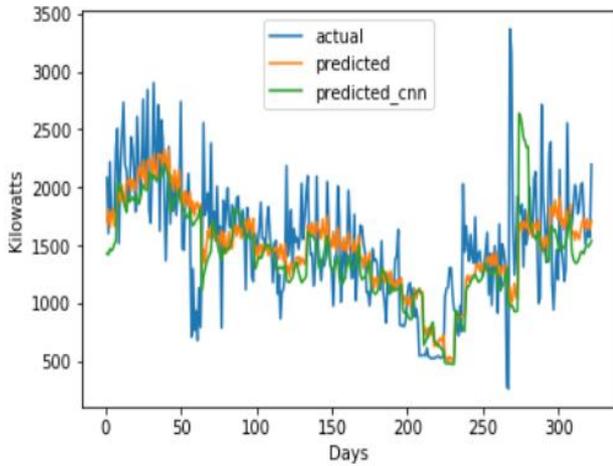


Fig. 2. Pure LSTM Test Results

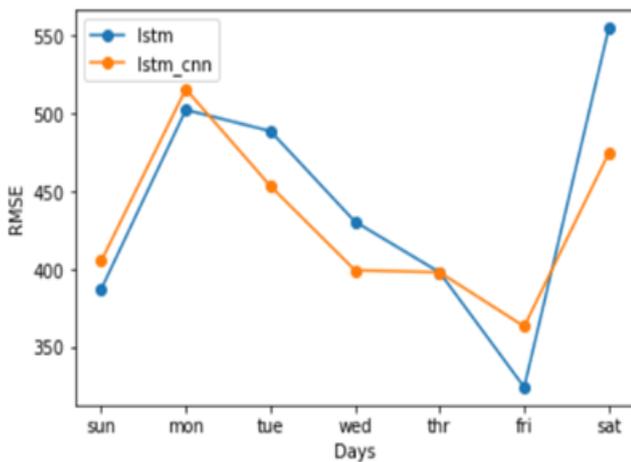


Fig. 3. Predictions for a week

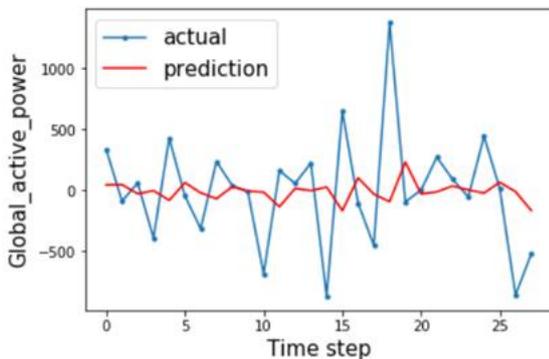


Fig. 4. Residual Predictions from Hybrid Model

RMSE of 421.23 which is marginally better than that of the SARIMA model's testing RMSE of 424.60, signifying an improvement in the model's performance due to residual modeling by LSTM. The residuals have been depicted in Fig. 4, along with the results in Fig. 5.

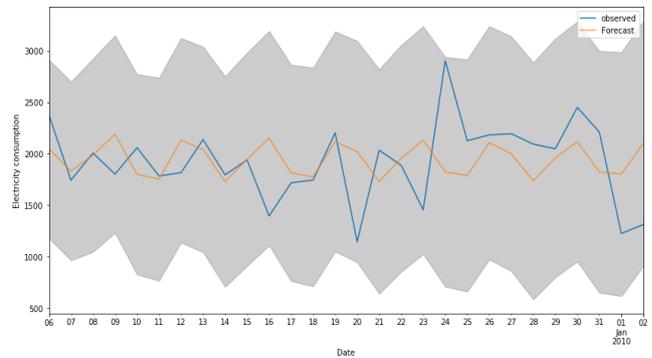
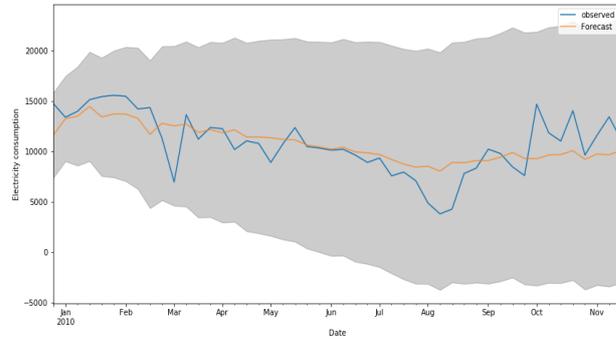
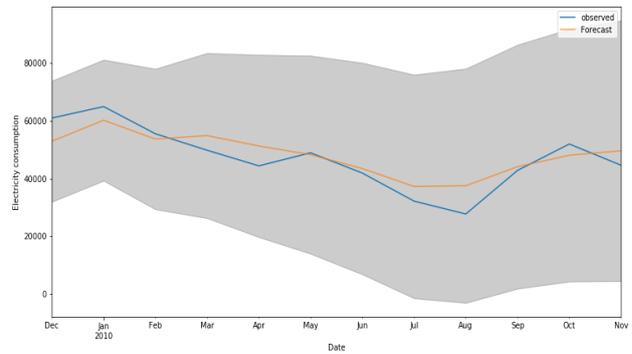


Fig. 5. Predictions from Monthly, Weekly and Daily data

## C. Clustering

To validate our clustering model, we ran experiments with slices of the time series data of a window size of two hours. These slices were first z-normalized to be able to capture similar patterns at different scales. This data was taken for an entire month, with a one hour overlap between slices.

For our experimentation, Dynamic Time Warping (DTW) distance was chosen as the dissimilarity measure between time slices, due to its phase-shift invariant nature. Hence, the distance matrix, which contains the DTW distance between each pair of slices was computed. This distance matrix was then fed into the clustering algorithm with various  $\epsilon$  values. Fig. 6 represents the largest cluster obtained with values 0.025, 0.05 and 0.1 respectively. It is observed from the same figure that the same cluster becomes less and less refined with increasing  $\epsilon$  value. There is an increase in the number of irregularly shaped slices which seem as though they do not belong in the cluster with the highest value of  $\epsilon$ . The clusters obtained with the lowest value of  $\epsilon$  are far more homogeneous. This can be justified as more slices fail to meet the cut-off distance criteria, and are removed from the clustering process for smaller  $\epsilon$  values. Hence,  $\epsilon = 0.025$  is used for clustering. The cluster shown in the Fig. 7 further displays the effectiveness of the

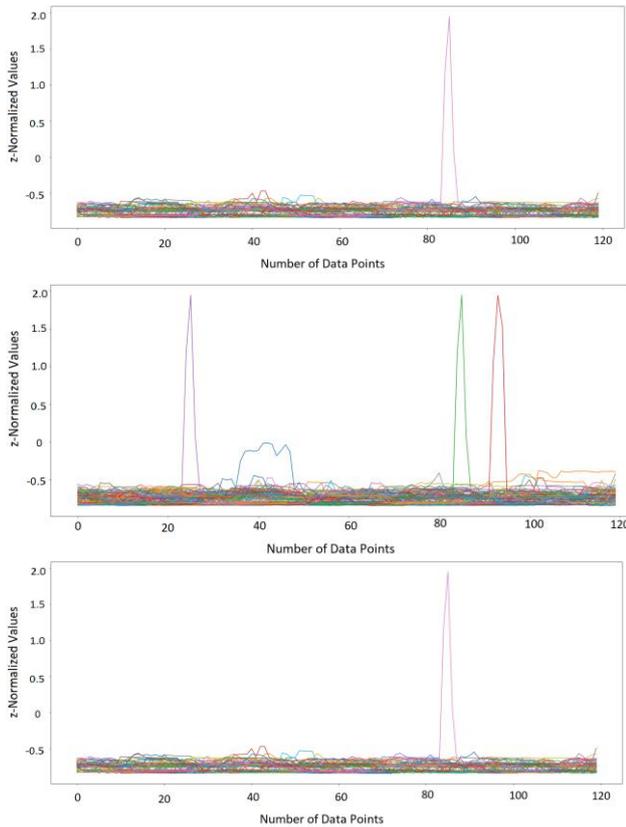


Fig. 6. Clusters obtained with  $\epsilon = 0.025$ ,  $\epsilon = 0.05$ , and  $\epsilon = 0.1$  respectively

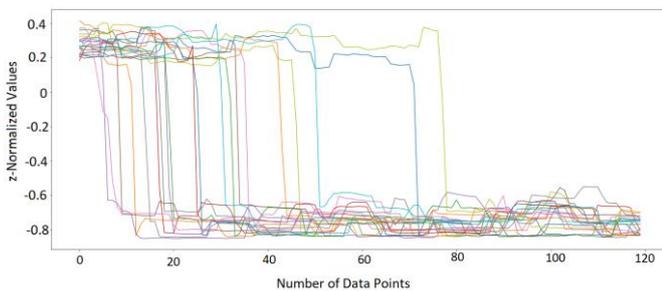


Fig. 7. Phase Differenced Slices in One Cluster

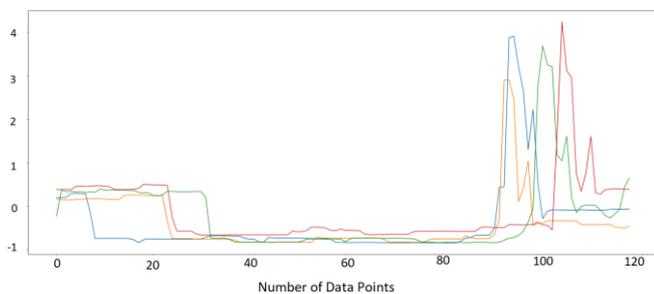


Fig. 8. Clusters occurring consistently at 4 in the morning

### ACKNOWLEDGMENT

The authors would like to thank the UCI Machine Learning Repository for their open source electricity dataset.

### REFERENCES

1. B. Young Choi, Jae Lee, "Combining lstm network ensemble via adaptive weighting for improved time series forecasting," 2018.
2. Z. Y. X. . C. H. . A. K. . F. K. Guo, Tian Xu, "Robust online time series prediction with recurrent neural networks," 2016.
3. S. Saab, E. Badr, and G. Nasr, "Univariate modeling and forecasting of energy consumption: the case of electricity in lebanon," 2001.

clustering algorithm with phase-differenced patterns. With the obtained clusters, it is now possible to leverage domain knowledge to make meaningful insights from the given data. As an example, the cluster represented in Fig 8. is one where the recurring patterns occurred at nearly the same time of day for the month, at 4 am. Hence, the regular spike in the electricity consumption can be taken into account by the utility provider to allocate extra blocks of power in this particular grid at 4am.

### V. CONCLUSIONS

We have analyzed time series data from electricity sensors of a household. The experimental results have demonstrated a hybrid model of RNN-LSTM and SARIMA works better than either of the models alone or other traditional regression-based forecasting approaches reported in literature. The predictions made by the hybrid have a lower RMSE value of 432.906 as compared to the traditional LSTM which had a RMSE of 446.994 and SARIMA with 459.899. We have also demonstrated that meaningful insights on electricity consumption patterns in a household can be gleaned from clustering the data on shorter time periods using a hybrid model that combines both density and centroid-based approaches for clustering. These insights can help allocate blocks of power that adapt to the usage patterns of various utility-providers.

4. P. Chujai, N. Kerdprasop, and K. Kerdprasop, "Time series analysis of household electric consumption with arima and arma models."
5. H. Bangui, M. Ge, and B. Buhnova, "Exploring big data clustering algorithms for internet of things applications," pp. 269–276, 01 2018.
6. H. T. Siegelmann and E. D. Sontag, "On the computational power of neural nets," 1992.
7. D. Yarotsky, "Optimal approximation of continuous functions by very deep relu networks," 2018.
8. A. Koencke and A. Gajewar, "Curriculum learning in deep neural networks for financial forecasting," 2019.
9. G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," 2009.
10. O. Maillard and R. Munos, "Compressed least-squares regression," in Advances in Neural Information Processing Systems 22 (Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds.), pp. 1213–1221, Curran Associates, Inc., 2009.
11. M. Vlachos, J. Lin, and E. Keogh, Proceedings of the Workshop on Clustering High Dimensionality Data and Its Applications, p. 2330. 2003.
12. T. Oates, M. Schmill, and P. Cohen, "A method for clustering the experiences of a mobile robot that accords with human judgments," 02 2001.
13. S. Hirano and S. Tsumoto, "Empirical comparison of clustering methods for long time-series databases," vol. 3430, pp. 268–286, 01 2003.
14. X. Wang, K. Smith-Miles, and R. Hyndman, "Characteristic-based clustering for time series data," Data Min. Knowl. Discov., vol. 13, pp. 335–364, 09 2006.
15. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," June 1999.
16. S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," Intelligent Data Analysis, vol. 11, p. 561580, Oct 2007.
17. D. Dua and C. Graff, "UCI machine learning repository," 2017.
18. S. Yakowitz, "Nearest neighbor method for time series analysis," Journal of Time Series Analysis, vol. 8, pp. 235 – 247, 06 2008.
19. S. Moritz, A. Sard, T. Bartz-Beielstein, M. Zaefferer, and J. Stork, "Comparison of different methods for univariate time series imputation in r," 2015.

20. M. Lepot, J.-B. Aubin, and F. Clemens, "Interpolation in time series: An introductory overview of existing methods, their performance criteria and uncertainty assessment," *Water*, vol. 9, p. 796, 10 2017.
21. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," 2015.
22. F. B. Hasim Sak, Andrew Senior, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014.
23. A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," 2018.
24. O. J. O. O. G. A. kintunde, M. O., "Detection of non-linearity in the time series using bds test," 2015.
25. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," 2003.
26. M. T. Hagan and S. M. Behr, "The time series approach to short term load forecasting," *IEEE Transactions on Power Systems*, vol. 2, pp. 785–791, Aug 1987.
27. H. Bozdogan, "Model selection and akaike's information criterion (aic): The general theory and its analytical extensions," *Psychometrika*, vol. 52, pp. 345–370, 02 1987.

### AUTHORS PROFILE



**Midhush Manohar T.K.**, is a Computer Science and Engineering graduate from PES University, Bangalore. He is an active member of the Centre for Cloud Computing and Big Data, PES University. He had spent his summer as a Research Assistant at the Indian Institute of Science, Bangalore in the Centre for Nano Science and Engineering, and is currently serving as a Teacher Assistant for the Big Data course. His research interests primarily lie in the areas of natural language processing, data analytics and deep learning based solutions to various computational problems.



**Naveen Suresh**, is a Computer Science and Engineering graduate from PES University, Bangalore. He is a member of the PES Center for Pattern Recognition, and his research interests lie in machine learning and deep learning methods for time series analysis and bioinformatics related computational problems.



**Srikumar Subramanian**, a Computer Science and Engineering graduate from PES University, Bangalore. He is currently a member of the Center for Pattern Recognition in PES University and the Center for Cloud Computing and Big Data in PES University. His research focuses on areas in machine learning and natural language processing.



**Gowri Srinivasa**, obtained her Bachelors in Computer Science and Engineering from PES Institute of Technology (2004) and her PhD in Biomedical Engineering from Carnegie Mellon University (2008). Since August 2008, she has been with PES University, where she is currently a Professor in the Department of Computer Science and Engineering and heads the Center for Pattern Recognition. Her research focuses on the design of signal processing, analytics and machine learning based solutions to problems primarily in the domains of education and healthcare. She is a Senior Member of the IEEE and has been on the Technical Committee of the IEEE Bio Imaging and Signal Processing Society (BISP) since 2009.