Investigating Forecasting Models, Exploratory Analysis, and Adversarial Vulnerabilities in Load Forecasting: UKDALE Dataset Study

Md Sahadul Hasan Arian
Electrical and Computer Engineering,
North South University
Dhaka, Bangladesh
hasan.arian@northsouth.edu

Mosroor Mofiz Arman
Electrical and Computer Engineering,
North South University
Dhaka, Bangladesh
mosroor.arman@northsouth.edu

Shaiara Tabassum
Electrical and Computer Engineering,
North South University
Dhaka, Bangladesh
shaiara.tabassum@northsouth.edu

***Abstract*—Accurate prediction of load consumption is crucial for both the demand side and supply side in order to foster economic growth. In this study, the forecasting task was conducted on the UKDALE actual real-life dataset, specifically focusing on house #4's aggregate channel. Various models including ARIMA, SARIMA, LSTM, and XGBOOST were employed, utilizing different sample rates. The trained models were then evaluated using six performance metrics: R2, MAE, MAPE, MSE, RMSE, and NRMSE. Remarkably, the current study achieved one of the best result, recording a value of 83.068 RMSE with LSTM, and a sample rate of 16 hours. These findings demonstrate the efficacy of LSTM for load consumption prediction in real-world scenarios.**

***Index Terms*—forecasting, ARIMA, XGBBoost, LSTM, adversarial**

I. Introduction

With the increasing global demand for electricity, both the demand and supply side are facing growing uncertainties [20]. Consequently, it becomes crucial to effectively manage electricity resources, implement demand response strategies, integrate renewable energy, and optimize energy purchasing and contract negotiations [21]. As the world evolves and energy consumption demand rises due to rising prosperity and expanding commercial activity, there has been a significant increase of over 15% in energy demand [22]. Consequently, it is imperative for businesses, decision makers, and consumers to rely on load forecasting techniques to support their operations and decision-making processes. Load forecasting offers numerous benefits such as improved resource allocation, enhanced demand response strategies, seamless integration of renewable energy sources, and efficient energy purchasing and contract negotiations [21]. In today's world, effective management of electricity resources is essential, considering the escalating energy consumption and environmental concerns. Load forecasting plays a pivotal role for electric utilities and energy management systems, particularly grid operators, as it enables them to anticipate and balance the electricity required to meet the demand. Accurate load forecasting is of utmost importance for power system planners and operators to ensure sufficient generation capacity to meet future demand. This research employs advanced data analytics techniques and timestamp data to predict residential electricity consumption patterns, aiming to develop reliable load forecasting models that inform decision-making for power utilities, grid operators, and policymakers. The anticipated outcomes of this study encompass optimized power generation, improved load management, and enhanced energy conservation.

The solution to the identified problem lies in the application of machine learning techniques to forecast future load, providing valuable insights into consumer behavior and business demands. Machine Learning (ML) has emerged as a powerful tool to address the challenges faced in this regard, offering enhanced accuracy and surpassing the limitations of traditional approaches. This research paper examines the effectiveness of ML in tackling the identified problems. Firstly, ML algorithms can effectively handle issues related to data availability and quality by employing imputation techniques and outlier detection. Secondly, ML models can capture the intricate dependencies of seasonality and weather conditions by incorporating historical weather data as input features. Thirdly, ML algorithms, such as artificial neural networks and random forests, are adept at modeling nonlinear and dynamic consumption patterns, thereby enabling accurate load forecasting. Fourthly, ML techniques excel in handling heterogeneity by utilizing clustering algorithms to segment residential consumers into homogeneous groups, enabling the development of personalized forecasting models. Fifthly, ML models automate data preprocessing tasks and feature selection, leading to significant time and effort savings. Sixthly, rigorous model selection and evaluation methods, including cross-validation and performance metrics, ensure the identification of the most appropriate ML models. Lastly, ML models can seamlessly integrate dynamic external influences, such as energy prices and environmental policies, by incorporating these factors as additional features. By leveraging these capabilities, machine learning proves to be a promising solution for forecasting future load and addressing the associated challenges.

In the industry, load forecasting and energy consumption prediction are addressed through the utilization of machine learning or deep learning models. Several well-known approaches, including ARIMA, LSTM, SARIMA, and XGBOOST, have been employed in separate research projects to effectively tackle various challenges in this domain. ARIMA models excel at capturing linear components and effectively handle seasonality, ultimately enhancing load scheduling within the power system. LSTM models, on the other hand, specialize in capturing long-term dependencies and provide valuable insights into energy usage patterns, enabling more efficient energy management strategies. SARIMA models demonstrate high accuracy in predicting short-term load demand, thereby facilitating improved planning and load flow management. XGBOOST models excel in handling complex relationships, leading to enhanced load forecasting accuracy. Lastly, Random Forest models contribute to improved energy consumption predictions and provide valuable insights into appliance-level usage patterns, offering significant benefits for household energy management. These existing solutions leverage the strengths of each respective model to address specific challenges and contribute to more accurate load forecasting and efficient energy consumption prediction. As the industry continues to evolve, further advancements in machine learning and deep learning models are anticipated, ensuring continuous improvement in load forecasting and energy management practices.

This research aims to address the problem of household electricity load forecasting using the UK Domestic Appliance-Level Electricity (UK-DALE) dataset. By leveraging this dataset, which contains detailed appliance-level electricity consumption data along with timestamp information from five houses, we can conduct a comprehensive analysis of residential energy usage patterns. In this study, we employ various forecasting models, namely ARIMA, LSTM, SARIMA, and XGBOOST, to predict electricity load and evaluate their performance. Notably, this research stands out by incorporating all these models into a single study and conducting a comparative analysis. By comparing and analyzing the results, our objective is to identify the most accurate and practical approach for load forecasting in the context of residential electricity consumption. To enhance the accuracy of our models, we utilize resampling and curve fitting techniques with different sampling rates such as 1H, 6H, 12H, and 24H. Additionally, we evaluate the models' performance using essential indicators such as AIC and BIC. The performance metrics are presented in a table format, with the models listed on the left side and their corresponding results on the right side. Furthermore, we investigate the robustness of these load forecasting models against adversarial and privacy attacks on the dataset. The outcomes of this research have the potential to improve energy planning, facilitate efficient resource allocation, and contribute to the development of sustainable and reliable energy systems.

The unique contributions are:

- The research aims to experiment with various forecasting models, including ARIMA, LSTM, SARIMA, and XGBOOST, to evaluate and compare their performance in predicting electricity load for residential consumption.
- Resampling and curve fitting techniques are utilized to train the forecasting models using different sampling rates (e.g., 1H, 6H, 12H, 24H), allowing for robust model performance evaluation under different temporal resolutions.
- Essential performance indicators such as AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) are contrasted to assess the forecasting models' performance. The results are presented in a table format, with the models listed on the left side and their corresponding performance metric results on the right side, providing insights into accuracy and reliability.

II. Literature Review

This section provides a comprehensive review of the existing literature on machine learning and deep learning approaches for time series forecasting, including an examination of related works on the UKDALE dataset and adversarial attack on a time series forecasting model.

A. *Machine-Learning based forecasting*

Machine learning models have been extensively studied for load forecasting in various contexts. There are current body of literatures [1-9] that have explored machine learning-based approaches for load forecasting of time series and compared the forecasting abilities of different models and proposing hybrid approaches to improve accuracy. One such study [1] compared ARIMA and SVM models for short-term load forecasting, highlighting the importance of incorporating external factors and outlier detection methods for improved accuracy. The use of SVM regression was also discussed as a solution for challenges such as small sample sizes, high dimensions, and local minima. Another paper [2] introduced a hybrid model that combined clustering and the ARIMA model to forecast peak loads of university buildings. This approach utilized AUTOARIMA for automated model selection based solely on time series data. In terms of addressing noise in time series data, a study [3] evaluated the tolerance of ARIMA models to noise and provided practical guidelines for load forecasting with noisy data. A hybrid ARIMA-SVM model was proposed in another paper [4], which integrated outlier detection, correction techniques, and external factors to enhance accuracy, speed, and convergence in load demand prediction. Furthermore, the feasibility of using XGBoost for load forecasting at the individual industrial customer level was explored in a different study [5]. This research introduced an adaptive decomposition technique to mitigate the negative effects of non-stationarity, non-linearity, and volatility in electricity load data. Additionally, various experiments conducted in China and Ireland demonstrated significantly higher prediction accuracy compared to state-of-the-art models. Another paper [6] proposed a scheme that leveraged weekly time series data, employed XGBoost for feature extraction and selection, and achieved superior accuracy metrics in load forecasting. Finally, a method combining K-nearest neighbors selection, Deep Belief Networks, NSGA-II algorithm, and adaptive kernel density estimation was proposed [7] to forecast electrical load with reduced computing costs. Another paper [8] presented a novel approach for weekly load prediction in the Turkish Electric Market, addressing the limitations of previous short-term load forecasting systems. Lastly, a study [9] evaluated and compared four distinct models (Seasonal ARIMA, Seasonal ARIMAX, RF Regressor, and GB Regression Trees) for day-ahead load forecasting in the New England Electricity Market.

B. *Deep-Learning Based Forecasting*

There are current body of literatures [10-12] that have explored deep learning approaches for load forecasting of time series. Deep learning models have gained significant attention in the field of load forecasting. One approach [10] incorporates long short-term memory (LSTM) and fully-connected (FC) layers to tackle the challenges of day-ahead load forecasting, integrating time-series characteristics, non-linear correlations, and additional data like weather information and day of the week. Another study [11] introduces a parallel CNN-LSTM attention (PCLA) model, which utilizes multivariate time-series data to forecast load. The model leverages CNN and LSTM for feature extraction, employs an attention mechanism to highlight important features, and captures spatial information through CNN while learning temporal features through LSTM layers. In a different paper [12], the authors contribute to short-term residential load forecasting by employing exploratory data analysis, a density-based clustering technique, and LSTM. They describe a residential load forecasting framework based on LSTM, develop an empirical optimization-based predictor, demonstrate the effectiveness of LSTM in capturing subtle temporal consumption patterns, and emphasize the accuracy improvement achieved by aggregating individual forecasts rather than directly forecasting aggregated loads. These studies showcase the versatility and superior performance of deep learning models in load forecasting tasks.

C. *UKDALE Dataset Related Studies*

There are literatures [13-15] that have addressed the challenges of working with the UKDALE dataset, primarily focusing on energy disaggregation, load disaggregation, and load prediction. One study [13] proposed the development of deep learning and probabilistic models for sequential data generation and prediction in the UKDALE dataset, with a specific emphasis on energy disaggregation. Another research effort [14] introduced a method that effectively tackled load disaggregation challenges in the UKDALE dataset. This method achieved high accuracy in classification, power estimation, and disaggregation of appliances, even in the presence of non-target appliances. Importantly, it was compatible with low-frequency sampled data from smart meters and demonstrated generalization capabilities on unseen data. Furthermore, a study [15] presented a load prediction method called federated deep learning (FedDL). FedDL combined federated learning and BiLSTM-Attention, ensuring user privacy and data security while achieving accurate forecasts by learning from the electrical power consumption of other households. Additionally, this work introduced a method based on NILM (Non-Intrusive Load Monitoring) that extracted load patterns from historical total load demand to enhance prediction results compared to traditional methods. Experimental validation was conducted using a CNN-LSTM hybrid deep learning model in various federated learning environments.

D. *Adversarial Attack Related Studies*

Literatures [16-19] have worked with adversarial attacks in with time series forecasting. In [16], the focus is on security issues in load forecasting procedures, highlighting vulnerabilities in current methods and introducing two data-driven attack algorithms. These algorithms generate adversarial input data that is difficult to detect, potentially leading to increased system operating costs or load shedding in power system operations. In [17], the research gap in adversarial attacks in deep learning-based load forecasting models is addressed. The paper proposes an availability adversarial attack targeting piecewise linear neural networks (PLNN) and an adversarial training algorithm to enhance the PLNN's robustness. Performance comparisons with integrity attacks and simulations demonstrate their effectiveness in load forecasting tasks. Moving to [18], the impact of adversarial attacks on fault classification problems is investigated. Different attack scenarios are examined, and empirical experiments on a dataset from the IEEE-13 test node feeder demonstrate the significant degradation of classification quality. Finally, [19] introduces an efficient adversarial attack crafting method based on forward derivative. This method considers factors such as input element magnitude, impact on regression output, and the number of controlled measurement meters.

III. Methodology

A. *Dataset Description*

The dataset used in this study is sourced from UKDALE [23], an open access dataset from the UK that records Domestic Appliance Level Electricity. It consists of electricity load consumption values for 5 houses, with each house having multiple channels representing different appliances. The dataset size is 16.9GB, but it can be compressed down to 1.62GB. The labels for the channels are stored in the labels.csv file. Figure 1 illustrates the channel count per house, where the aggregate channel (channel 1) represents the total load of the entire house. Initially, each channel's data included a UNIX timestamp and the power demand in watts (figure 3). The data spans from November 2012 to January 2015, and the channels have a minimum sampling rate of 6 seconds according to [23].

For this study, house 4 from the UKDALE dataset was selected due to several reasons. Firstly, it has a moderate size of only 183MB/16.9GB. Secondly, compared to other small data houses like house 3, house 4 utilizes appliances more

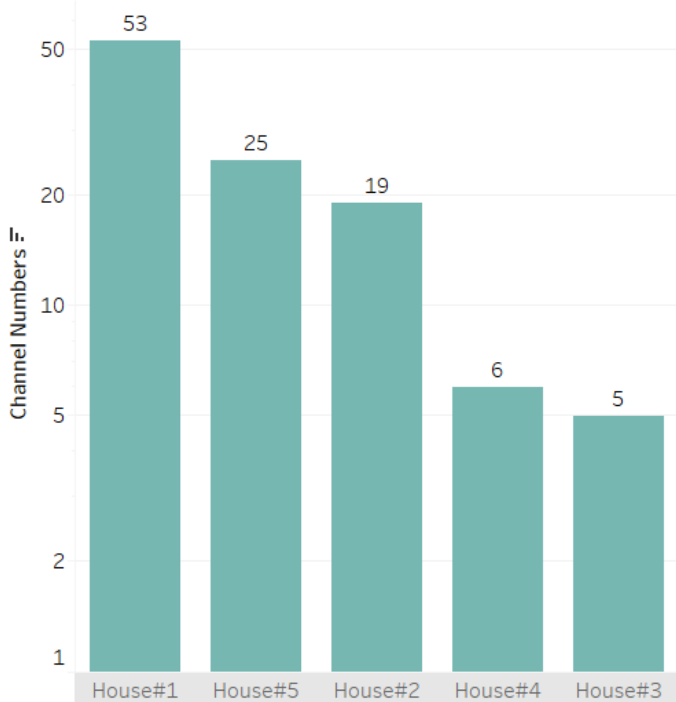

Fig. 1: Number of channels in each house.

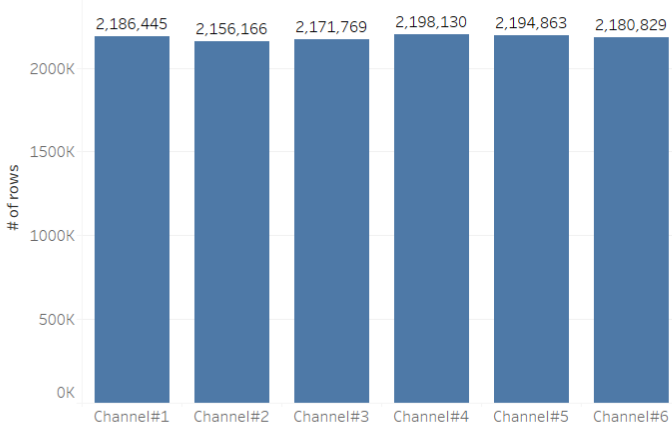

Fig. 2: Number of rows in each channel of house 4.

Timestamp	Power Demand (in Watt.)
1362818413	625
1362818419	625
...	
...	
1380582907	270

Fig. 3: Sample data in channel 1 of house 4.

Datetime	Power Demand (in Watt.)
2013-03-09 14:40:13	625
2013-03-09 14:40:19	625
...	
...	
2013-10-01 05:15:07	270

Fig. 4: Transforming UNIX timestamp to human-readable datetime.

frequently. Thirdly, the size and computational requirements of working with house 4's data align with the authors' logistical capabilities. Fourthly, house 4 provides a substantial record of 205 days, and the samples have moderate uptime when resampling, as discussed in the data preprocessing phase. Lastly, this house has a low number of channels, making it easier to handle within the confines of our logistical framework.

B. Dataset Preprocessing

The dataset consists of channels with two columns each: UNIX Timestamp and power demand (in watt), shown in Figure 3. An important observation is that the timestamp is not human-readable, causing issues in our exploratory data analysis (EDA) process. To address this, the timestamp has been converted into human-readable datetime format (highlighted in Figure 4). From this context, the first measurement date in channel#1, House#4 is March 9th, 2013, and the last measurement date is October 10th, 2013, resulting in 205 days of data in this channel. Additionally, it is crucial to note that the timestamp has a minimum frequency of 6 seconds. Therefore, if we were to resample the dataset, we need to consider the base sampling rate of 6 seconds.

When examining the data points of each channel in house#4 (as shown in figure 5), it becomes apparent that there are missing values from July 2013 to August 203. These missing values occur towards the end of the timeframe, leading the authors to decide to exclude data from July 2013 onwards in order to address the missing data. The outcome of this approach is illustrated in figure 6.

Figure 6 reveals that removing the 40% datapoint from July 2013 onwards resolved the issue of missing data. Additionally, the data contains a high frequency sample rate, necessitating resampling. The author intended to downsample the datapoints to intervals of 1, 6, 12, 16 (not displayed), and 24 hours from the original 6 seconds sampling rate. The outcomes of this resampling process are depicted in figure 7,8,9,10.

Resampling preprocessing involves a tradeoff, as increasing the sample rate in each resampling results in an exponential decrease in the number of data points. This can be seen in Figure 11. Therefore, having a larger number of data points in the raw data leads to better results after resampling. Proper resampling can enable accurate forecasting, particularly with larger house data such as house#1.

The initial dataset consisted of 1.3 million data points, which significantly decreased to around 2,000 after adjusting the sample rate from 6 seconds to 1 hour. This change resulted in a substantial reduction in the number of data points, marking the significant leap in the process.

C. Exploratory Data Analysis

In this study, we will exclusively utilize channel#1 of house#4 as it serves as the aggregate channel containing implicit information about all other channels, as depicted in figure 12. By plotting all channels together, we observe that channel#1, or the aggregate channel, encompasses information from all other data points (figure 12). Therefore, forecasting channel#1 enables us to consistently forecast the other channels as well.

Upon analyzing house#4 (referred to as house#4 or channel#1 interchangeably), on average on weekly consumption basis, their highest energy usage occurs on Tuesdays, while the lowest consumption is observed on Saturdays, Sundays, and Wednesdays (See figure 13).

In addition to that, on overall basis, house#4 shows a consistent decrease in electricity usage though out the recorded timeline, as depicted in figure 14. The residents have actively reduced their consumption, particularly after the summer season. Notably, a seasonal pattern emerges, indicating a repetitive trend influenced by behavioral habits. However, there are residual variations in the time series, potentially resulting from changes in human behavior or statistical errors.

The significance of lagged values in understanding resident behavior can be determined by analyzing the ACF and PACF curves in figure 15. The ACF reveals that the current value in the time series has a strong correlation with the four most recent past values, surpassing the significance level. Thus, with a sampling rate of 24 hours, we can infer that the power consumption from the past four days holds greater importance compared to other periods. Additionally, the ACF shows a weak significance level for the eighth lag, indicating a trend from the start of the week (approximately seven days ago) that influences the current value. The PACF confirms the significance of the first, second, third, and eighth lags in determining the present day's power consumption, aligning with the insights obtained from the ACF.

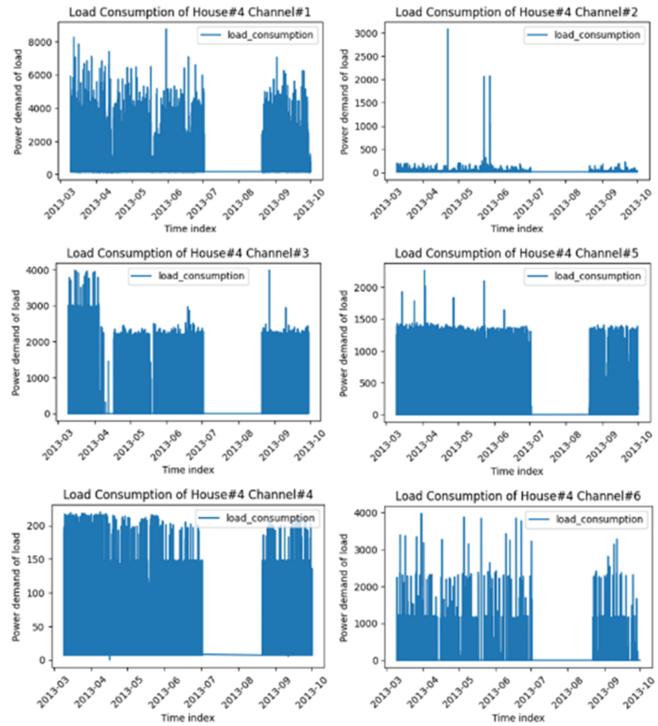

Fig. 5: Visualizing all channels of house 4.

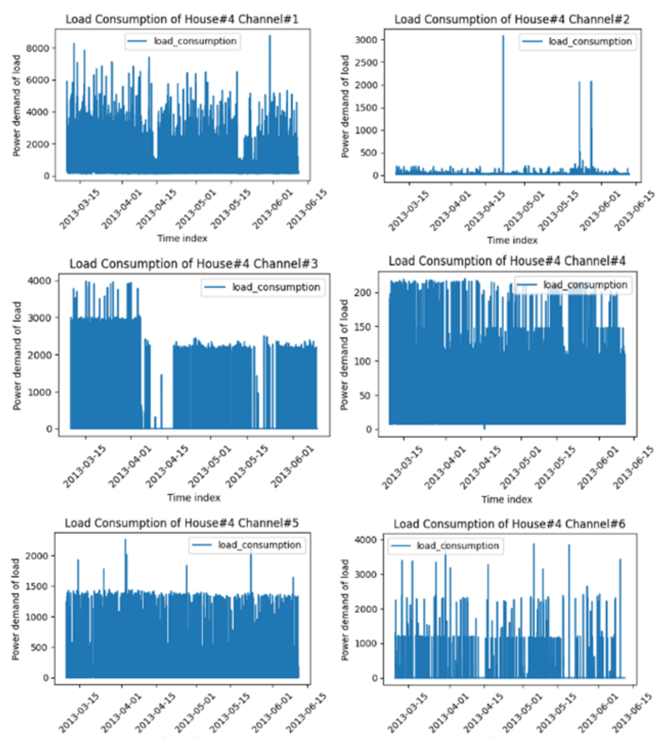

Fig. 6: Aftermath of handling missing data.

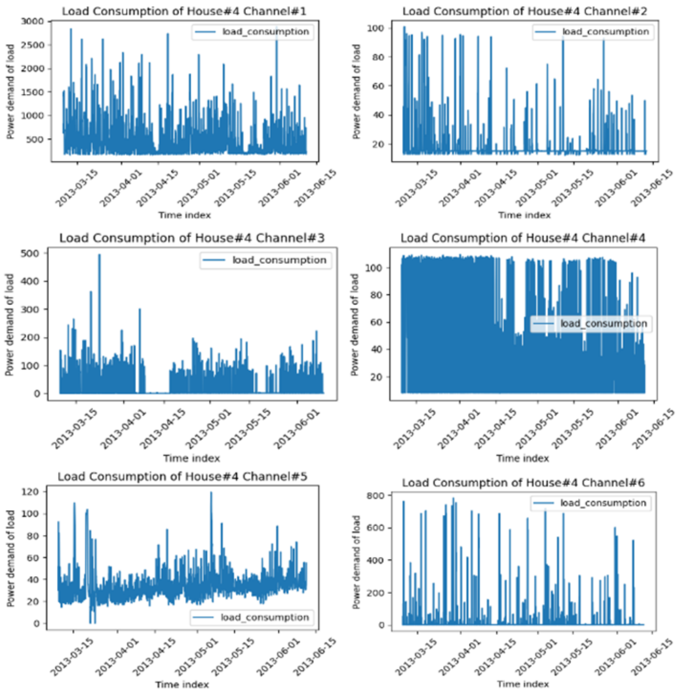

Fig. 7: Resampling: Sample rate 1H.

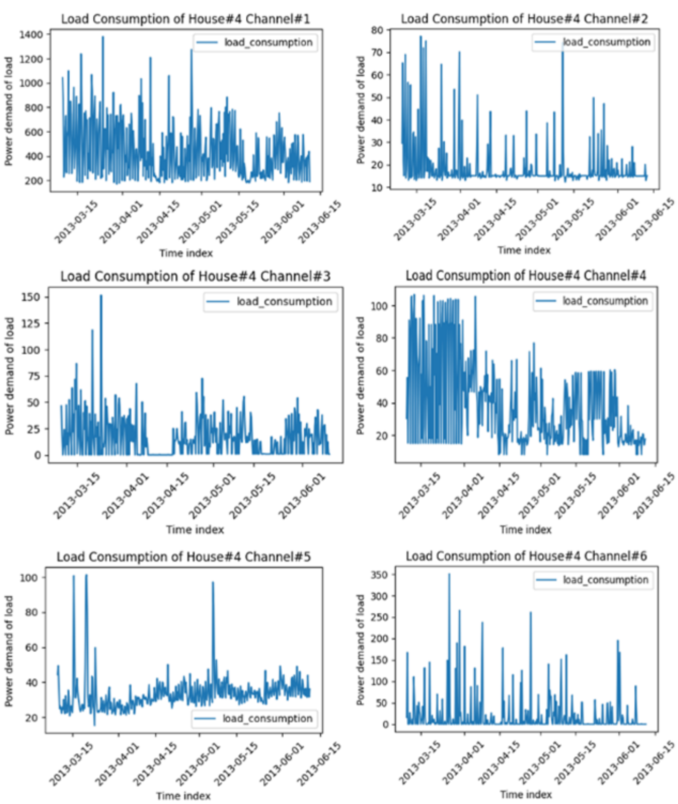

Fig. 8: Resampling: Sample rate 6H.

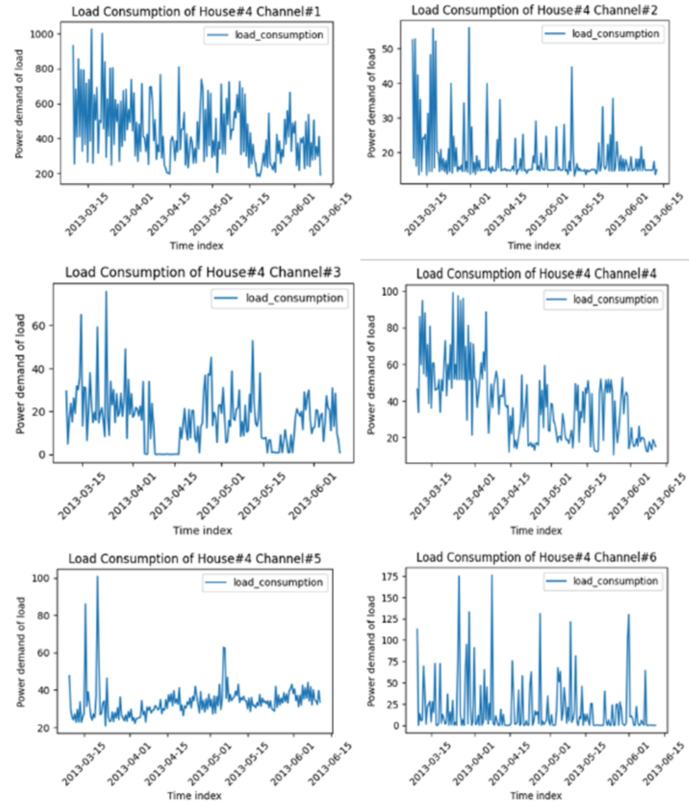

Fig. 9: Resampling: Sample rate 12H.

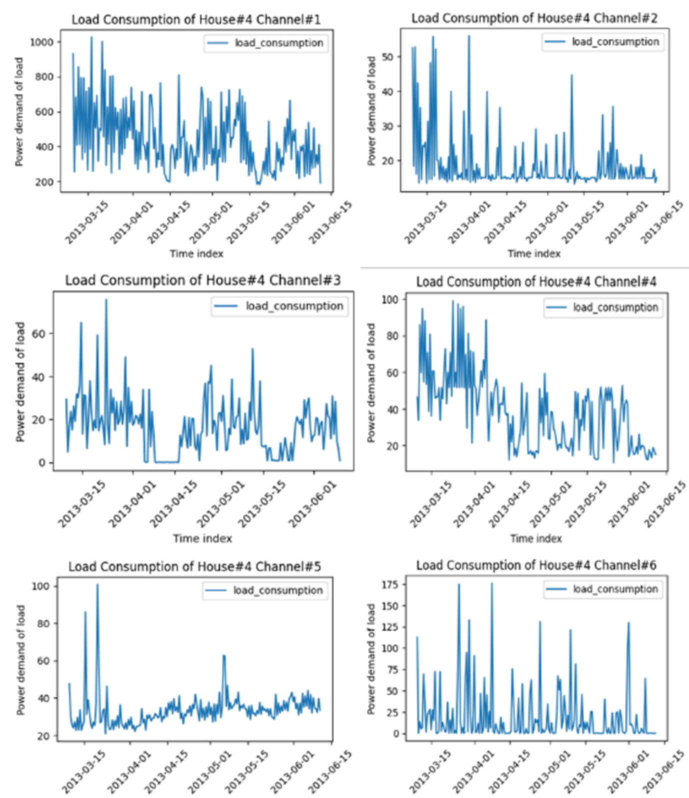

Fig. 10: Resampling: Sample rate 24H.

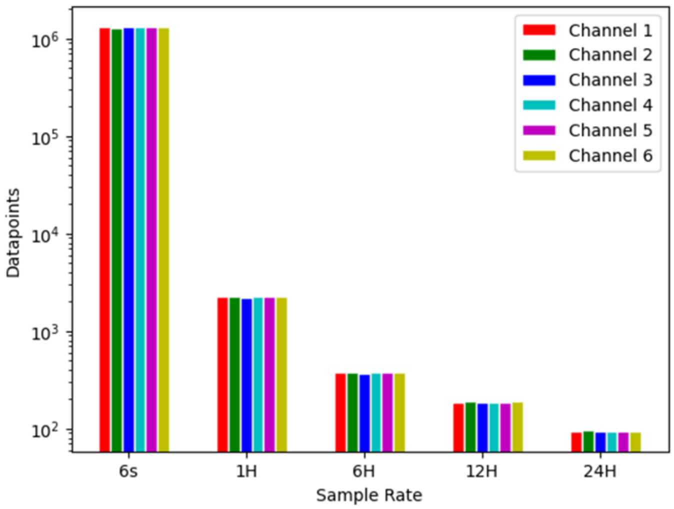

Fig. 11: The tradeoff: number of datpoints keep reducing exponentially upon downsampling.

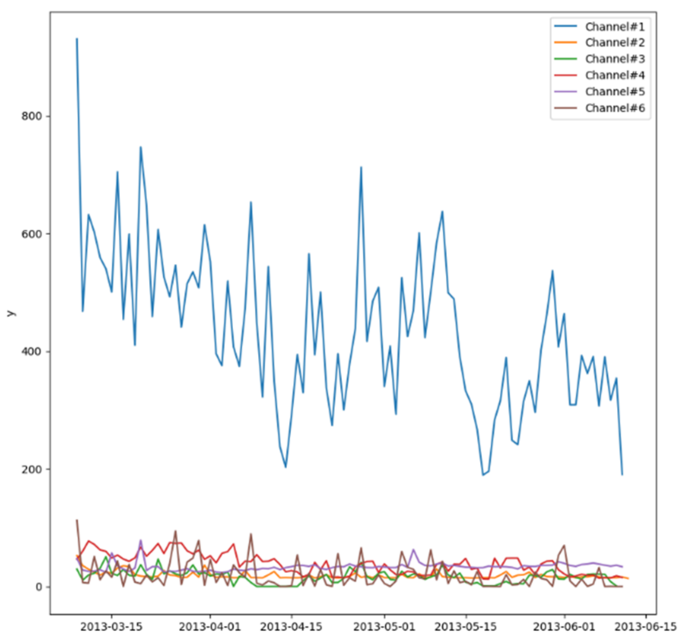

Fig. 12: Aggregate channel is the total aggregation of the house load consumption, so, channel#4 (blue) will be used for all further tasks in this study.

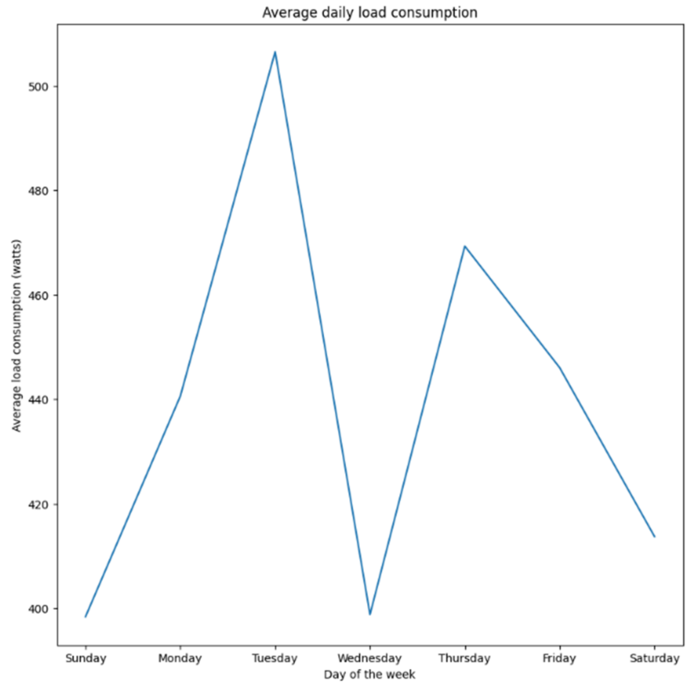

Fig. 13: Weekly based average load consumption behavior of the housemates in house#4.

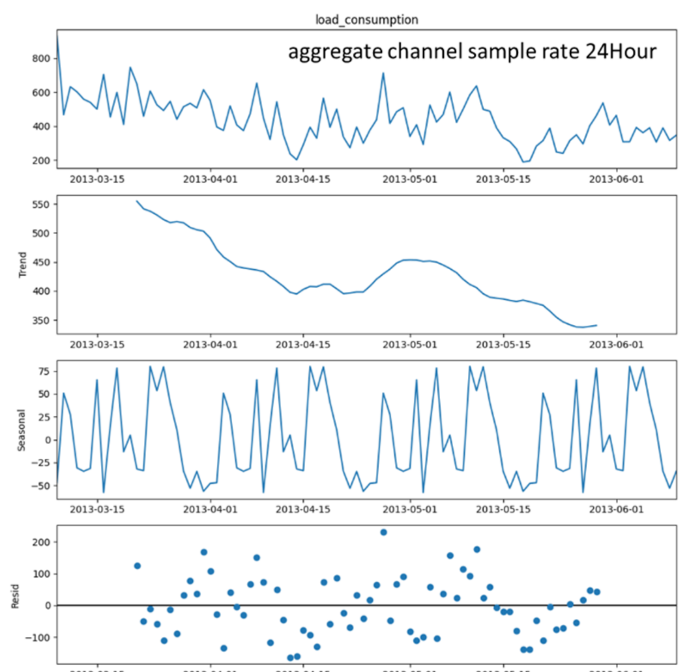

Fig. 14: the seasonal decomposition of the house#4 electricity decomposition.

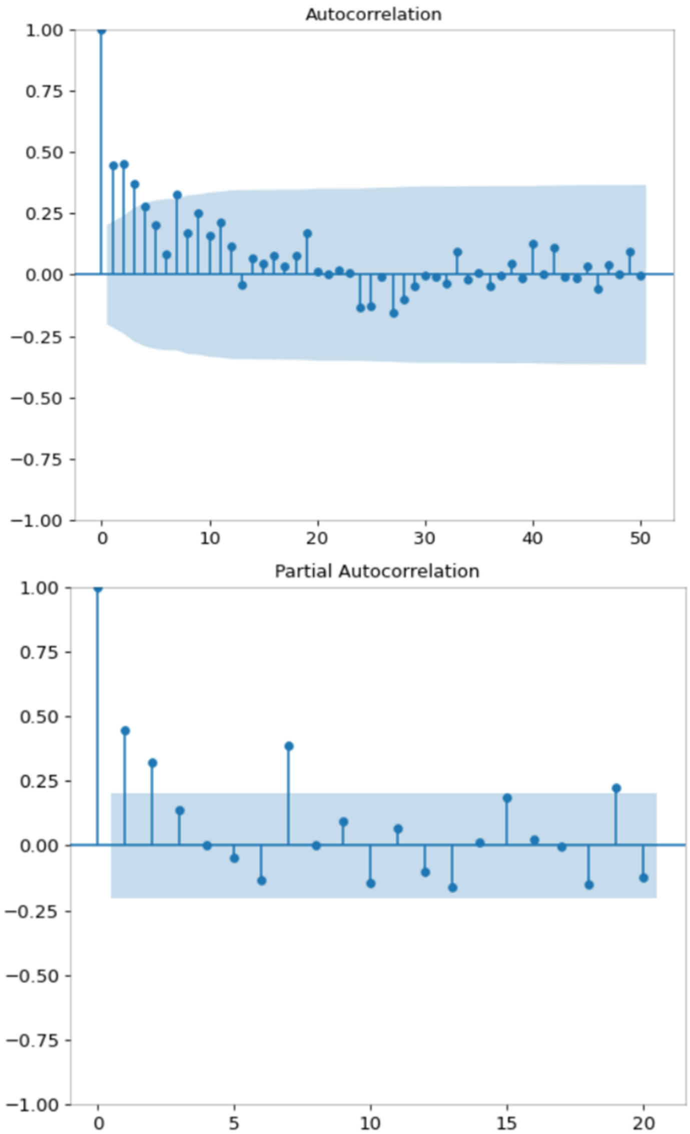

Fig. 15: The autocorrelation function and partial autocorrelation function plots.

D. *Forecasting Workflow*

The current study's forecasting tasks's workflow diagram is denoted in figure 16. In the current forecasting study, UKDALE dataset's [23] house#4 was selected. Data preprocessing, including handling missing data, deleting rows, and resampling, was conducted. Exploratory data analysis with relevant observations was presented (refer to methodology subsection C). channel#1 was chosen for load forecasting, with sampling rates of 1, 6, 12, 16, and 24 hours. For each sampling rate, a train-test split of 80% and 20% respectively was performed on the continuous time series data without randomization. The split data was utilized to train both machine learning (ARIMA, SARIMA, XGBOOST) and deep learning (LSTM) models. It should be noted that four models were trained for each sampling rate, resulting in a total of 20 models (see results). The findings of these models are analyzed in the

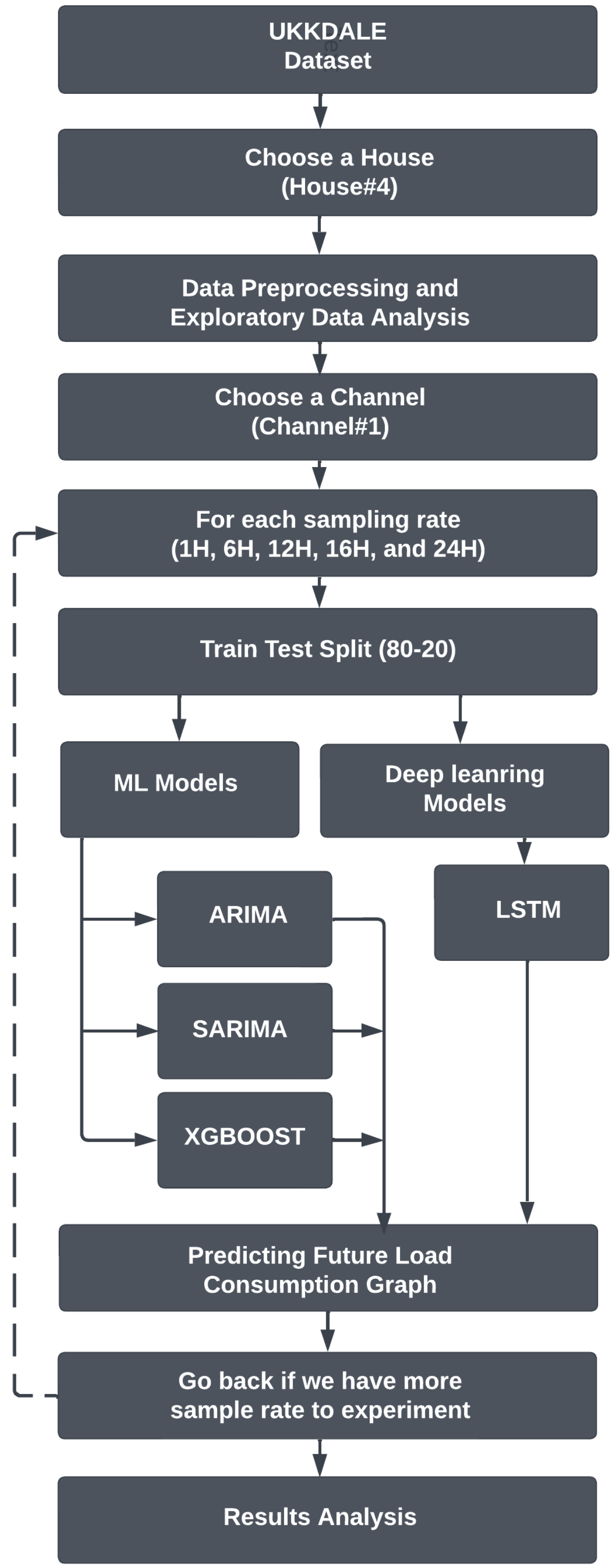

Fig. 16: The workflow diagram of current study.

Results section, while the methodology subsections F, G, H, and I provide descriptions and architectures of the models.

E. LSTM: Long Short-Term Memory

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) designed to capture long-term dependencies in time series forecasting. It employs memory cells with self-connected recurrent units, allowing the network to retain or forget information over extended intervals. The LSTM architecture incorporates input, forget, and output gates to regulate information flow and memory updates, enabling effective processing and prediction of sequential patterns. LSTM excels in extrapolating and learning complex non-linear relationships within time series [24]. In this study, we employed LSTM to train individual time series of channel#1 of house#4 at different sample rates. The LSTM model utilized hyperparameters such as sequence length = 10, input size = 1, hidden size = 32, output size = 1, batch size = 64, and learning rate = 0.001. The model architecture comprised LSTM followed by a fully connected layer projecting the hidden size into a single output for timestamp prediction. The training employed the MSE loss function and Adam optimizer. The MSE Loss function is given below:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

F. ARIMA

The Autoregressive Integrated Moving Average (ARIMA) model is a statistical approach widely utilized in time series analysis and forecasting. ARIMA integrates three key components: autoregressive (AR), differencing (I), and moving average (MA). The AR component accounts for the linear relationship [23] between the current value and previous values, while the MA component captures the dependence on past forecast errors. Additionally, the I component addresses non-stationarity through differencing. By appropriately adjusting the orders of these components, ARIMA can proficiently model and predict time series data. The general form of the ARIMA model is presented below:

$$\text{ARIMA}(p,d,q) : \mu + \epsilon_t + \sum_{i=1}^{p}\phi_i X_{i-1} + \sum_{i=1}^{q}\phi_j \epsilon_{t-1} \tag{2}$$

Here, X_t is actual observation, μ is vertical translation, $X_(i-1)$ is past time series, ϵ_{t-1} error in prediction on previous data points, ϕ_i is AR coefficient, and ϕ_j MA coefficient.

G. SARIMA

Seasonal Autoregressive Integrated Moving Average (SARIMA) is an extension of the ARIMA model that incorporates seasonality in time series data. It includes additional seasonal components to capture patterns that repeat at regular intervals. SARIMA combines the autoregressive (AR), differencing (I), and moving average (MA) components of ARIMA with the seasonal counterparts (SAR, SI, SMA). By considering both the non-seasonal and seasonal factors, SARIMA can effectively model and forecast time series data with seasonal patterns. It is a powerful tool for analyzing and predicting data that exhibits recurring seasonal behavior, such as quarterly sales or monthly weather patterns.

H. XGBOOST

XGBoost (Extreme Gradient Boosting) is a machine learning algorithm that can be applied to time series forecasting tasks. While XGBoost is primarily designed for cross-sectional data, it can also be used for time series by transforming the data into a supervised learning problem. In this approach, lagged versions of the target variable and other relevant features are used as inputs to predict the future value. XGBoost builds an ensemble of decision trees and optimizes their weights in a boosting fashion to minimize the prediction error. It can handle complex relationships and nonlinear patterns in time series data, making it effective for forecasting tasks.

IV. RESULTS

Six (6) performance metrics for four (4) model is illustrated in table 1, as well as two model selection metrics (AIC, and BIC) for ARIMA and SARIMA based models. The six performance metrics are: R2, MAE, MAPE, MSE, RMSE, and NRMSE. Their equations are given below:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{5}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

$$\text{RMSE} = \sqrt{MSE} \tag{7}$$

$$\text{NRMSE} = \frac{RMSE}{\max(y) - \min(y)} \tag{8}$$

Performance metrics in equation 4 to 8, the lower the value, the better the model is in performance. However, in equation 3, for R^2 score, the higher the value of R^2, the better.

Four models: ARIMA, SARIMA, LSTM, and XGBOOST are trained with sample rates 1H, 6H, 12H, 16H, and 24H indipendedtly which lead to total twenty (20) different trained model, each of the model's performance is measured with the equation 3 to equation 8: R^2, MAE, MAPE, MSE, RMSE, and NRMSE. The results are depicted in table 1 properly and the best results in each sample rate bin (i.e. best model result in some sample rate S_i, is denoted with asterisk*. More detailed visual is provided in Figure 17, 18, 19, 20, 21, and 22.

TABLE I: Forecasting channel#1 of house#4 with different model and their performance in different sample rate configurations. The best results in each criteria is marked with asterisk*

	Different	Model Selection Parameters		Performance Measures					
	Sample Rate	AIC	BIC	R2	MAE	MAPE	MSE	RMSE	NRMSE
ARIMA	$1H$	25532.645	25571.083	-0.09	155.4725	36.548*	76367.97	276.35	0.1
	$6H$	4070.488	4092.691	-0.42	139.0737	50.79	26035.09	161.35	0.28
	$12H$	1869.458	1884.444	0.33*	69.03728*	18.668*	8286.688*	91.038*	0.28*
	$16H$	1462.58	1470.709	-2.08	132.0115	33.28	23976.87	154.84	0.45
	$24H$	921.192	925.8	-0.32	63.4148	16.338*	7089.93	84.2	0.28
SARIMA	$1H$	25529.996	25546.47	-0.12	217.127	77.36	78414.31	280.03	0.1
	$6H$	3984.962	4010.865	0.11*	106.77228*	35.668	16403.328*	128.088*	0.228*
	$12H$	911.103	1923.092	0.27	75.2155	23.76	9123.75	95.52	0.21
	$16H$	1448.007	1466.974	-3.03	156.1369	40.15	31377.82	177.14	0.52
	$24H$	921.192	925.8	-0.32	63.4148	16.33	7089.93	84.2	0.28
LSTM	$1H$	–	–	0.08*	155.25828*	44.798	63788.748*	252.568*	0.098*
	$6H$	–	–	-0.6	145.1418	53.54	29625.49	172.12	0.3
	$12H$	–	–	0.02	84.6011	23.63	12629.67	112.38	0.24
	$16H$	–	–	0.17*	4.43818*	19.868*	7138.038*	84.498*	0.238*
	$24H$	–	–	-0.11*	58.38768*	19.58	6898.368*	83.068*	0.248*
XGBOOST	$1H$	–	–	-0.17	175.3887	51.14	81286.38	285.11	0.1
	$6H$	–	–	-0.15	110.2044	31.948*	21368.52	146.18	0.26
	$12H$	–	–	-0.29	99.8509	29.34	16707.35	129.26	0.27
	$16H$	–	–	-0.84	106.3894	35.26	15858.89	125.93	0.34
	$24H$	–	–	-0.85	83.7768	27.05	11494.32	107.21	0.31

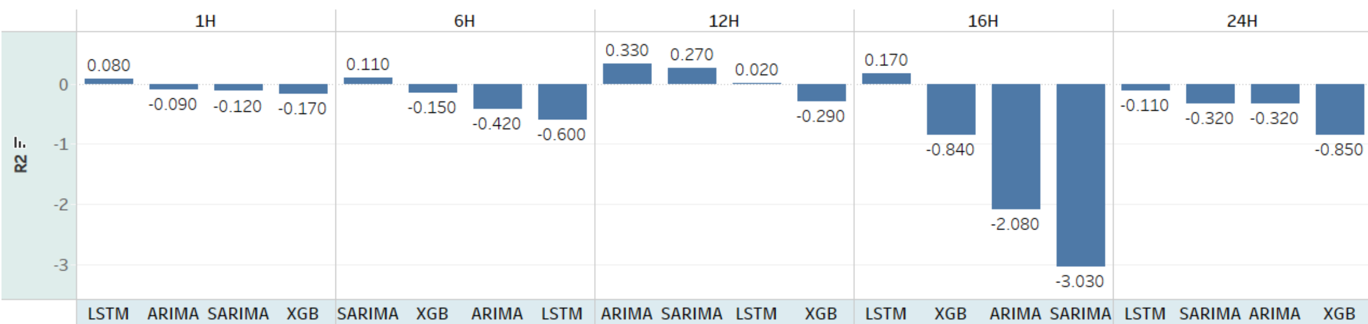

Fig. 17: The R2 scores of the trained models for each sample rate are arranged in **ascending** order. (Higher R2 scores (left to right in the graph) indicate better performance, i.e., LSTM, SARIMA, ARIMA, LSTM, LSTM exhibited good performance on 1H, 6H, 12H, 16H, and 24H sample rates, respectively.)

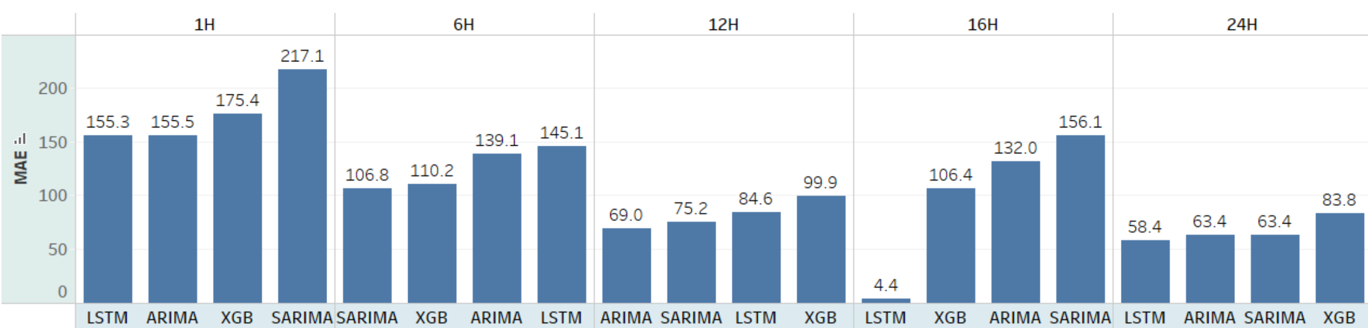

Fig. 18: The MAE scores of the trained models for each sample rate are arranged in **descending** order. (Lower MAE (left to right in the graph) indicate better performance, i.e., LSTM, SARIMA, ARIMA, LSTM, LSTM exhibited good performance on 1H, 6H, 12H, 16H, and 24H sample rates, respectively.)

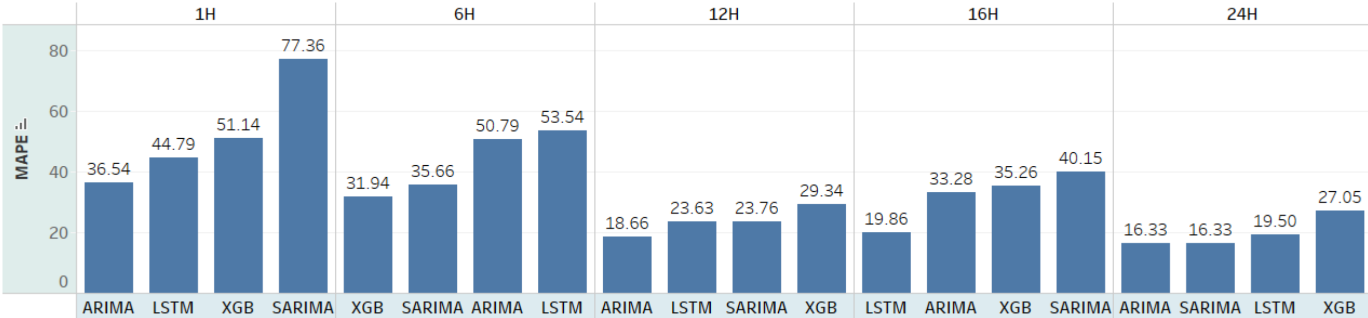

Fig. 19: The MAPE scores of the trained models for each sample rate are arranged in **descending** order. (Lower is better)

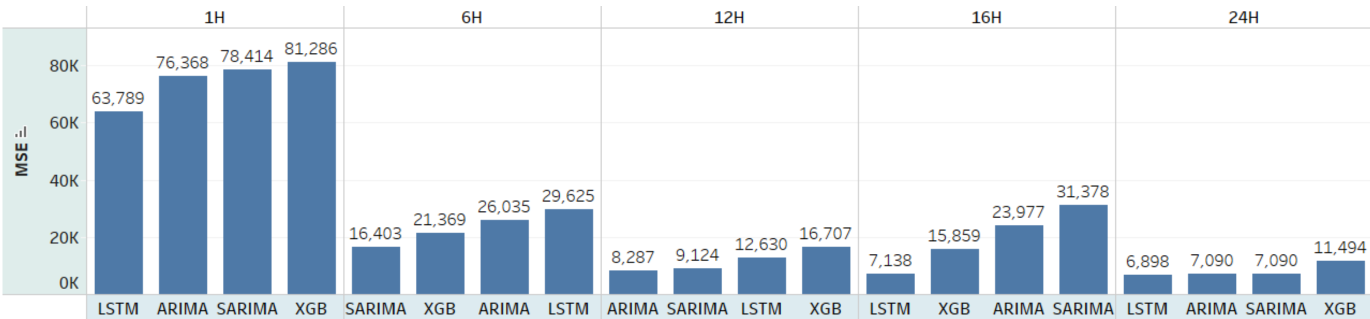

Fig. 20: The MSE scores of the trained models for each sample rate are arranged in **descending** order. (Lower is better)

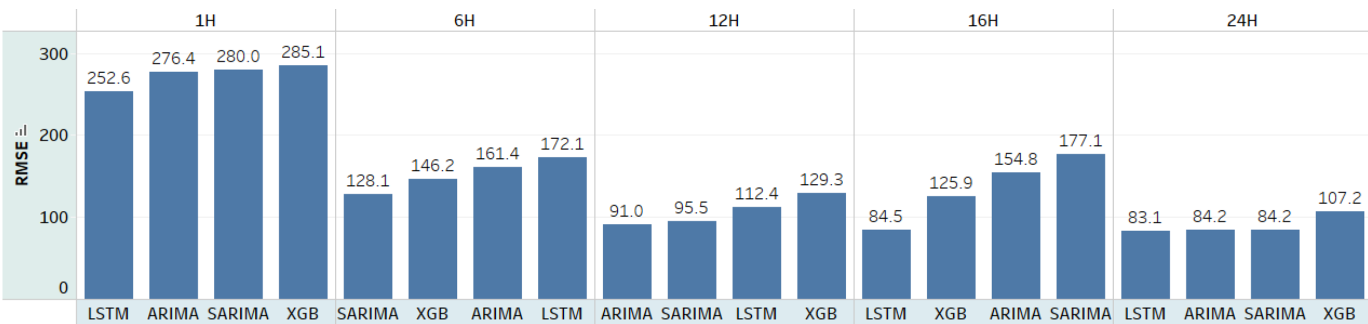

Fig. 21: The RMSE scores of the trained models for each sample rate are arranged in **descending** order.(Lower is better)

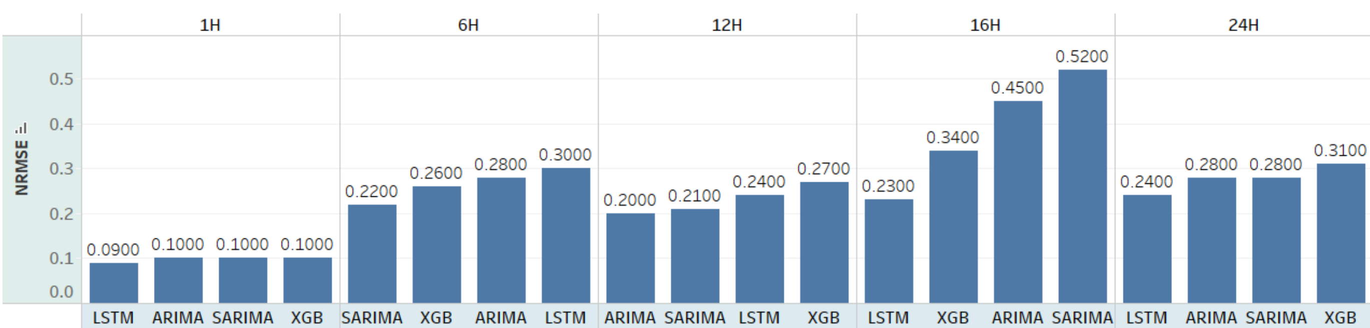

Fig. 22: The NRMSE scores of the trained models for each sample rate are arranged in **descending** order. (Lower is better)

For each performance metric, we studied thoroughly and created the visuals from figure 17 to 22, where we have plotted the performance metrics vs models for each sample rate and sorted the best results in left side by sorting in ascending or descending based on whether the performance metric is better when its higher or lower for each sample rate bin. For example, with R^2 score, we have arranged per sample rate bin into ascending order because R^2 best when its higher. Which lead to result where we could see LSTM performed well under 1H, 16H, and 24H. However, SARIMA dominated in 6H and in 12H ARIMA performed well.
In another performance metric, such as NRMSE, we have sorted the results in descending order in figure 22, because the lower the value of NRMSE, its considered better. If we observe figure 22, we can see LSTM performed well under 1H, 16H, and 24H. However, SARIMA dominated in 6H and in 12H ARIMA performed well.
If we count the number of asterisk* (i.e. the number of time the model performed in any of the performance metrics and beat other models) in table 1, we can see that LSTM has 16 asterisk*, and far ahead of any other competitor (closest competitior is ARIMA with 8 asterisk*). The lowest performer was XGBOOST with only 1 asterisk*.

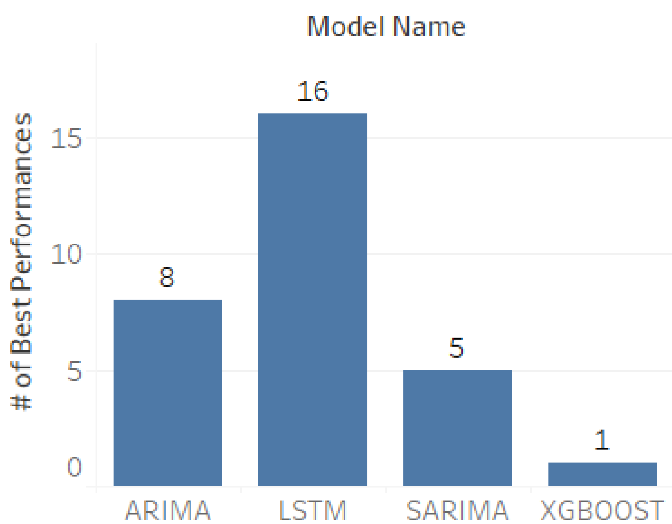

Fig. 23: The number of times the model had achieved best performance in any of the metrics.

As seen in figure 23, it is evident that for current study, LSTM performed best in most number of performance criteria than other models. The performances ranklist is: LSTM (best), ARIMA, SARIMA, and lastly XGBOOST.

Some of the best model's training, testing, and prediction time-series plot is provided below (the blue lines are training, red lines are actual testing values, and the green lines are the forecasted values with trained model):

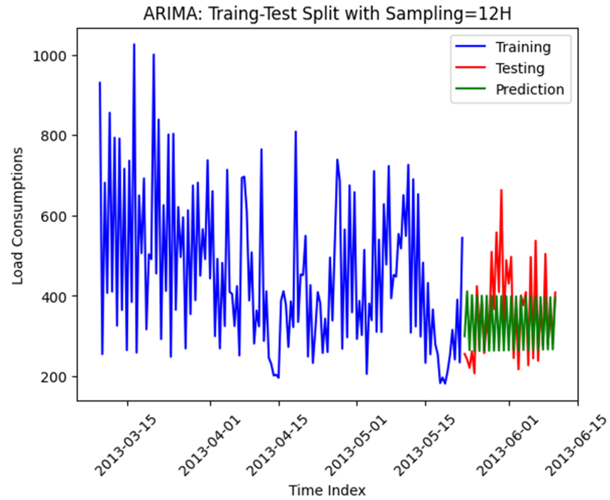

Fig. 24: One of the best ARIMA prediction.

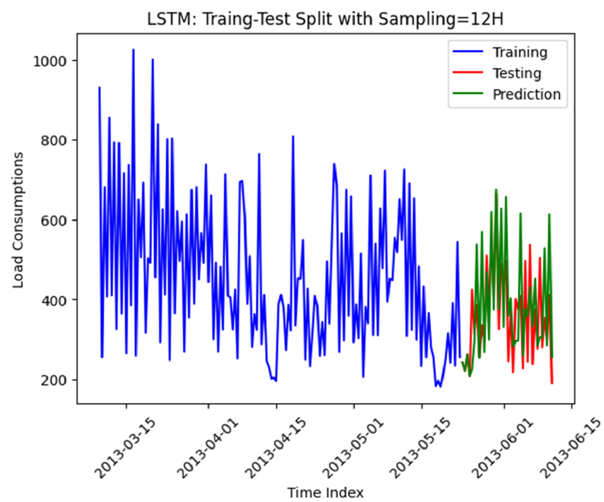

Fig. 25: One of the best LSTM prediction.

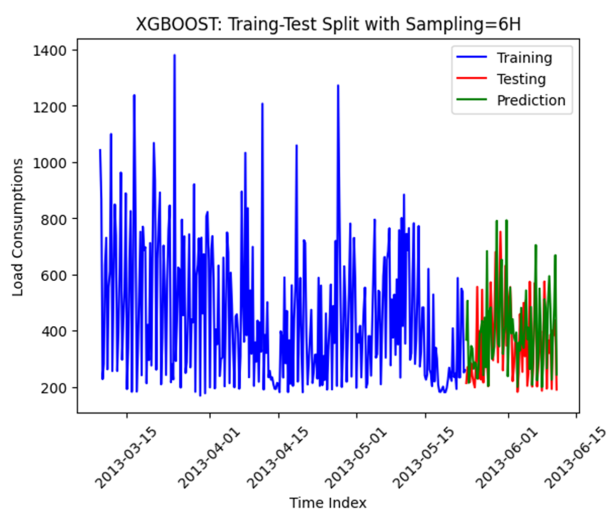

Fig. 26: A best performing XGBOOST prediction.

V. Conclusion & Future Work

The performance of the model suggests that deep learning based approach outperforms machine learning based models in the same train:test split and logistics. The performance of deep learning based approach dominated,where the second best performer is far behind.

One of the limitation of the current study is to predicting only single house, and comparing among different sample rate in single house channel, however, original plan was to compare among all five house but, due to logistical bottleneck, the authors had to limit with keep sticking with house#4.

The UK-DALE dataset is static dataset, and not a real time dataset to train onto, which is a limitation for this study, whereas with live data it is possible to explore much more in current study and even create a software for forecasting in demand side and supply side also for greater understanding of human behavior in load consumption and economic welfare of a state.

For future work, adding adversarial attack, privacy attack, and explainable artificial intelligence for model interpretability would be priority. With more logistical power, opportunity, and patience it is possible to run forecasting on house#1 in UK-DALE which is the largest data house in the whole dataset. Exploring other deep learning techniques to train a fine tuned forecasting model maybe a prospective area for research and experiments.

References

[1] Al Amin, M. A., & Hoque, M. A. (2019, March). Comparison of ARIMA and SVM for short-term load forecasting. In 2019 9th annual information technology, electromechanical engineering and microelectronics conference (IEMECON) (pp. 1-6). IEEE.

[2] Nepal, B., Yamaha, M., Yokoe, A., & Yamaji, T. (2020). Electricity load forecasting using clustering and ARIMA model for energy management in buildings. Japan Architectural Review, 3(1), 62-76.

[3] Chodakowska, E., Nazarko, J., & Nazarko, Ł. (2021). Arima models in electrical load forecasting and their robustness to noise. Energies, 14(23), 7952.

[4] Karthika, S., Margaret, V., & Balaraman, K. (2017, April). Hybrid short term load forecasting using ARIMA-SVM. In 2017 Innovations in Power and Advanced Computing Technologies (i-PACT) (pp. 1-7). IEEE.

[5] Wang, Y., Sun, S., Chen, X., Zeng, X., Kong, Y., Chen, J., ... & Wang, T. (2021). Short-term load forecasting of industrial customers based on SVMD and XGBoost. International Journal of Electrical Power & Energy Systems, 129, 106830.

[6] Abbasi, R. A., Javaid, N., Ghuman, M. N. J., Khan, Z. A., Ur Rehman, S., & Amanullah. (2019). Short term load forecasting using XGBoost. In Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33 (pp. 1120-1131). Springer International Publishing.

[7] Dong, Y., Ma, X., & Fu, T. (2021). Electrical load forecasting: A deep learning approach based on K-nearest neighbors. Applied Soft Computing, 99, 106900.

[8] Bozkurt, Ö. Ö., Biricik, G., & Tayşi, Z. C. (2017). Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market. PloS one, 12(4), e0175915.

[9] Papadopoulos, S., & Karakatsanis, I. (2015, February). Short-term electricity load forecasting using time series and ensemble learning methods. In 2015 IEEE Power and Energy Conference at Illinois (PECI) (pp. 1-6). IEEE.

[10] Kwon, B. S., Park, R. J., & Song, K. B. (2020). Short-term load forecasting based on deep neural networks using LSTM layer. Journal of Electrical Engineering & Technology, 15, 1501-1509.

[11] Chung, W. H., Gu, Y. H., & Yoo, S. J. (2022). District heater load forecasting based on machine learning and parallel CNN-LSTM attention. Energy, 246, 123350.

[12] Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-term residential load forecasting based on LSTM recurrent neural network. IEEE transactions on smart grid, 10(1), 841-851.

[13] Bejarano, G. (2018, June). PhD Forum: Deep Learning and Probabilistic Models Applied to Sequential Data. In 2018 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 252-253). IEEE.

[14] Rafiq, H., Shi, X., Zhang, H., Li, H., Ochani, M. K., & Shah, A. A. (2021). Generalizability improvement of deep learning-based non-intrusive load monitoring system using data augmentation. IEEE Transactions on Smart Grid, 12(4), 3265-3277.

[15] Zhou, X., Feng, J., Wang, J., & Pan, J. (2022). Privacy-preserving household load forecasting based on non-intrusive load monitoring: A federated deep learning approach. PeerJ Computer Science, 8, e1049.

[16] Chen, Y., Tan, Y., & Zhang, B. (2019, June). Exploiting vulnerabilities of load forecasting through adversarial attacks. In Proceedings of the tenth ACM international conference on future energy systems (pp. 1-11).

[17] Xu, W., & Teng, F. (2023). Availability Adversarial Attack and Countermeasures for Deep Learning-based Load Forecasting. arXiv preprint arXiv:2301.01832.

[18] Ardito, C., Deldjoo, Y., Di Noia, T., Di Sciascio, E., Nazary, F., & Servedio, G. (2023). Machine-learned Adversarial Attacks against Fault Prediction Systems in Smart Electrical Grids. arXiv preprint arXiv:2303.18136.

[19] Tian, J., Wang, B., Li, J., & Konstantinou, C. (2022). Adversarial attack and defense methods for neural network based state estimation in smart grid. IET Renewable Power Generation, 16(16), 3507-3518.

[20] Mir, A. A., Alghassab, M., Ullah, K., Khan, Z. A., Lu, Y., & Imran, M. (2020). A review of electricity demand forecasting in low and middle income countries: The demand determinants and horizons. Sustainability, 12(15), 5931.

[21] Patel, R., Patel, M. R., & Patel, R. V. (2019). A review: Introduction and understanding of load forecasting. Journal of Applied Science and Computations (JASC), 4(4), 1449-1457.

[22] Energy demand: Three drivers. ExxonMobil. (2022, October 5).https://corporate.exxonmobil.com/what-we-do/energy-supply/outlook-for-energy/energy-demand

[23] Kelly, J., & Knottenbelt, W. (2015). The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Scientific data, 2(1), 1-14.

[24] Moon, J., Kim, Y., Son, M., & Hwang, E. (2018). Hybrid short-term load forecasting scheme using random forest and multilayer perceptron. Energies, 11(12), 3283.