Indian Journal of Energy and Energy Resources (IJEER)
ISSN: 2583-1186 (Online), Volume-4 Issue-3, May 2025

Time Series Forecasting Analysis for Automated Smart Meter Reading System

Sujaudeen N., Lakshmi Priya, Gayathri Venkatesan, Dharshni M

***Abstract**: The Smart Meter Reading System modernizes traditional meter reading by enabling real-time energy monitoring and short-term consumption predictions for sectors like banking and automotive. The research in this paper focuses on the comparative application of time series forecasting techniques for enhancing the performance of Automated Smart Meter Reading(AMR) Systems. With a growing need for efficient energy management, especially in the context of smart grids and real-time analysis, this study explores how advanced machine learning and deep learning models can predict electricity energy consumption based on smart meter data. The research uses a real-world dataset from the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/290/tamilnadu+electricity+board+hourly+readings. The study leverages time series forecasting models, including ARIMA, SARIMA, SARIMAX, LSTM, XGBoost, and CATBoost, to capture trends, seasonality, and long-term dependencies in sequential data. Each model is evaluated on benchmark metrics such as Root Mean Squared Error(RMSE) and Mean Squared Error (MSE) to measure forecasting accuracy. We have observed that the best models for our purpose of short-term predictions are our two ensemble models. We also find XGBoost to have significantly high predictive reliability. Traditional models like ARIMA have not produced adequate results for the energy data. This study is significant as it demonstrates that integrating machine learning and deep learning into AMR systems enhances the intelligence and responsiveness of the overall system. Accurate forecasting allows utilities to make informed decisions, optimize grid load, and foster consumer awareness. The findings advocate for adopting advanced data-driven methods in modern energy infrastructures to promote efficiency, reliability, and sustainability.*

***Keywords**: Time Series Forecasting, SARIMA, SARIMAX, XGBoost, CATBoost, ARIMA, LSTM, Streaming Data, Real Time Data, Short Term Load Forecasting, Automatic Meter Reader(AMR), Smart Meter*

Abbreviations:
AMR: Automatic Meter Reading
SMA: Smart Meter Application
MSE: Mean Squared Error
RMSE: Root Mean Squared Error
ML: Machine Learning
MA: Moving Average
LSTM: Long Short-Term Memory
RNN: Recurrent Neural Network

Manuscript received on 25 April 2025 | First Revised Manuscript received on 30 April 2025 | Second Revised Manuscript received on 04 May 2025 | Manuscript Accepted on 15 May 2025 | Manuscript published on 30 May 2025.

***Correspondence Author(s)**

Dr. Sujaudeen N.*, Assistant Professor, Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai (Tamil Nadu), India. Email ID: sujaudeenn@ssn.edu.in, ORCID ID: 0000-0002-8480-270X

Dr. Lakshmi Priya, Assistant Professor, Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai (Tamil Nadu), India. Email ID: lakshmipriyas@ssn.edu.in

Gayathri Venkatesan, Student, Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai (Tamil Nadu), India. Email ID: gayathri2110485@ssn.edu.in

Dharshni M, Student, Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai (Tamil Nadu), India. Email ID: dharshni2110769@ssn.edu.in

I. INTRODUCTION

Electric power is fundamental to modern society, playing a crucial role in infrastructure, industry, and daily life. It powers essential services such as hospitals, schools, and communication networks, ensuring public safety, education, and seamless connectivity. Electricity drives machinery and production processes in industrial settings, enabling large-scale manufacturing, transportation, and commercial activities. Within households, it supports lighting, heating, cooling, cooking, and the operation of electronic devices, improving comfort and quality of life. However, traditional energy consumption measurement methods, such as analog meters, are inefficient and prone to human error. Utility workers must manually record readings at the end of each billing cycle, an approach that is time-consuming and susceptible to inaccuracies. Even with electronic meters, the system remains outdated, lacking the ability to provide real-time data or long-term energy forecasts. To overcome these limitations, automatic meter reading (AMR) technology has been developed, allowing real-time data collection and transmission to a central server. This system benefits consumers and energy providers by enhancing accuracy, enabling remote monitoring, and offering instantaneous access to energy usage data.

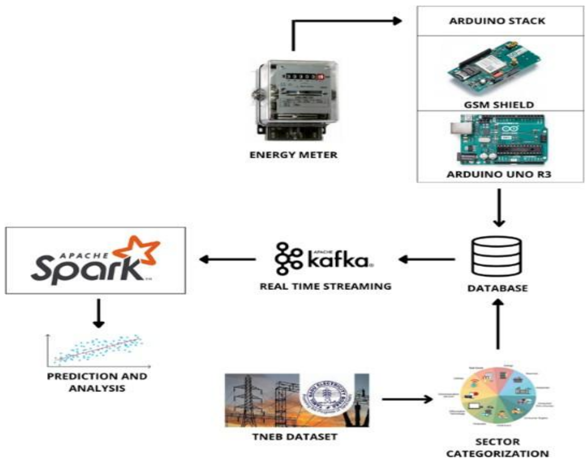
AMR systems provide users with real-time insights into their electricity consumption, helping them make informed and cost-effective decisions. These systems utilize various hardware components, including Arduino-based stack modules, to track power consumption, voltage levels, and current flow. The collected data is then transmitted via wired or wireless connections to a central server for processing. Consumers can access this information through mobile applications or web interfaces, enabling them to monitor usage trends, generate reports, and receive alerts when their consumption surpasses predefined thresholds. Such functionalities encourage energy-saving habits and improve overall efficiency. AMR systems also contribute to predictive maintenance by identifying inefficiencies and potential faults before they develop into significant issues. By detecting abnormal patterns in energy usage, these systems can help prevent electrical failures, optimize resource allocation, and enhance grid stability. This proactive approach minimizes energy waste and ensures a more sustainable and reliable electricity supply for consumers and utility providers alike.

AMR systems also facilitate short-term and long-term predictive modeling for future energy consumption based on energy consumption patterns present in the historical data. This is particularly helpful in industries like manufacturing, banking, and urban planning. This type of classical task of predictive modeling that predicts the future trend changes of a time series is referred to as time series forecasting (TFT) [1]. There exist two variations of TFT: long sequence and short sequence predictions. Long sequence predictions are used to predict longer sequences of data. Long-term forecasting is often used in fields like finance, energy, and even transportation [2]. Short sequence predictions are used to predict shorter time frames. For example, 12 hours to a few days may count as a short sequence, depending on the historical data taken. A long sequence could be anywhere from a few days to multiple months of predictions.

There are many different models used for time series analysis and forecasting. There are many traditional methods for time series forecasting, like autoregressive (AR), exponential smoothing [3], or structural time-series models [1]. With the advancement in the fields of machine learning and deep learning, time series forecasting has allowed for more accurate predictions for a diversity of time-series problems. LSTM is a widely used technique in time series forecasting. From this architecture, the authors of introduced Deep LSTM, which is capable of learning non-linear relationships and can adapt to the complexities in time series data [4]. CatBoost, a gradient-boosting decision tree algorithm, is also used for its efficiency in handling nonlinear relationships within the data [5]. LightGBM is a tree-based learning algorithm that can be made to advance traditional gradient boosting networks, like CatBoost or XGBoost, or can be used alone using proper hyperparameter tuning [19]. Models can also be combined to create ensemble models. The authors of used Facebook's Prophet and XGBoost to create an effective ensemble model for the forecasting of solar power generation [6]. XGBoost has been used in a variety of applications, such as sales volume forecasting [7]. ARIMA, and consequently its variants, is an incredibly popular time series forecasting model [20]. It utilizes the traditional method of AR and adds a moving average component and an integrated order of differencing [21]. This allows it to capture the complexities of the relationships in the data, which has been used for multiple applications [8]. The key to successful forecasting lies in choosing the right representation among these approaches. This paper presents a comparative analysis of these forecasting models, evaluating their effectiveness in the context of AMR-based energy predictions [22]. By leveraging advanced predictive analytics, AMR systems enhance not only billing accuracy and efficiency but also long-term energy management strategies, contributing to a smarter and more sustainable power distribution network [23].

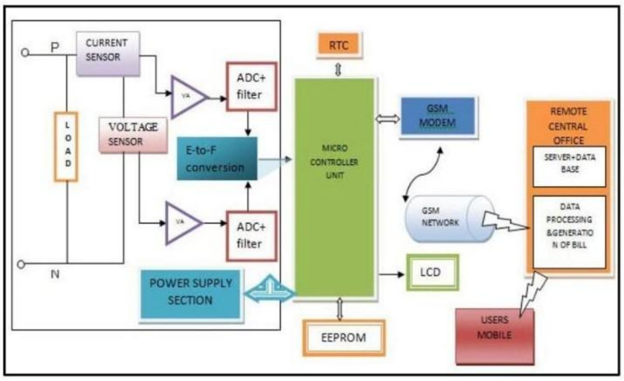
II. METHODOLOGY

The Smart Meter Application [9], or SMA, has been split into four sections: The GSM Arduino module, the Kafka setup module, the Spark architecture module, and the forecasting models module. Figures 1 and 2 display the system's overall flow of architecture. Figure 1 describes the end-to-end system, from the energy meter to the prediction and analysis. The energy meter provides continuous values, which are added to a database. These values are then analyzed using different machine learning and deep learning models using Apache Spark and Kafka.



[Fig.1: End-to-End Smart Meter System]

Direct tampering with the electricity meter to obtain real-time readings is not feasible due to the proprietary protocols used for obtaining meter data. Instead, external signals such as rotating discs, LED pulses, or RS232 pins must be accessed to capture the readings and automatically feed them to an online server. The system utilizes an Arduino Stack, comprising an Arduino YUN and an Arduino GSM Shield, for this purpose. The Kafka setup forms the core architecture of the SMA, designed to handle a continuous stream of real-time data via Kafka Producers, Brokers, and Consumers. The system then integrates MySQL, Spark SQL, and machine learning (ML) analytics for efficient data processing, analysis, and reporting. Spark Streaming plays a crucial role in retrieving and processing real-time data through Discretized Streams (DStreams). MySQL serves as a simple relational database for data storage, meeting the system's minimal attribute requirements. Spark SQL acts as the computational backbone, enabling large-scale data querying and processing. This setup provides real-time insights, analytics, and reporting, as illustrated in Figure 1. The integration of Sparkling Water and MLlib enhances the system's analytical capabilities, allowing it to execute machine learning models that generate real-time predictions and identify anomalies.



[Fig.2: Smart Meter System Prototype]

A. Machine Learning Forecasting Models

Predictive analysis is a powerful tool used across industries to forecast future outcomes based on historical data and statistical algorithms. In the realm of electricity consumption, predictive analytics is crucial for optimizing energy usage, enhancing efficiency, and reducing costs. By analyzing historical consumption patterns, weather data, and other relevant factors, it can accurately forecast future electricity demand. This enables utility companies to anticipate peak usage periods, allocate resources efficiently, and prevent grid overloads or outages. As discussed in the Introduction, one such tool for predictive analysis is time series forecasting. There are many different methods used for time series forecasting. Machine learning approaches are quite common in time series forecasting [10]. Certain approaches involve using models like ARIMA, SARIMA, and SARIMAX. ARIMA is a very common statistical approach, along with SARIMA and SARIMAX. Other approaches include the usage of deep learning models like XGBoost or CatBoost. Ensemble models can also be used by combining different ML or DL models for means of forecasting.

B. DATASET

The dataset was employed from the UC Irvine Machine Learning Repository. This time series dataset was a multivariate dataset with 45781 instances. The dataset itself had 5 columns: ForkVA, ForkkVA, Sector, Type, and ServiceID. The ForkkVA column refers to the power consumption by the sector. Type refers to the sector of Tamil Nadu where the power is consumed. A few sectors that were part of this dataset were Bank, Health Care, Chemical Industry, and Supermarkets. There were overall 20 different sectors. Sector was a boolean variable to determine whether or not the data point could be classified into a sector or not. ForkVA was the voltage used. The components of time series data refer to underlying patterns and structures that can be found within the data itself. These components make up the data, and how the data behaves when analyzed. There are quite a few patterns that may occur:

i. **Trend** - an upwards or downwards movement of data.
ii. **Seasonality** - repeating pattern in data at regular intervals (in this case, can be hourly, daily, or yearly).
iii. **Cycle** - pattern in the data that occurs repetitively after a certain number of observations or periods.
iv. **Irregularity** - fluctuations that cannot be explained by trend or seasonality.
v. **Autocorrelation** - correlation between observations and previous observations in the same time series (e.g., *n* and *n-1* data points).
vi. **Outliers** - extreme observations that do not align with the behavior of the rest of the data.
vii. **Noice** - unpredictable or random variations in the data.

Predicting future data behavior would help better comprehend user and social power consumption regularly. In the next few subsections, we will touch on the models being implemented.

C. ARIMA

Auto-Regressive Integrated Moving Average, or ARIMA, is a popular time series forecasting model. It is a statistical method for analyzing and forecasting historical time series data based on previous observations. The pure ARIMA model consists of three main components: the Autoregressive(AR) Component, the Integrated(I) Component or order of differencing, and the Moving Average(MA) Component. The order of an ARIMA model is denoted as ARIMA(p,d,q), where p is the AR Component, d the I Component, and q is the MA Component. The parameters can be described in Table 1.

ARIMA has been used in multiple different use cases as well. In the research introduced by the authors of [8], the basic ARIMA model is enhanced through different preprocessing and model optimization steps. The authors also highlight the benefit of the statistical basis of ARIMA, particularly in its ability to model linear trends and seasonal components effectively when properly configured. The study conducted by the authors of compares ARIMA, GRU, and LSTM models for time series forecasting [11]. In this study, while ARIMA is outperformed by the deep learning models, it is praised for its simplicity, interpretability, and strong reliability. ARIMA has also been used to build forecasting models tailored to specific patterns in electrical and electronics datasets [12]. The study underlines the ARIMA's advantages in decomposing time series data into its autoregressive, integrated, and moving average components.

Table-I: Parameters of ARIMA

Parameter	Description
p	**AR or Autoregression:** Autoregression is a regression model that uses the dependent relationship between the current observations and previous observations. These past values are utilized in the mathematical relation as discussed later.
d	**Integration**: The integration component uses differencing of current observations to make the time series stationary. Stationary data is data where the statistical properties of the data, like mean, variance, and autocovariance, remain constant over time. Formally, a time series is considered to be so if it is stationary. There are a variety of tests to determine whether a time series can be formally considered, like the Dickey-Fuller test.
	Non-stationary data has changing statistical properties. Usually, this includes things like trends and seasonality. Any variance with time can be considered as non-stationary data.
q	**Moving Average**: This component utilizes the dependency between an observation and a residual error from a moving average model applied to lagged observations. The moving average component depicts the error of the model as a combination of previous error terms.

D. SARIMA

SARIMA is a version of ARIMA. It stands for Seasonal Autoregressive Integrated Moving Average, and is versatile and widely used. It captures both long and short-term dependencies within data. It is also designed to handle seasonality in data. It combines the previously discussed Autoregressive (AR) component, Integrated (I) component, and Moving Average (MA) component
with seasonal components. SARIMA(p,d,q)(P, D, Q,s) is the notation for SARIMA. Table 2 discusses these parameters, as well as the values we used in our

experimentation.

SARIMA has been used for different use cases. SARIMA has been used to forecast temperature trends in Nanjing, China [13]. The authors analyzed historical data, identified seasonal trends, and used the SARIMA model to account for both seasonal and non-seasonal variations in the data.

Table-II: Parameters of SARIMA

Parameters	Description	Values Used
p	Autoregressive component of order	3
d	Integrated component of order	1
q	Moving Average component of order	3
P	Seasonal Autoregressive component of order	2
D	Seasonal Integrated component of order	3
Q	Seasonal Moving Average component of order	2
s	Seasonal period - number of observations per season in the time series	24

E. SARIMAX

Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX), like the SARIMA, is a modified version of the ARIMA model. It implements both seasonal and external factors into it. SARIMAX is invaluable when analyzing time series data that exhibits both specific variations over periods of time. One can consider the SARIMAX model as SARIMA(p,d,q)(P, D, Q,s) + exogenous. Table 3 discusses the parameters for the model used. method called Ordered Boosting. Some of the key features of CatBoost include: Ordered boosting, Gradient Boosting, Categorical Features, Learning Rate, and L2 Regularization.

Table-III: Parameters of SARIMAX

Parameters	Value
p	4
d	1
q	2
P	6
D	3
Q	4
s	24
Exogenous Variable	24-hour cycle

F. Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a special type of recurrent neural network (RNN) designed to handle long-term dependency problems. LSTM networks consist of units called memory cells, each containing three main gates: the input gate, the forget gate, and the output gate. LSTMs overcome the vanishing gradient problem common in traditional RNNs. They can capture long-term dependencies and are effective in tasks requiring memory of previous inputs over long periods. This is demonstrated in the comparative study between LSTM, XGBoost, and ARIMA. LSTM captured complex temporal dependencies and showed a strong accuracy for their dataset [14].

G. XGBoost

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost is very efficient in handling missing values and has built-in support for parallel processing, which helps train large datasets within a reasonable amount of time. For example, XGBoost has been used to preprocess historical sales data [6]. The study emphasised the model's ability to handle non-linear relationships and complex interactions in the data, which demonstrated high forecasting precision for the experimentation conducted.

H. CatBoost

CATBoost, short for "Categorical Boosting," is a powerful machine learning algorithm designed for classification and regression tasks. It stands out for its ability to handle categorical variables without manual encoding, using a

I. Informer

Informer is a transformer-based model designed for long-sequence time-series forecasting. It improves efficiency through a self-attention mechanism called ProbSparse Self-Attention, which reduces computational complexity while maintaining accuracy. By utilizing a distilling operation, Informer compresses redundant information, making it more suitable for handling large-scale forecasting tasks with reduced memory and time costs.

J. LightGBM

LightGBM (Light Gradient Boosting Machine) is a high-performance gradient boosting framework developed by Microsoft. It uses histogram-based learning to speed up training and reduce memory usage. LightGBM is particularly effective for structured data and time-series forecasting, as it handles large datasets efficiently while maintaining high accuracy. Its ability to capture complex feature interactions makes it a popular choice for forecasting problems.

K. Prophet

Prophet is a time-series forecasting tool developed by Facebook that is particularly useful for handling data with strong seasonal and trend components. It uses an additive model that incorporates trend, seasonality, and holiday effects, making it robust to missing data and outliers.

Prophet is designed to be user-friendly, requiring minimal parameter tuning while delivering accurate forecasts for business and economic applications.

L. Ensemble Model: LightGBM + Informer

The idea of ensemble models for time series forecasting is to combine the strengths of multiple individual statistical, machine learning, and deep learning approaches to improve robustness, prediction accuracy, and generalization across varying data patterns. The authors of propose an ensemble model combining the strengths of Facebook Prophet and XGBoost for forecasting solar power output [15].

The model can handle trends and seasonality components, as well as nonlinear patterns and residuals. The ensemble model thus outperforms the individual models, achieving lower MSE and RMSE scores. Another example of ensemble models in forecasting is in the ensemble forecasting framework proposed for predicting the bloom of harmful algae [16]. This uses XGBoost, LightGBM, and CatBoost with an attention-based CNN-LSTM

Retrieval Number:100.1/ijeer.C104304030525
DOI: 10.54105/ijeer.C1043.04030525
Journal Website: www.ijeer.latticescipub.com

Published By:
Lattice Science Publication (LSP)

model to leverage the structured features and the spatio-temporal features of the environment. The authors of have also proposed a hybrid model that combines BiLSTM and CatBoost to improve accuracy and electricity price forecasting. Ensemble models for time series forecasting offer, thus, a powerful and flexible approach by combining the strengths of diverse algorithms [17].

Combining LightGBM and Informer leverages both tree-based gradient boosting and deep learning-based sequence modeling for time-series forecasting. LightGBM captures complex feature interactions and quickly learns patterns in structured data, while Informer excels at handling long-range dependencies in sequential data. By integrating these models, the ensemble benefits from LightGBM's efficiency in feature-based learning and Informer's ability to model temporal dependencies, leading to improved forecasting accuracy. This combination is particularly useful when working with large datasets where computational efficiency and sequence modeling are both critical.

M. Ensemble Model: LightGBM + Informer + Prophet

The ensemble of LightGBM, Informer, and Prophet brings together three distinct modeling approaches to enhance forecasting accuracy. LightGBM efficiently handles structured data, Informer captures long-term dependencies in time-series data, and Prophet models seasonality, trends, and holiday effects. This hybrid approach allows the ensemble to benefit from the strengths of all three models: LightGBM's fast and accurate boosting mechanism, Informer's advanced transformer-based forecasting capabilities, and Prophet's ability to incorporate external factors and irregular patterns. This makes the ensemble highly effective for scenarios involving complex temporal patterns, external influences, and long-sequence forecasting.

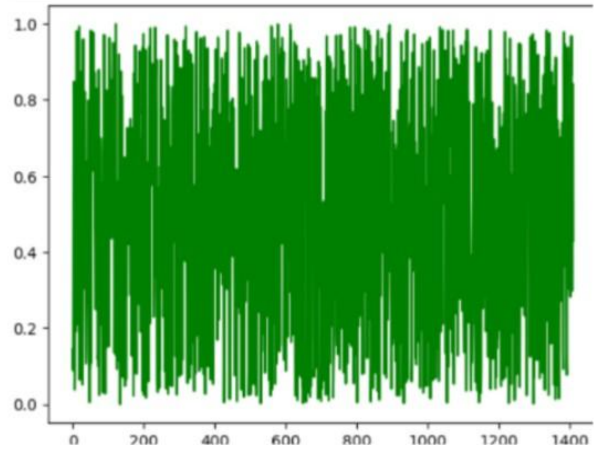
III. RESULTS AND DISCUSSION

We had attempted to predict consumption rates for 12 hours into the future. The results can be seen in the following diagrams and graphs obtained from the analysis of the data using the models SARIMA, SARIMAX, ARIMA, LSTM, CATBoost, XGBoost, and two ensemble models. To test the performance of each model, we have used 2 different evaluation metrics. These are Mean Squared Error(MSE) and Root Mean Squared Error(RMSE). Table 4 displays the formula for the two metrics used.

Table-IV: Evaluation Metrics

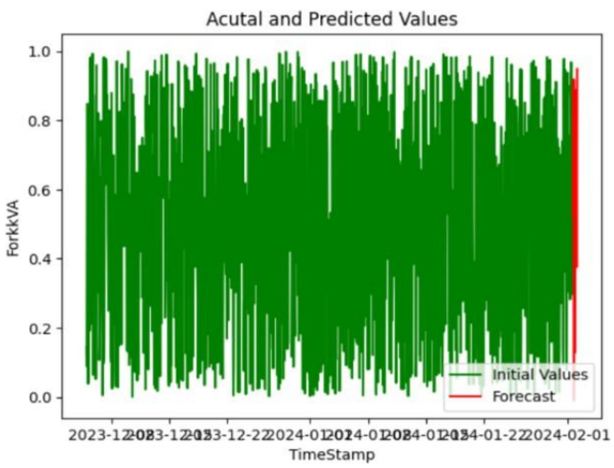
Metric	Formula
Mean Square Error (MSE)	$(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Root Mean Square Error (RMSE)	$\sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

For our dataset, the ARIMA model was unable to predict values properly. It returned null values. This output can be due to a few reasons: insufficient data, overfitting, or the model was unable to learn from the data provided. As seen in Fig. 3, the values are not being predicted.



[Fig.3: ARIMA Results]

The values produced by SARIMA were observed to be in line with the expected seasonality, although the values were much lower than the expected values. However, both actual and predicted values were depicted with similar trends.

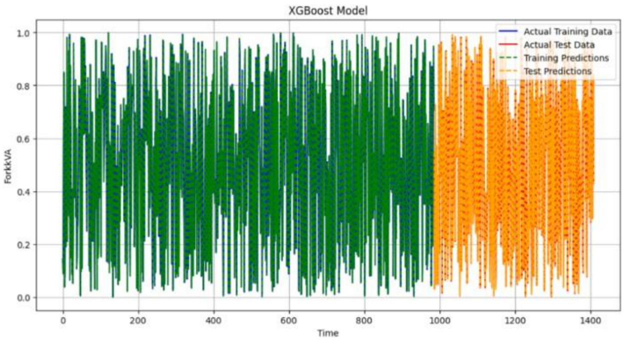


[Fig.4: SARIMA Results]

SARIMAX, as seen in Fig. 6, shows moderate performance in terms of prediction. This could be due to the smaller observed window of time. In this case, it is 12 hours. This may prevent the model from learning meaningful patterns. SARIMAX also focuses on linear relationships, which may result in improper performance for more complex and nonlinear data.

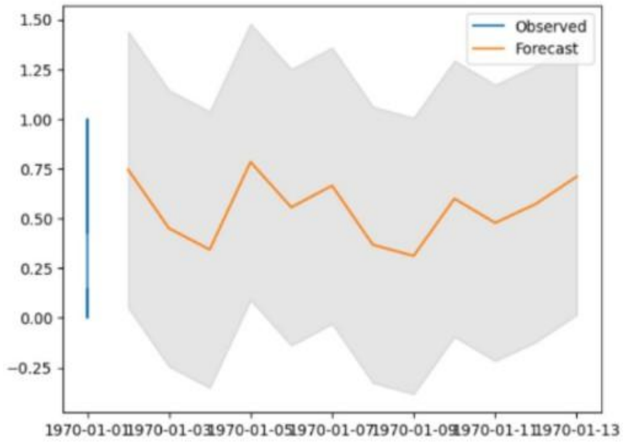
CatBoost, XGBoost, and LSTM all performed well on this dataset. They were able to understand the complexities present within this dataset better than the ARIMA, SARIMA, and SARIMAX models.

XGBoost demonstrated strong performance in prediction. This can be seen in Fig. 5, which shows that the training and testing predictions follow the training and testing seasonality. XGBoost effectively captures complex and non-linear relationships, which is useful for time series data for energy consumption. It also includes both L1 and L2 regularization, which prevents overfitting and ensures better generalization.



[Fig.5: XGBoost Results]

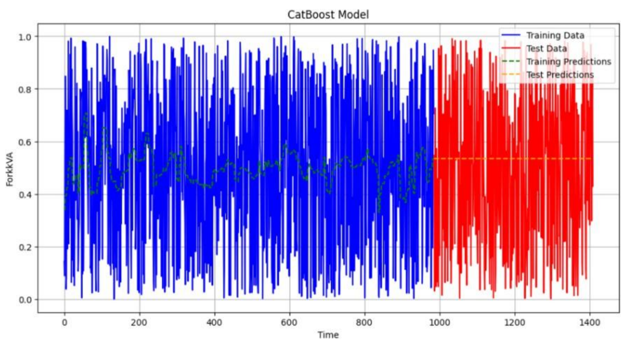
The results for CatBoost are seen in Fig. 7. The prediction lines indicate that CatBoost captures the central tendency or trend of the data, but fails to capture the more intricate high-frequency fluctuations or the volatility of the data. CatBoost is a gradient boosting tree model that focuses on minimizing overall loss and tends to smooth out high-frequency noise. Thus, to would be able to model the average pattern, which is what is being reflected in its predictions.



[Fig.6: SARIMAX Results]

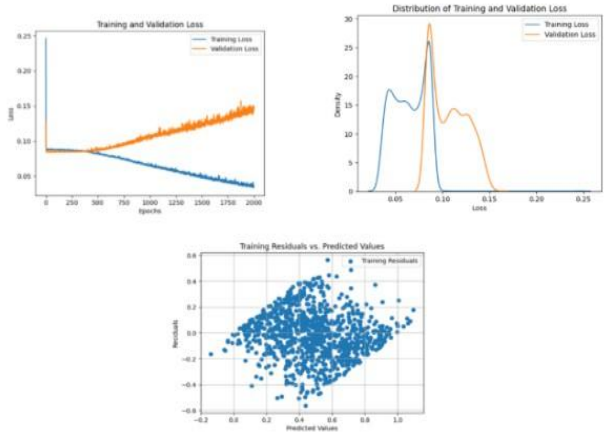
Table-V: Experimental Results

Model	MSE	RMSE
ARIMA	0.329	0.494
SARIMA	0.311	0.407
SARIMAX	0.221	0.398
LSTM	0.306	0.374
CatBoost	0.452	0.441
XGBoost	0.246	0.342
Light GBM + Informer	0.178	0.212
Light GBM + Informer + Prophet	0.089	0.208



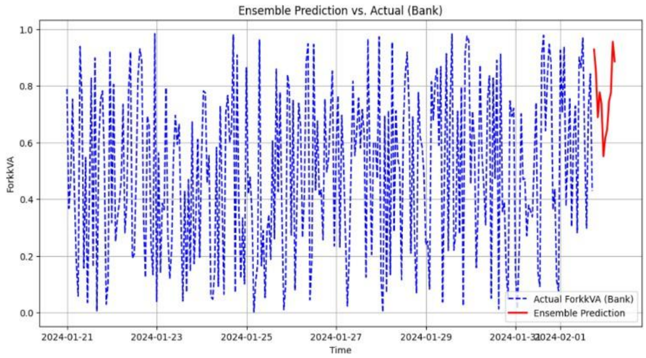
[Fig.7: CatBoost Results]

The LSTM model demonstrates a strong leaning in the training data, which is reflected in the steady decrease of the training loss. This is observed in Fig. 8.



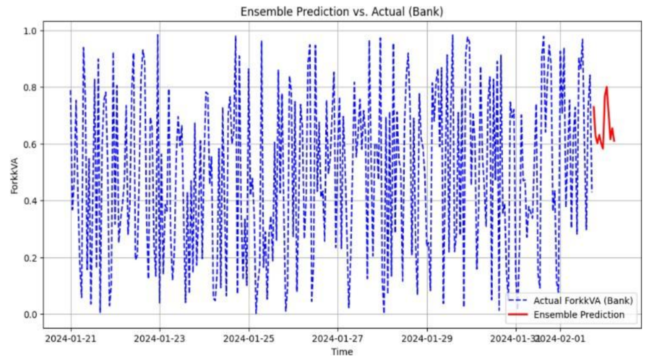
[Fig.8: LSTM Results]

The ensemble models were the best-performing models. Not only did they have low MSE and RMSE scores, but their predicted values for the 12 hours were very close to the displayed seasonality present in the dataset.



[Fig.9: Light GBM + Informer Results]

As seen in Table 5, our model performances are better than other implementations of Prophet and XGBoost alone. In [11], it is observed that XGBoost outperforms Prophet. In our implementation, we have observed that the ensemble model with Prophet is the best for our application. In [13], it is observed that with each implemented dataset, the best model changes between the three tested models: LSTM, ARIMA, and XGBoost. Thus, for our dataset, it is observed that XGBoost, LSTM, and the ensemble models perform the best out of the tested models.



[Fig.10: Light GBM + Informer + Prophet Results]

Retrieval Number:100.1/ijeer.C104304030525
DOI: 10.54105/ijeer.C1043.04030525
Journal Website: www.ijeer.latticescipub.com

Published By:
Lattice Science Publication (LSP)

IV. CONCLUSION AND FUTURE WORK

The implementation of an Automated Meter Reading (AMR) system offers significant environmental benefits. One of the key advantages is the reduction in carbon footprint associated with the traditional meter reading processes. By eliminating the need for manual meter readings, which typically involve vehicle travel and thereby fuel consumption, AMR systems reduce greenhouse gas emissions. This transition to automated systems helps decrease the reliance on fossil fuels and minimizes the environmental impact of energy consumption monitoring. Furthermore, the real-time data provided by AMR systems allows for better energy management and efficiency. Consumers can monitor their usage patterns more accurately and make informed decisions to reduce wastage, leading to overall energy conservation. This heightened awareness and control over energy consumption contribute to a more sustainable and eco-friendly approach to electricity usage.

Additionally, AMR systems facilitate the integration of renewable energy sources into the grid. With the capability to provide detailed consumption data, energy providers can better manage the supply and demand dynamics, ensuring optimal use of renewable energy sources like solar and wind. This efficient energy management not only supports the stability of the power grid but also encourages the adoption of greener energy solutions. By leveraging advanced forecasting models such as ARIMA, LSTM, and XGBoost, AMR systems can predict energy consumption patterns, helping to balance load and integrate intermittent renewable energy sources effectively. Consequently, the shift towards automated, data-driven energy management systems plays a crucial role in promoting environmental sustainability and supporting global efforts to combat climate change.

The Automated Meter Reading (AMR) system proposed in the document represents a significant advancement in energy management for Tamil Nadu. By transitioning from traditional analog meters to a modern AMR system, the project aims to enhance the accuracy and efficiency of energy consumption data collection. This shift is not only crucial for reducing human error but also for providing real-time data that can facilitate more informed and economical energy use by consumers.

Further development and expansion of this idea may provide a new paradigm for the collection and analysis of power consumption for Tamil Nadu. Some prospective steps for the future can be considered as follows:

- **A. Expansion and Integration** – Scaling the AMR system to cover all sectors, including residential, commercial, and industrial, ensuring comprehensive energy management across the board.
- **B. Advanced Analytics** - Leveraging machine learning models such as ARIMA, SARIMA, LSTM, XGBoost, and CatBoost for more accurate and granular forecasting of energy consumption patterns.
- **C. Smart Grid Development** - Integrating the AMR system with smart grid technologies to enhance grid reliability and facilitate the incorporation of distributed energy resources.
- **D. Consumer Engagement** - Developing user-friendly interfaces for consumers to monitor and manage their energy consumption in real time, promoting energy-saving behaviors.
- **E. Policy and Incentives** - Working with government bodies to create policies and incentives that encourage the adoption of energy-efficient practices and technologies.
- **F. Security and safety protocols –** Implementing end-to-end robust security frameworks to protect the smart meter data from unauthorized access, tampering, or cyberattacks [18].

Implementation of the above steps can improve energy efficiency and management in this smart meter. It would also reduce operational costs and make significant strides towards sustainable energy.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Authors Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. B. Lim and S. Zohren, “Time-series forecasting with deep learning: a survey,” *Philos. Trans. A Math. Phys. Eng. Sci.*, vol. 379, no. 2194, 2021, Art. no. 20200209. DOI: http://doi.org/10.1098/rsta.2020.0209
2. Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, “Long sequence time-series forecasting with deep learning: A survey,” *Inf. Fusion*, vol. 97, pp. 101819, 2023. DOI: http://doi.org/10.1016/j.inffus.2023.101819
3. E. S. Gardner Jr., “Exponential smoothing: The state of the art,” *J. Forecast.*, vol. 4, no. 1, pp. 1–28, 1985. DOI: http://doi.org/10.1002/for.3980040103
4. A. Sagheer and M. Kotb, “Time series forecasting of petroleum production using deep LSTM recurrent networks,” *Neurocomputing*, vol. 323, pp. 203–213, 2019. DOI: http://doi.org/10.1016/j.neucom.2018.09.082
5. X. Cui, “A time series forecasting model based on machine learning,” in *Proc. IEEE Int. Conf. Power Intell. Comput. Syst. (ICPICS)*, 2024, vol. 6, pp. 226–230. DOI: http://doi.org/10.1109/ICPICS62053.2024.10796427
6. L. Zhang, W. Bian, W. Qu, L. Tuo, and Y. Wang, “Time series forecast of sales volume based on XGBoost,” *J. Phys. Conf. Ser.*, vol. 1873, no. 1, p. 012067, 2021. DOI: http://doi.org/10.1088/1742-6596/1873/1/012067
7. R. P. Masini, M. C. Medeiros, and E. F. Mendes, “Machine learning advances for time series forecasting,” *J. Econ. Surv.*, vol. 35, no. 5, pp. 1–26, 2021. DOI: http://doi.org/10.1111/joes.12429
8. S. Mehrmolaei and M. R. Keyvanpour, “Time series forecasting using improved ARIMA,” in *Proc. Artif. Intell. Robot (IRANOPEN)*, 2016, pp. 92–97. DOI: http://doi.org/10.1109/RIOS.2016.7529496
9. S. Palaniappan, R. Asokan, S. Bharathwaj, and S. N. Sujaudeen, “Automated Meter Reading System - A Study,” *Int. J. Comput. Appl.*, 2015. DOI: http://doi.org/10.5120/20440-2783
10. S. Kamolov, D. I. Dilshod, and B. Ziyaev, “Machine learning methods in time series forecasting: a review,” *Ann. Math. Comput. Sci.*, vol. 2, pp. 10–14, 2021. DOI: http://doi.org/10.56947/amcs.v2.13
11. P. T. Yamak, L. Yujian, and P. K.

Gadosey, "A comparison between ARIMA, LSTM, and GRU for time series forecasting," in *Proc. Int. Conf. Algorithms Comput. Artif. Intell. (ACAI)*, 2019, vol. 2, pp. 49–55. DOI: http://doi.org/10.1145/3377713.3377722
12. D. M. Xue and Z. Q. Hua, "ARIMA-based time series forecasting model," *Recent Adv. Electr. Electron. Eng.*, vol. 9, no. 2, pp. 93–98, 2016. DOI: http://doi.org/10.2174/2352096509999160819164242
13. P. Chen, A. Niu, D. Liu, W. Jiang, and B. Ma, "Time series forecasting of temperatures using SARIMA: an example from Nanjing," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 394, no. 5, p. 052024, 2018. DOI: http://doi.org/10.1088/1757-899X/394/5/052024
14. I. Paliari, A. Karanikola, and S. Kotsiantis, "A comparison of the optimized LSTM, XGBOOST, and ARIMA in time series forecasting," in *Proc. Int. Conf. Inf. Intell. Syst. Appl. (IISA)*, 2021, vol. 12, pp. 1–7. DOI: http://doi.org/10.1109/IISA52424.2021.9555520
15. R. Gupta, A. K. Yadav, S. Jha, and P. K. Pathak, "Time series forecasting of solar power generation using Facebook Prophet and XG Boost," in *Proc. IEEE Delhi Sect. Conf. (DELCON)*, 2022, pp. 1–5. DOI: http://doi.org/10.1109/DELCON54057.2022.9752916
16. J. M. Ahn, J. Kim, and K. Kim, "Ensemble machine learning of gradient boosting (XGBoost, LightGBM, CatBoost) and attention-based CNN-LSTM for harmful algal blooms forecasting," *Toxins (Basel)*, vol. 15, no. 10, p. 608, 2023. DOI: http://doi.org/10.3390/toxins15100608
17. F. Zhang, H. Fleyeh, and C. Bales, "A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting," *J. Oper. Res. Soc.*, vol. 73, no. 2, pp. 301–325, 2020. DOI: http://doi.org/10.1080/01605682.2020.1843976
18. V. V. Vineeth, V. Ambrish, D. V. Haricharann, V. Harshini, and C. Abilash, "Power theft recognition and data security in smart meter reading of a smart grid," *J. Phys. Conf. Ser.*, vol. 1916, no. 1, p. 012216, 2021. DOI: http://doi.org/10.1088/1742-6596/1916/1/012216
19. Sharma, T., & Sharma, R. (2024). Smart Grid Monitoring: Enhancing Reliability and Efficiency in Energy Distribution. Indian Journal of Data Communication and Networking, 4(2), 1–4. DOI: https://doi.org/10.54105/ijdcn.d7954.04020224
20. Chowdhury, S., & Sen, S. K. (2020). Security in Smart Meter using Iot. International Journal of Engineering and Advanced Technology, 9(4), 2293–2295. DOI: https://doi.org/10.35940/ijeat.d9087.049420
21. Asundi, A. K., Jyoti, P. B., Nagaraj, M. S., & Sultan, S. S. (2019). An Efficient Cryptography Key Management for Secure Communications in Smart Metering. International Journal of Innovative Technology and Exploring Engineering, 8(10), 2908–2916. DOI: https://doi.org/10.35940/ijitee.i7846.0881019
22. Paul, D., Pal, O. K., Islam, Md. M., Mohammad, M., & Babu, R. M. (2023). Design and Implementation of an Efficient Smart Digital Energy Meter. International Journal of Soft Computing and Engineering, 13(1), 25–30. DOI: https://doi.org/10.35940/ijsce.a3600.0313123
23. Gupta, S. K. (2022). Smart Grid System in India. Indian Journal of Energy and Energy Resources, 1(4), 5–6. DOI: https://doi.org/10.54105/ijeer.c1018.081422

AUTHOR'S PROFILE

Dr. N. Sujaudeen is an Assistant Professor in the Department of Computer Science and Engineering with nearly 18 years of experience in teaching and research. He earned his B.E. in Computer Science and Engineering from Madurai Kamaraj University and M.E. in CSE from Anna University. He completed his Ph.D. in Cloud. Computing at Anna University, Chennai in 2020. His research interests span Distributed Computing, Networking, IoT, Big Data Analytics, and Machine Learning. He has published over 14 papers in reputed international journals and conferences, including four in SCI-indexed and Thomson Reuters journals. Notable publications include works in *Concurrency and Computation: Practice and Experience* (Wiley) and *Information Systems and e-Business Management* (Springer). Dr. Sujaudeen is an Oracle Certified Associate in Database Administration and an active member of professional bodies including ACM, CSI, ISTE, and NEN. He serves as a Faculty Advisor for the ACM Student Chapter. He has contributed to and led several funded projects such as smart meter data processing, IoT-enabled solar distillation systems, portable health assistants, and smart campus solutions. His innovations extend to practical applications like a QR code-based smart trolley system and a child eye-safety add-on for smart TVs. Dr. Sujaudeen is also involved in entrepreneurship through his startup **TOSPIE Systems**. He actively organizes and participates in national-leve workshops, FDPs, and conferences, reflecting his commitment to academic excellence and technological advancement.

Dr. S. Lakshmi Priya is an Assistant Professor in the Department of Computer Science and Engineering with over 10 years of teaching experience. She earned her B.E. in Information Technology from Bannari Amman Institute of Technology and her M.E. in Computer Science and Engineering from Sri Venkateswara College of Engineering, both with first-class distinction. During her postgraduate studies, she was awarded the ninth university rank from Anna University. She later completed her Ph.D. at Anna University with a thesis titled "Hyperparameter tuning using Adaptive Bayesian Contextual Hyperband for Time Series Forecasting." Her primary areas of research include Big Data Analytics, Machine Learning, Deep Learning, and Probabilistic Models. She has actively guided undergraduate and postgraduate student projects in these domains and has participated in and conducted several workshops and faculty development programmes. Dr. Lakshmi Priya is a member of IEEE and maintains a strong research profile with multiple publications in reputed journals and conferences. Notable among her publications are recent works such as "Effects of Exogenous Factors and Bayesian-Bandit Hyperparameter Optimization in Traffic Forecast Analysis" and "Hyperparameter Optimization Using Budget-Constrained BOHB for Traffic Forecasting," both published in Springer proceedings. She has also contributed to studies in image processing, including adaptive image fusion, MRI brain image classification, and contour-based face recognition. Dr. Lakshmi Priya continues to combine her academic expertise with applied research, aiming to develop intelligent systems that address real-world challenges in data-driven technologies.

Gayathri Venkatesan is an undergraduate student pursuing her Bachelor of Engineering in Computer Science and Engineering at SSN College of Engineering (2021–2025). She is also an alumnus of the UN Millennium Fellowship of class 2023. With a strong foundation in Artificial Intelligence, Machine Learning, and Computer Vision, her academic journey has been shaped by a deep curiosity for solving real-world problems using cutting-edge technologies. She has gained hands-on experience in working with advanced deep learning models, particularly in video anomaly detection, facial recognition, and NLP-based classification tasks. Her recent work involves developing an anomaly detection system with interpretable AI systems. Gayathri has also contributed to interdisciplinary research, such as the detection of autism using Double-Bounded Rough Sets, showcasing her ability to apply computational methods in healthcare.

Dharshni M is an enthusiastic and detail-oriented Computer Science Engineering student at Sri Sivasubramaniya Nadar College of Engineering, with a deep interest in technology, innovation, and community impact. Her academic journey has equipped her with strong programming skills in Python, Java, and C, as well as a solid understanding of data structures, algorithms, system design, and machine learning. She has worked on a variety of impactful projects ranging from smart energy monitoring systems and anomaly detection frameworks to AI-based educational tools and tourism management platforms. She was honored to be selected as a UN Millennium Fellow in 2023, recognizing her commitment to global development goals. Her work has also been showcased through hackathon achievements and a research paper accepted for presentation at an international conference. Beyond academics, she actively contributes to co-curricular and extracurricular initiatives, serving in leadership roles for technical events, cultural clubs, and outreach programs. A classically trained Bharatanatyam dancer, she has been recognized by the Asian Book of Records and currently leads her college's dance committee. Balancing technology with creativity, she thrives in collaborative environments and constantly seeks opportunities to learn, grow, and contribute to meaningful change through innovation.