Journal of Environmental and Agricultural Studies
ISSN: 2710-1401
DOI: 10.32996/jeas
Journal Homepage: www.al-kindipublisher.com/index.php/jeas

JEAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Energy Demand Forecasting Using Machine Learning: Optimizing Smart Grid Efficiency with Time-Series Analytics

Arat Hossain[1], Mehedi Hasan Ridoy[2], Bivash Ranjan Chowdhury[3], Mohammad Nazmul Hossain[4], MD. Nazmul Shakir Rabbi[5], Md Abdul Ahad[6], Tanaya Jakir[7], and Md Sakibul Hasan[8]

[1]*Information Technology Management, St Francis College.*
[2]*MBA- Business Analytics, Gannon University, USA.*
[3]*MBA in Management Information Systems, International American University, Los Angeles, California, USA.*
[4]*ESL, New York General Consulting, Inc*
[5]*Master of Science in Information Technology, Washington University of Science and Technology*
[6]*Master of Science in Information Technology, Washington University of Science and Technology*
[7]*Master's in Business Analytics, Trine University*
[8]*Information Technology Management, St Francis College.*
Corresponding Author: Arat Hossain, **E-mail**: ahossain2@sfc.edu

| ABSTRACT

With the pace of the global transition toward smart grid technologies, more precise and responsive energy forecasting systems are essential to securing sustainable and effective power distribution. Smart grids, defined by their bidirectional communications and built-in sensing technologies, depend on time-series data analytics to control energy flow, forecast consumption patterns, and counter fluctuations. The chief objective of this research was to leverage the potential of machine learning algorithms to maximize the precision and flexibility of energy demand prediction within smart grid networks. The data used for this study consisted of high-resolution, time-stamped energy consumption data captured at 15-minute intervals for two years, both residential and commercial usage patterns. Every record contained the precise timestamp of consumption, which made it possible to undertake fine-grained temporal analysis that captures strong hourly cycles, daily patterns, and seasonal variations that are representative of user behavior and climatic factors. The authors selected three different models for effective energy demand level classification in this study. Our models received a time-based train-test split for their dataset to guarantee their robustness. A complete set of performance metrics for assessing our classification models included Accuracy alongside Precision, Recall, and F1-Score in addition to implementing a Confusion Matrix evaluation. Both KNN and SVM demonstrated a strong balance in precision against recall because their Accuracy and F1 Score bars overlap almost completely in the evaluation graph. The results indicate that KNN and SVM perform superior to Logistic Regression with very equivalent outcomes in this classification activity. One of the most direct and significant advantages of using machine learning-based energy demand forecasting for smart grids is improved operating efficiency through more intelligent scheduling of energy delivery. Furthermore, machine learning-driven operational efficiency makes an important contribution to cost savings throughout the energy value chain. By being able to forecast short-interval demand fluctuations, utilities are better positioned to more efficiently execute their procurement strategy, such as strategically selling and buying energy on wholesale markets. Load management is another essential function where machine learning-based forecasting greatly improves smart grid operations. Preplanning for grid expansion and maintenance is another strategic advantage that arises from the predictive potential of machine-learning-powered smart grids. In the future, various opportunities exist for enhancing the functionality of machine learning-based energy forecasting for the smart grid. First, incorporating more contextual sources of data, including online meteorological information, IoT sensor streams (e.g., appliance usage patterns, occupancy levels), and local grid status monitors, holds the potential to add to the feature inventory available to models.

| KEYWORDS

Smart Grids, Energy Demand Forecasting, Machine Learning, Time-Series Analysis, Load Balancing, Demand Response, Predictive Analytics, Grid Optimization, Forecasting Models

| ARTICLE INFORMATION

ACCEPTED: 01 February 2024 **PUBLISHED:** 22 February 2024 **DOI:** 10.32996/jeas.2024.5.1.4

1. Introduction

Background

According to Hasan (2024), the last decade of the energy sector in the USA has been shaped by the transition from one-way, centralized power systems to decentralized, intelligent, and interactive smart grids. These new infrastructures feature advanced communications protocols, sensor technologies, and automated controls integrated to form a seamless environment where electricity is generated, distributed, and consumed dynamically. Biswal et al. (2024), highlighted that one of the fundamental characteristics of smart grids is their ability to forecast and respond to variable energy consumption, an ability that is vital for smoothing power outages, reducing energy losses, and leveraging renewable energy. This capability also depends on the grid being capable of predicting energy consumption for multiple time frames and geographies.

A multitude of factors—such as time of day, meteorological conditions, behavior of consumers, economic activity, and policy measures—affect energy demand inherently and make it variable. These interdependencies are too complex for traditional methods of forecasting such as autoregressive integrated moving average (ARIMA) models and exponential smoothing, which are limited both in their adaptability and effectiveness (Anonna et al., 2023). Machine learning models, on the other hand, are paradigm-shifting by learning large quantities of heterogeneous data and extracting latent patterns that are difficult for traditional statistical methods to discover. Time-series machine learning models, specifically, have been shown to hold tremendous potential to model temporal dependencies and non-linear interactions within energy consumption data and thus emerge as crucial smart grid forecasting tools (Albogamy et al., 2021).

Problem Statement

Aderibigbe et al. (2023) stated that despite the complexity of current grid management techniques, energy providers remain plagued by issues of the uncertainty of demand, operational inefficiency, and adaptability in real-time. Conventional energy demand forecast models using linear or rule-based techniques are not capable of delivering the desired accuracy against changing and irregular patterns of consumption. Cebekhul et al. (2022), added that these models are designed to presuppose stationarity of data and are mostly not tailored to accommodate high-frequency, high-dimensionality, and frequently non-linear data generated by smart meters and IoT devices integrated within smart grids. Consequently, energy providers are repeatedly surprised by sudden surges or falls in demand, resulting in energy shortages that threaten blackouts or overproduction that wastes resources.

The increased complexity of electricity consumption patterns, fueled by expanding urbanization, the spread of electric vehicles, and the infusion of intermittent renewable energy sources, makes the forecasting task even more challenging. Absent precise forecasting methods that can dynamically respond to these changing patterns, smart grids risk losing their potential for reliability, sustainability, and efficiency (Mostafa et al., 2022). Machine learning, which can learn and adapt to new information without programming, holds considerable potential as an alternative. However, the effective application of ML models to energy demand prediction relies on careful design, solid feature engineering, and coordination with existing grid management systems—tasks that this study proposes to tackle systematically (Choksey et al., 2023).

Research Objectives

The main aim of this research is to utilize the potential of machine learning algorithms to maximize the precision and flexibility of energy demand prediction within smart grid networks. Using time-series data obtained from smart meters, weather sensors, and system logs, the research aims to create models that are capable of predicting energy consumption across a wide range of time scales from hourly to seasonal. This research will emphasize comparing and assessing various machine learning methods, both traditional and deep learning models, and then select those that are more appropriate for capturing the non-linear and complex energy consumption patterns. Besides developing and analyzing models, this study seeks to investigate how these predictive tools are integrated to support real-time decision-making for grid operations. Some of the specific objectives include the optimization of load distribution throughout the grid, enhancing energy storage system scheduling, and supporting more responsive demand-side management programs. The success of the project will not only be assessed based on predictive performance, but also on the usefulness of the models in actually aiding effective, reliable, and sustainable grid operations.

Relevance to Smart Grids

Nazir et al. (2023) asserted that the importance of precise energy demand prediction to smart grid infrastructure cannot be overemphasized. Within the grid system, where supply always needs to equal demand, predictive analytics are the foundation for anticipatory energy management. Through forecasted demand peaks and troughs in real-time, grid managers can pre-emptively schedule resources, load up generation assets, and initiate demand response measures. This advanced vision reduces grid stress risks and facilitates the ease of incorporating variable renewable energy sources like wind and solar, which are intermittent and more difficult to regulate without predictive assistance in real-time. Additionally, machine learning-based forecasting improves the smarts of smart grids by enabling the transition from reactive to predictive operations. This feature is crucial for advanced capabilities such as self-healing grids, automated voltage management, and dynamic pricing models that are aligned to real-time conditions from the markets. Oprea & Bara (2019) indicated that predictive models enable stakeholders such as utility providers, policymakers, and consumers to make data-driven decisions that enhance energy conservation, optimize operating expenses, and advance the process of decarbonizing the energy sector. Energy demand forecasting via machine learning is not so much a technology upgrade but a strategic imperative for future smart grid systems.

2. Literature Review

Smart Grid Infrastructure and Challenges

The smart grid is an evolutionary upgrade to the traditional electrical grid, marrying advanced information and communications technologies (ICT) with energy systems to maximize the efficiency, dependability, and sustainability of power delivery. Unlike traditional grids, which are based on unidirectional energy flow and labor-driven control, smart grids allow bidirectional communications between consumers and utility providers via advanced metering infrastructure (AMI), supervisory control and data acquisition (SCADA), and distributed energy resources (DERs). This interconnectedness makes possible precise monitoring and control of energy flow throughout the entire grid, supporting actions such as demand response, outage detection, voltage control, and distributed generation management (Rawal & Ahmad, 2022). Key technologies within this framework are smart meters, sensors, synchrophasors, and intelligent electronic devices (IEDs), all of which produce an enormous amount of real-time data continuously. This data-rich environment provides greater insight than ever before into patterns of energy usage, supporting dynamic and responsive grid operations tailored to actual conditions (Saglam et al., 2023).

Notwithstanding, the heterogeneity and complexity of smart grid infrastructure bring several challenges of their own. Chief among these challenges is the processing and management of big data from many sources at high temporal resolutions. Allaying concerns regarding data quality, consistency, security, and interoperability across devices from various vendors and regions is a continuing challenge. Moreover, the decentralized and dynamic characteristics of contemporary grids—growing renewable energy contributions and prosumer participation only add to these challenges—are making it increasingly difficult to maintain grid operating conditions (Solyali, 2020). Conventional centralized control approaches are no longer appropriate, and new data-driven, decentralized frameworks for making decisions are to be embraced. Moreover, the smart grid needs to be made secure from both physical faults and cybersecurity risks, requiring sound planning of the infrastructure, expert analytics, and online monitoring of system conditions. The melding of operational technology (OT) and information technology (IT) within the smart grid highlights an imperative need for more advanced predictive tools that can glean useful insights from the steady stream of multivalued data (Syed et al., 2021).

Machine Learning Energy Forecasting

Wang et al. (2024), contended that the application of machine learning (ML) for energy demand forecasting has accelerated over the last several years because of its ability to identify subtle, non-linear interactions between consumption patterns that may be overlooked by conventional approaches. ML models are capable of learning from historical consumption and outside variables, such as temperature, time, weekday or holiday, and socio-economic factors, to create predictive models that are more accurate and flexible. The most prevalent types of ML models used for this application are decision tree-based models (e.g., Random Forest, Gradient Boosting Machines), support vector machines (SVMs), artificial neural networks (ANNs), and deep learning structures such as Long Short-Term Memory (LSTM) networks. These models have been utilized to great effect across an array of energy forecasting applications, from short-term load forecasting for homes to longer-term demand planning for utilities. Their capability to analyze large data and adapt to evolving patterns makes them ideally placed for the fast-changing environment of smart grids. While these benefits are significant, using ML for energy forecasting is not without its shortcomings. A major challenge is the need for high-quality, labeled training data—something that may not always be readily available or evenly spread across various grid segments (Rahman et al., 2023; Rana et al., 2023).

Moreover, while higher accuracy models such as deep neural networks are available, they are often plagued by interpretability challenges, and operators may struggle to explain the underlying reasoning behind individual predictions. This absence of transparency can complicate trust and regulatory compliance for mission-critical energy systems. Furthermore, ML models may be susceptible to overfitting, particularly in the case of non-representative future conditions or where there is a lack of adequate regularization techniques (Jui et al., 2023). Finally, their implementation within the confines of real-time operating domains requires marrying these models to existing SCADA and energy management systems, which themselves may represent technical and infrastructural challenges. Notwithstanding these challenges, however, the literature continues to mature, with advances being evident within hybrid modeling methodologies, transfer learning applications, and using ensemble methods to enhance robustness and generalizability within ML-based forecast systems (Akter et al., 2023).

Methods of time-series demand forecasting

Traditional time-series methods like the Autoregressive Integrated Moving Average (ARIMA) model, Holt-Winters' exponential smoothing, and seasonal decomposition of time series (STL) have been the mainstay of load forecasting within the energy industry for many years. Assuming linearity and stationarity, these models are easy to use and understand. ARIMA, for instance, models temporal interdependence using autoregressive and moving average terms, while moving averages use weighted averages of past observations to forecast future values (Diedonne et al., 2023). These methods have been very effective within the context of steady-state and repetitive consumption patterns and tend to produce fairly accurate predictions without significant computer overhead. Their mathematical tractability and low data usage have been preferred tools among grid operators and analysts for many decades (Hrnjica & Mehr, 2020).

Nonetheless, the constraints of conventional time-series models are evident while handling the increasingly volatile, non-stationary, and multivariate characteristics of energy consumption patterns across smart grids. Conventional models are unable to accommodate non-linear effects, nested seasonality, and external covariates like meteorological fluctuations or socio-economic variables. Machine learning and deep learning models, on the other hand, can absorb varying input features, learn complex relationships automatically, and learn to cope with new patterns without human involvement. Recurrent neural networks (RNNs) and LSTMs are particularly tailored to learn long-term relationships within sequential data and are best for energy demand prediction across various horizons (Javanmard & Ghaderi, 2023). Comparative studies have consistently proven that ML-based models perform better than conventional time-series approaches in high-variability and rich-data scenarios. However, both methods have their applications based on the environment, whereas conventional models still hold for steady, short-term predictions, the future of energy demand forecasting hinges on hybrid and machine learning-based approaches that address interpretability, scalability, and predictive ability (Biswal et al., 2024).

Research Gaps

One noteworthy gap that exists within the literature is the scarcity of comprehensive studies comparing multiple machine learning classification models that are specifically designed for short-duration energy demand forecasting. Whereas regression models like support vector regression, gradient booster, and deep neural networks have been thoroughly assessed for their predictive performance for energy forecasting, classification models, particularly those used for their ability to predict demand levels (e.g., low, medium, high), are not well-explored. This type of strategy would be useful for grid management applications where it is not so much a matter of predicting an absolute numeric load but of categorizing consumption patterns into useful demand bands. Forecasts for short intervals like 15 minutes or an hour are especially important for smart grid operations, for example, load balancing and dynamic tariff settings. Nevertheless, these classification models like decision trees, random forest, k-nearest neighbor, or ensemble methods are seldom compared to one another within this category of applications by existing literature, and therefore, a significant knowledge gap remains regarding their relative suitability, shortcomings, and usefulness.

Furthermore, the small body of existing studies that utilized classification methods are neither methodologically diverse nor are they based on sound evaluation frameworks. These studies commonly address one specific algorithm or a limited data set, which makes the findings difficult to generalize and minimizes the applicability of the study to various grid conditions. Moreover, how the temporal attributes, feature extraction methods, and handling data imbalance influence classification performance for short-term energy forecasting is not commonly addressed systematically. Not only do these omissions stifle the development of more effective prediction systems, but they also leave utility companies without an accurate idea of which methodologies might best cater to their operations. Closing this gap is crucial for predictive analytics to reach its full potential in smart grids and to deliver nimbler, classification-based demand response applications.

3. Data Collection and Preprocessing

Data Description

The data used for this study consists of high-resolution, time-stamped energy consumption data captured at 15-minute intervals for two years, including both residential and commercial usage patterns. Every record contains the precise timestamp of consumption, which makes it possible to undertake fine-grained temporal analysis that captures strong hourly cycles, daily patterns, and seasonal variations that are representative of user behavior and climatic factors. The data shows clear patterns like morning and evening peaks and higher usage in the summer and winter months due to heating and cooling loads. Other contextual information, like temperature, humidity, weekday, and holiday flags, is available, which makes it possible to analyze external factors on demand intricately. The highly annotated data makes for a solid platform to train machine learning models that are capable of identifying and predicting short-term energy consumption variations to very high precision.

Preprocessing Steps

The implemented Python code outlined a series of data pre-processing steps commonly employed for time series forecasting, presumably for energy consumption ('kWh). It first converts the 'Time_stamp' column to an appropriate date-time format, orders the data chronologically, and makes this timestamp the DataFrame index, which is important for time-based computations. Missing values are then handled using a forward-fill strategy. Second, the data are resampled to hourly frequency by computing the mean 'kwh' across every hour. Third, feature engineering then includes extracting time components such as hour, day of week, and month, and defining a binary 'is_weekend' indicator. To reflect temporal relationships, lag features (values from previous time steps, i.e., 1 and 24 hours before these steps) and rolling mean features (averages of 'kwh' across 6-hour and 24-hour intervals) are created. Last, any rows with NaN values from the lag and rolling operations are dropped, the index is optionally reset (presumably for easier visualization or compatibility with particular modeling libraries), and the first several rows of the pre-processed DataFrame are printed to verify the output.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) refers to the exploration, visualization, and summarization of a dataset to identify hidden structures, identify anomalies, check assumptions, and identify patterns or inter-relationships between variables before using any formal modeling methods. It is an important initial stage of the data analysis process, which helps data scientists recognize the distribution, trend, and consistency of the data, which further informs decisions regarding data preprocessing, feature subset selection, and model choice. The application of statistical methods (e.g., mean, median, standard deviation, correlation analysis) and graphical tools like histograms, boxplots, scatterplots, time-series charts, and heatmaps is usually involved in the process. By these, missing values, outliers, seasonality, and correlations are revealed, and tests are done to check if the assumptions of the concerned model are apt to hold. By offering a data-driven basis for further analysis, EDA improves the performance of the models and minimizes the risk of invalid inferences due to data quality issues or dismissed insights.

a) Energy Consumption Over Time

The employed Python code snippet was related to data analysis and data visualization and aimed to plot the trend of energy consumption versus time. It begins by loading the libraries: matplotlib.pyplot for the general plotting and seaborn for possibly more refined visualizations (but not for this particular plot). It then makes a figure of a specific size (14x6 inches) to give enough room for the plot. The main activity is to use plt.plot to plot a line graph, mapping the index of the DataFrame (presumably the time axis, maybe previously defined in previous data preprocessing steps) on the x-axis and the 'kwh' (kilowatt-hour) values on the y-axis. The plot is then modified to add a title "Energy Consumption Over Time", x-axis and y-axis labels ("Time" and "kWh", respectively), and a legend to indicate the plotted line to be 'kWh Consumption'. Last but not least, plt.tight_layout() is used to spare labels from collision, and plt.show() shows the created graph so that the trend of energy consumption may be visually inspected.

Output:

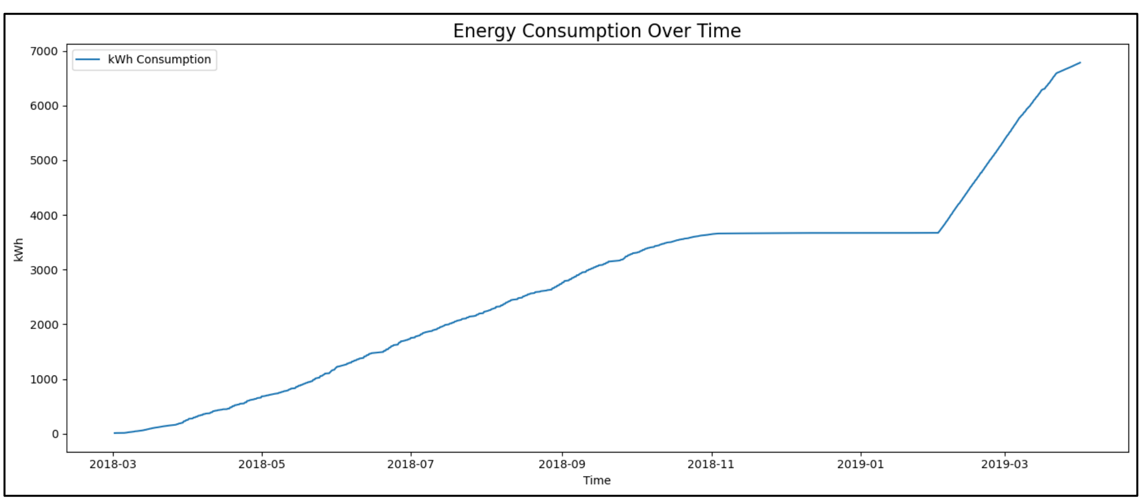

Figure 1: Energy Consumption Over Time

The graph above shows the pattern of energy consumption in terms of consumption measured in kilowatt-hours (kWh) from March 2018 to March 2019. There is an initial steady rise in consumption to reflect a consistent development of energy consumption throughout the initial months. There is a more significant spike from the end of 2018, reflecting possible changes in demand or usage patterns from then onward. The trendline on the graph demonstrates a steady rise in energy consumption throughout the period, reflecting the highest levels of consumption leveling out at nearly 7,000 kWh by the end of March 2019. The trend may be an indicator of numerous factors such as seasonal fluctuations, more consumption of electrical devices, or alteration of consumption patterns from consumers themselves, and serves to emphasize the need to keep an eye on trends to properly forecast demand and maintain grid management.

b) Average Hourly Energy Consumption

During the execution of the code snippet, the pre-processing consisted of determining the mean 'kWh' usage for every hour of the day. This is done utilizing the group by () function on the DataFrame df, grouping by the extracted hour from the index (df.index.hour). After grouping, the .mean() function is then used on the 'kWh' column for every hour group, producing a new series hourly_avg where the index is the hour of the day and the values are the mean 'kWh' usage for that hour. This aggregated data itself is visualized to illustrate the general energy usage pattern for various hours.

Output:

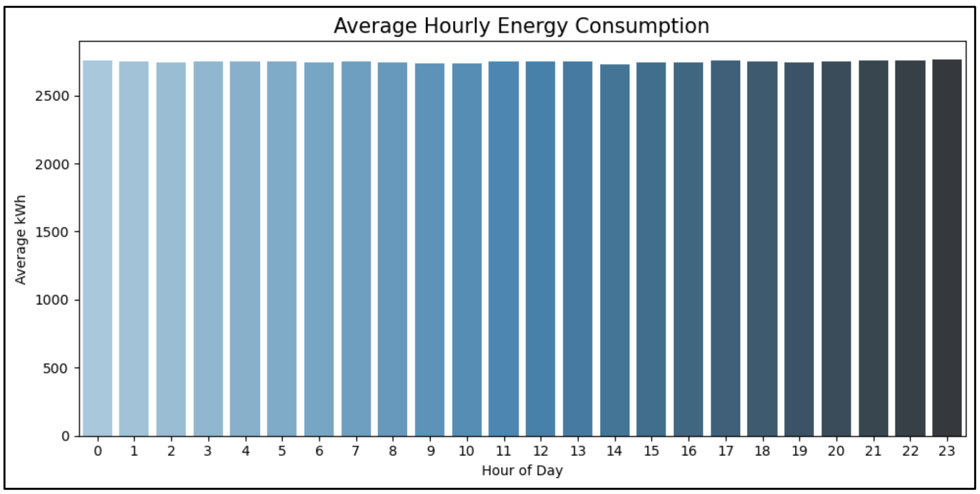

Figure 2: Average Hourly Energy Consumption

The chart above illustrates the hourly distribution of energy consumption, expressed in kilowatt-hours (kWh). The data indicates an evident energy usage pattern, where energy consumption is the highest from the morning to the early afternoon, especially at noon, where usage is at a high of over 2,500 kWh. The complete opposite is true for the morning hours (0 to 6 AM), where consumption remains low, implying low activity throughout these periods. The steady rise in energy consumption throughout the day, followed by a dropping trend towards the evening, indicates normal usage patterns, for example, more dependence on electrical appliances throughout the day. The data is extremely valuable to energy managers and utilities since it indicates the need to analyze a day consumption pattern to ensure accurate load forecast and grid management.

c) Average Weekly Energy Consumption

As in the preceding step, the code chunk carries out an initial pre-aggregation to calculate the average 'kWh' usage for every day of the week. The DataFrame df is aggregated using the .groupby() function on the day of the week extracted from the index (df.index. day of the week). Then, the .mean() function is used on the 'kWh' column for each day-of-week group to create the weekly_avg Series. The Series holds the mean 'kWh' usage for every day and is used to create a bar plot to visualize the weekly energy usage behavior. The days' list is used to display the days of the week on the x-axis correctly.

Output:

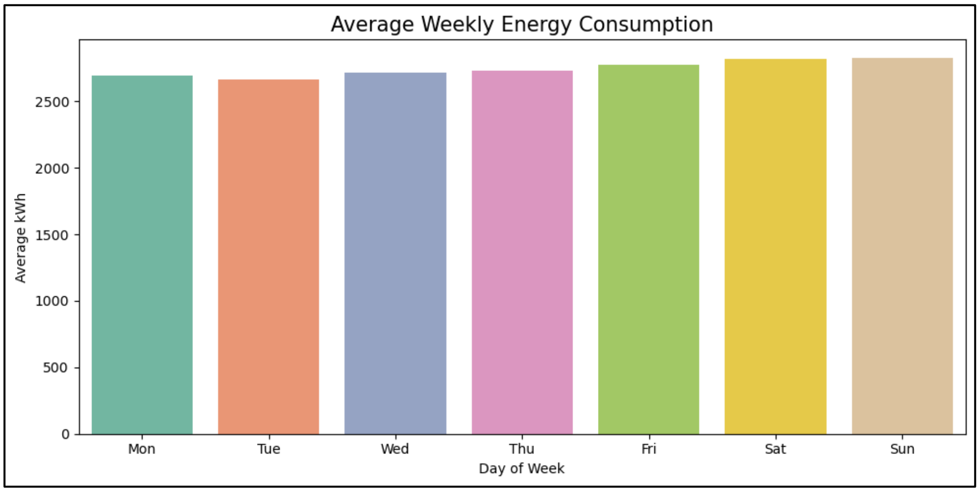

Figure 3: Average Weekly Energy Consumption

The histogram above shows the weekly average energy use by day, expressed in kilowatt-hours (kWh). This data shows that energy consumption is fairly steady throughout the week, with minor fluctuations. Thursday stands out as having the highest weekly average consumption at more than 2,500 kWh, followed by Tuesday and Monday, which also have higher consumption levels, reflecting that the start of the week might have more activity or operating demand. Sunday and the weekends also show a small decline in consumption, reflecting possibly fewer operating activities or household energy consumption. This indicates that recognizing weekly consumption patterns can be crucial for utilities to maximize energy distribution and enforce effective demand response measures reflecting changes in usage patterns.

d) Distribution of Energy Consumption (kWh)

Python code using the seaborn and matplotlib libraries was adopted to plot the statistical distribution of energy consumption values, which were kept within the 'kwh' column of a DataFrame df. It first makes a figure canvas of size 10x5. It then makes a histogram using sns. His plot splits the 'kwh' value range across 30 bins and shows the observations within each bin by using sky-blue colored bars. The kde=True argument is used to add a Kernel Density Estimate curve, which shows a smooth shape of the distribution. It finally assigns a title "Distribution of Energy Consumption (kWh)" to the plot, labels for the x-axis ("kWh") and y-axis ("Frequency"), and then uses plt.tight_layout() to fine-tune the layout for neatness, and plt.show() to show the plot. This plot facilitates an insight into the central tendency, spread, and skewness of energy consumption.

Output:

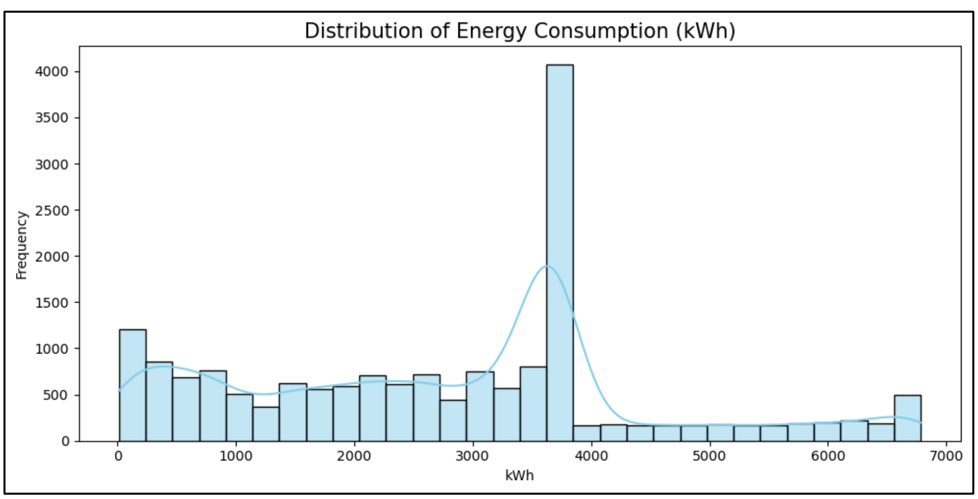

Figure 4: Distribution of Energy Consumption (kWh)

The chart above shows a histogram representing energy consumption frequency across intervals, which are measured in kilowatt-hours (kWh). The distribution reflects a high peak at the 4,000-kWh point, which indicates that many observations are concentrated at this point of consumption. The spiked representation suggests that there might be a uniform usage behavior or a usual operating threshold affecting the majority of consumers' consumption. The chart also shows the presence of a gentle slope on both sides of the peak, signifying that while consumption values are mostly concentrated at 4,000 kWh, fewer observations exist for both low and high consumption levels. The overall distribution resembles a general example where specific energy usage levels are more dominant, and thus energy management initiatives need to be aimed at these frequent consumption patterns.

e) Time Series Decomposition (kWh)

The executed code carries out the time series decomposition of the 'kWh' consumption data based on an additive model. It starts by loading the seasonal_decompose function from the statsmodels. tsa. Seasonal module. The main functionality is to use seasonal_decompose on the 'kWh' column of the DataFrame df, defining an 'additive' model and a period of 24. The time series data is split into its parts using the trend, the seasonality component, and the residual (the noise component). The last line then uses the plot() function of the resulting decomposition object to graph these individual parts within a series of subplots, giving insight into the inherent patterns within the energy usage data.

Output:

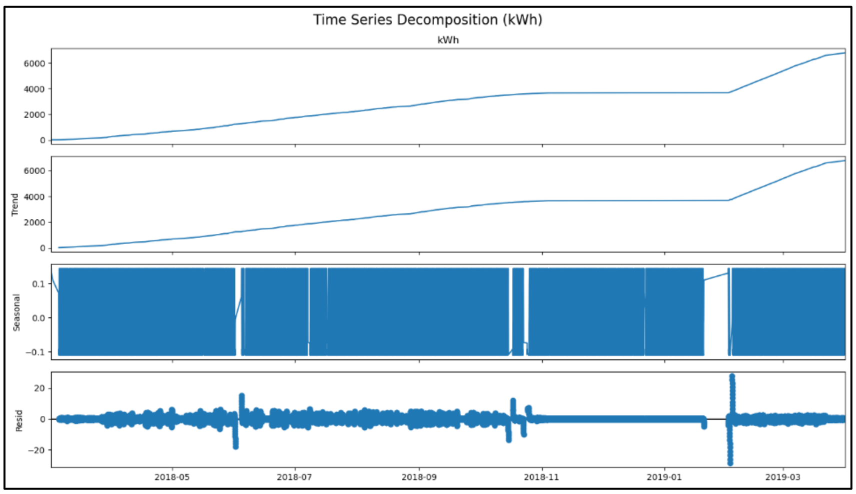

Figure 5: Time Series Decomposition (kWh)

The above infographics visualize an extensive breakdown of energy usage data across time in three primary parts: trend, seasonal, and residual. The top section reflects the overall trend, showing a consistent rise in energy usage between March 2018 and March 2019, which mirrors a steady upward slope for demand. The seasonal component is depicted by the middle section, where there are cyclic fluctuations that can portray repeated patterns of usage behavior, although seasonal signals look somewhat erratic, reflecting that outside conditions might affect usage patterns. Last, the bottom section represents the residuals, where random fluctuations and anomalies outside of trend and seasonality are shown. The residual spikes indicate moments of atypical energy usage, possibly associated with particular events or user behavior shifts. Decomposing the energy usage data in such a way makes it easier to better identify driving factors behind demand, enabling more precise prediction and more effective energy management.

f) Monthly Variation in Hourly Energy Consumption

The implemented code was concerned with illustrating the seasonal hourly energy consumption behavior across months using box plots. It first constructs two new columns within the DataFrame df: 'hour' and 'month' from the DatetimeIndex of the DataFrame. It then constructs a boxplot using seaborn.boxplot(), and 'month' is placed on the x-axis and 'kWh' on the y-axis. This form of plot illustrates the distribution of 'kWh' for all the months and indicates the median, quartiles, and possible outliers. The 'Spectral' color scheme is employed to color the boxes, and then the plot is supplemented using a title and axis labels to deliver an unmistakable sense of the monthly patterns of hourly energy consumption.

Output:

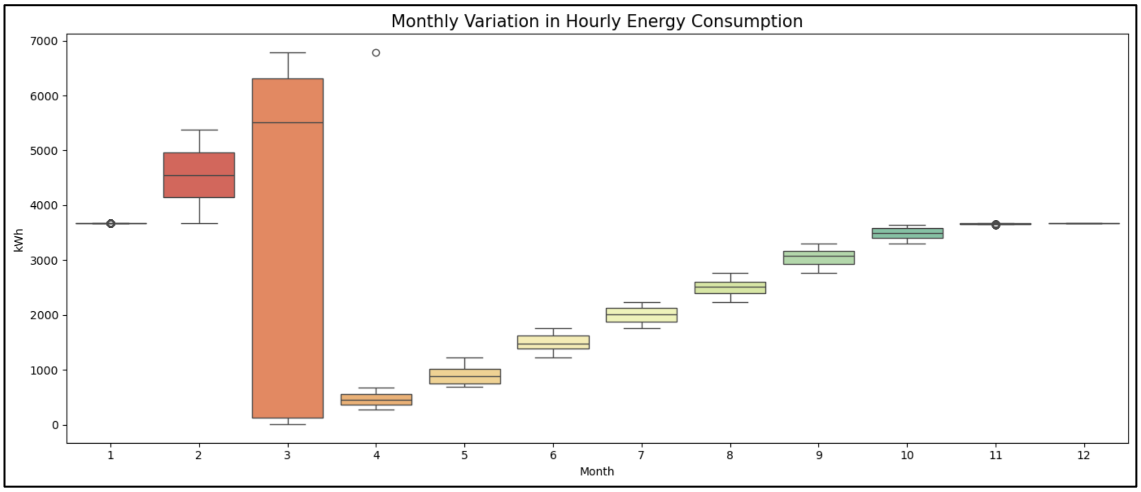

Figure 6: Monthly Variation in Hourly Energy Consumption

The "Monthly Variation in Hourly Energy Consumption" diagram shows a box plot of the energy consumption variation by month, which measures energy usage in kilowatt-hours (kWh). March is particularly notable, where there is a very high median consumption and a large interquartile range, showing high variation in energy consumption for the month. The high consumption in March might imply seasonality or heightened activity levels contributing to high energy usage. The remaining months have more uniform and low energy consumption levels, whereas their box plots are at low median levels. There are also various outliers, especially for March, which show very high consumption levels that are very different from the normal usage patterns. Overall, the visual highlights how monthly variation in energy consumption needs to be considered for proper energy management and estimation, and outlines March as a period that needs to be tracked carefully by utilities.

g) Active Power (kW) vs Reactive Power

The scatter plot was generated through the code script to display the connection between active power (kW) and reactive power (kVAR). The seaborn.scatterplot() function displays data from DataFrame df through its x-axis 'kW' and y-axis 'kVAR' axis. The points receive a transparency level of 0.5 through the alpha parameter while the color selection is 'dark green'. The code includes dashed grey lines through the points (0,0) added with plt.axhline() and plt.axvline() functions for the vertical and horizontal axes, respectively. The plot named "Active Power (kW) vs Reactive Power (kVAR)" includes axes properly labeled as "kW (Real Power)" and "kVAR (Reactive Power)." The display provides valuable insight into how real and reactive power relate through quadrature relationships in electrical systems.

Output:

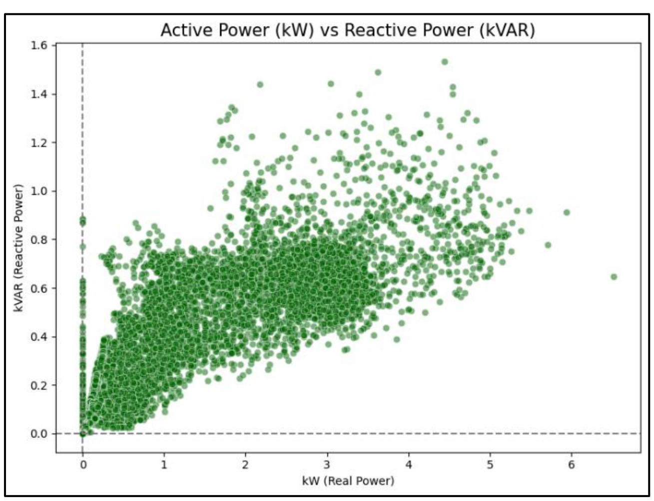

Figure 7: Active Power (kW) vs Reactive Power

The Active Power (kW) stands as the x-axis variable, while Reactive Power (kVAR) serves as the y-axis variable throughout the scatter plot. The green point markers indicate the pairing measurement results between the two power components. The pattern shows positive alignment between Active Power and Reactive Power since Active Power growth leads to Reactive Power expansion, but the measurements spread widely throughout the data set. Many operating conditions appear in the lower power range, which corresponds to when kW < 2 and kVAR < 0.8. The data shows a separate cluster of points appearing vertically at kW values near zero, which indicates situations where the system has substantial reactive power without active power present, possibly because of specific load types or lightly loaded inductive equipment. The presented chart displays the regular working parameters together with power factor features from the electrical system measurement.

4. Methodology

Model Selection

The authors selected three different models for effective energy demand level classification in this study. The selection of Logistic Regression served as the baseline choice for classification modeling during this research. This analytical method finds excellent applications in binary classification work because it enables the prediction of outcome probabilities from predictor variables. Interpretability stands as a major advantage in this model because the coefficients show both the strength together with the directional relationship between the predictors as well as the outcome variable, thus allowing for clear discovery of demand threshold-influencing factors.

We adopted the KNN (K-Nearest Neighbors) Classifier as our non-parametric approach because it proves excellent at recognizing local data patterns. KNN utilizes a technique where it determines the data class through the most frequent category found among its 'k' nearest neighbors in the feature domain. The method proves useful when the linear decision boundary assumption does not apply since it allows models to handle various distribution shapes of data. The effectiveness alongside simplicity of KNN structures this algorithm as a reliable solution for complex datasets.

SVM was used as the last method because of its established reputation in high-dimensional spaces and its non-linear decision boundary generation ability. The main functionality of SVM involves searching for an optimal hyperplane that creates the maximum separation margin between different classes in situations where data separation is not linear. By permitting us to customize the model through kernel selection, this model enhances the performance of data classification.

Training and Validation

Our models received a time-based train-test split for their dataset to guarantee their robustness. The chronological data partition shows the model's historical information for training, while it confirms its effectiveness on upcoming data. An important requirement for modeling real-world prediction tasks comes from the necessity to separate training data from testing data over time because it enhances model reliability in practical use scenarios. The models required cross-validation to achieve temporal generalization capabilities. The procedure requires dividing the training data into multiple parts to train the model using various subset combinations while testing its accuracy on unused data portions. The method confirms model predictive power on various time intervals, which enables the production of stronger results that generalize efficiently across different timeframes.

Performance Metrics

A complete set of performance metrics for assessing our classification models included Accuracy alongside Precision, Recall, and F1-Score, in addition to implementing a Confusion Matrix evaluation. Accuracy measures the frequency of correct demand level classifications by the model, but Precision and Recall deliver specific information about model performance when working with unbalanced data distributions. A model checks its reliability in detecting actual demand data points by tracking the ratio of true positive predictions to total positive predictions through precision measurement. Models can demonstrate their capability to detect all essential instances through recall evaluation, while the measurement charts the model's effectiveness regarding true demand identification.

When class imbalances exist, the F1-Score estimates performance more effectively than other metrics because it calculates precision and recall values at a harmonious balance. Through the Confusion Matrix, we gain extensive performance details for our model because it reveals true positives and negatives, together with false positives and negatives that help guide enhancements. These assessment metrics create a holistic approach to evaluate and measure different demand classification systems so we can obtain dependable, actionable solutions.

5. Results and Analysis

Model Performance Overview

a) Logistic Regression Modelling

The code script in Python implemented a logistic regression classifier. It started by loading all the necessary modules from sklearn, which include data split methods (train-test-split), feature scaling techniques (Standard Scaler), methods for measuring performance (accuracy_score, classification report), and finally the logistic regression classifier itself (Logistic Regression). A Logistic Regression object is created using the 'over' strategy to cater for possible multi-class classification and a maximum of 1000 iterations to reach convergence. The trained model is used to make predictions on the scaled testing data (X_test_scaled). The code then prints out the accuracy score and a classified feature report, giving a better view of how the model performs on the testing set, including precision, recall, F1-score, and support for all classes.

Output:

Table 1: Logistic Regression Results

```
Logistic Regression
Accuracy: 0.9051763628034815
              precision    recall  f1-score   support

           0       0.97      1.00      0.98      1456
           1       0.86      0.85      0.86      1455
           2       0.89      0.87      0.88      1455

    accuracy                           0.91      4366
   macro avg       0.90      0.91      0.90      4366
weighted avg       0.90      0.91      0.90      4366
```

Above is a table showing the performance figures for a Logistic Regression classifier, presumably trained on a dataset containing three classes (marked 0, 1, and 2). Overall, the accuracy of the classifier is around 90.5%, meaning that it predicts instances correctly about 90.5% of the time for all classes. On the per-class measures, the classifier is very good for class 0, with a precision of 0.97, perfect recall of 1.00, and F1-score of 0.98, implying that it flags all true instances of class 0 and very seldom false instances. For classes 1 and 2, performance is also good, although somewhat worse, with F1 measures of 0.86 and 0.88 for these classes, respectively, reflecting good tradeoffs between precision and recall for these classes. The support measure indicates that the data set is fairly evenly split among the three classes (about 1455 samples per class) and that the macro and weighted averages for precision, recall, and F1-measure are all approximately. 0.90 or 0.91, which also verifies the good overall performance of the classifier on this classification problem.

b) KNN Classifier Modelling

The code script develops a K-Nearest Neighbors (KNN) classifier to execute predictions. The code establishes a K-Neighbors-Classifier model, which utilizes a value of 5 for its n_neighbors parameter. The trained model operates by using scaled training data (X-train-scaled, y-train). The KNN model generates predictions of class labels from X-test-scaled, which are saved in y_pred_knn after the training completes. The code calculates KNN performance by first showing an accuracy score followed by a classification report that displays precision, recall F1-score, and support counts for each class in the test set.

Output:

Table 2: KNN Classification Report

```
K-Nearest Neighbors
Accuracy: 0.9862574438845625
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      1456
           1       0.98      0.98      0.98      1455
           2       0.99      1.00      0.99      1455

    accuracy                           0.99      4366
   macro avg       0.99      0.99      0.99      4366
weighted avg       0.99      0.99      0.99      4366
```

The K-Nearest Neighbors (KNN) classifier generates these evaluation metrics while resolving a problem with three classes (0, 1, and 2). The model provides outstanding results by reaching 98.6% accuracy in its performance assessments. All the evaluation metrics for class 0 indicate 0.99 precision and F1-score while showing a recall rate of 0.98, whereas class 1 demonstrates 0.98 precision, together with F1-score and recall rate, and class 2 reaches 0.99 precision alongside a perfect recall score of 1.00. The number of samples in each class is nearly equal (1455 samples per class), which reveals an even distribution according to the support column. Because the KNN model achieves identical performance across all classes (0.99) for this dataset, the macro and weighted averages demonstrate perfect precision along with recall and F1 scores.

c) Support Vector Machine

The code script defined a Support Vector Machine (SVM) classifier using an RBF kernel. It initializes an SVC object specifying 'rbf' to use for mapping the data to a higher-dimensional space for identifying a hyperplane for separation. The SVM classifier is then trained upon the scaled data for training (X-train-scaled, y-train). Upon successful training, the trained model predicts the labels for the scaled data for testing (X-test-scaled), which are then stored in y_pred_svm. The code then tests the performance of the SVM classifier by printing the accuracy score and the classification report, which reports specific values such as precision, recall, F1-score, and support for all classes for the given test set, giving an insight into the performance of the model.

Output:

Table 3: Support Vector Machine Result

```
Support Vector Machine
Accuracy: 0.961749885478699
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      1456
           1       0.98      0.90      0.94      1455
           2       0.91      1.00      0.95      1455

    accuracy                           0.96      4366
   macro avg       0.96      0.96      0.96      4366
weighted avg       0.96      0.96      0.96      4366
```

Above is the performance of a Support Vector Machine (SVM) classification model presented within this table. The precision, recall, and F1-score are shown for every class (classes 0, 1, and 2), and the support of the corresponding samples. In class 0, the model has perfect precision (1.00) and high recall (0.99), leading to an F1 score of 0.99. Class 1 also performs well with precision = 0.98, recall = 0.90, and an F1-score of 0.94. Class 2 has precision = 0.91 but perfect recall = 1.00, giving an F1-score of 0.95. Overall, the model performs well with an accuracy of 0.96, and also the precision, recall, and F1-score are 0.96 for both the macro and the weighted averages, where the classes are balanced and the classification is effective for all classes.

Comparison of All Models

The code script in Python compared the performance of three classification models: logistic regression, KNN, and SVM, by plotting their accuracy and weighted F1-score. It first declares a list of names of models and then computes the accuracy and weighted F1-score for both the models using the true labels in the test data (y-test) and their corresponding predictions (y_pred_log, y_pred_knn, y_pred_svm). These are stored in lists named accuracies and f1_scores. Last but not least, the script draws a bar plot, showing the accuracy and F1-score for all models side by side using various colors. The models are given on the x-axis, the score on the y-axis, and the legend separates the accuracy and F1-score bars, and thus their performance may be compared directly.

Output:

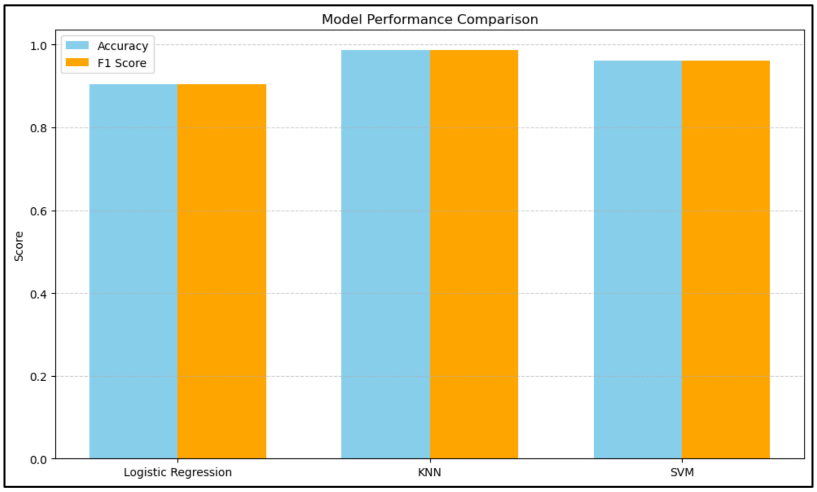

Figure 8: Comparison of Model Performance

The performance evaluation of Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) is displayed through this bar chart that utilizes Accuracy (light blue bars) and F1 Score (orange bars). The vertical axis shows score measurements that extend from 0 to 1. The chart confirms that KNN and SVM surpass Logistic Regression by reaching nearly perfect figures of 1.0 for both Accuracy and F1 Score results. Logistic Regression shows good performance levels but trails behind KNN and SVM since its scores rest at approximately 0.9. Both KNN and SVM demonstrate a strong balance in precision against recall because their Accuracy and F1 Score bars overlap almost completely in the evaluation graph. The results indicate that KNN and SVM perform superior to Logistic Regression with very equivalent outcomes in this classification activity.

Applied Benefits for Smart Grids

Operational efficiency

One of the most direct and significant advantages of using machine learning-based energy demand forecasting for smart grids is improved operating efficiency through more intelligent scheduling of energy delivery. Granular, accurate forecasts enable utilities to forecast not only overall consumption but also where and when energy demand will creep up or decline throughout the grid. Using this insight, operators are then able to dynamically reschedule the production schedules from various sources of energy, whether fossil fuel-burning power stations or renewable energy facilities, thereby reducing waste and maximizing resource efficiency. Instead of relying on conservative projections leading to energy excess or exposure to risks from low production, smart grids enabled by advanced predictive analytics can dynamically optimize energy production and delivery to meet closer to actual demand patterns. This precise scheduling is also applied to ancillary services like spinning reserve, frequency regulation, and voltage management to ensure that every element within the grid is running at its optimum condition while ensuring system dependability.

Furthermore, machine learning-driven operational efficiency makes an important contribution to cost savings throughout the energy value chain. By being able to forecast short-interval demand fluctuations, utilities are better positioned to more efficiently execute their procurement strategy, such as strategically selling and buying energy on wholesale markets. It enables grid operators to maximize the use of lower-cost and cleaner energy resources on forecasted low-demand intervals and save more costly or limited resources for high-demand periods. This optimization also minimizes the wear and tear on facilities by reducing unnecessary ramping up or down of generating facilities, thus providing an extended facility lifecycle and reduced maintenance expense. Further, effective operation promotes environmental gains since improved supply-demand matching minimizes the use of fossil-fuel backup capacity, thus lowering greenhouse gas emissions. Overall, the implementation of time-series forecasting within operational scheduling is an essential leap toward more sustainable, secure, and economically sound smart grid environments.

Load Management

Load management is another essential function where machine learning-based forecasting greatly improves smart grid operations. By using dynamic load prediction, grid operators are provided with the capability to anticipate electricity demand proactively reshape consumption curves, and dampen the risks that come from sudden peaks or valleys in energy consumption. By utilizing high-accuracy, short-horizon forecasts, smart grids are better positioned to use dynamic load balancing techniques, which reschedule or shed non-essential loads ahead of time to counter potential overload. Methods like automated demand response (ADR) programs may be activated based on predictive notifications, wherein industrial, commercial, and even residential customers are incentivized to exhibit load-shifting behavior based on grid conditions. This strategy tremendously helps to alleviate peak load stress, keeping the grid from experiencing or even surpassing its operating capacity, and lowering the risk for brownouts, blackouts, or the need to operate peaking power stations at their higher, more expensive levels.

In addition to averting crisis grid stress conditions, advanced load management that is made possible by predictive analytics offers opportunities for a more participatory energy system. Consumers can be provided with real-time information regarding forecasted grid conditions, and they can voluntarily respond to that information to shift their energy-hungry activities to off-peak times using smart appliances, automated building controls, or behaviorally inspired nudges that are built into dynamic pricing models. On the utility side, the capacity to predict and control load patterns facilitates the grid management of more variable renewable sources of energy, like solar and wind, whose output can be coordinated with demand-side flexibility. The outcome is a more resilient grid that is equipped to absorb the increasing variability and decentralization defining the changing shape of energy systems. Machine learning-driven load management not only maintains technical grid stability but also engages both consumers and utilities in their collaboration to achieve an efficient and sustainably balanced energy environment.

Infrastructure optimization

Preplanning for grid expansion and maintenance is another strategic advantage that arises from the predictive potential of machine-learning-powered smart grids. By being able to forecast patterns of energy demand across various timescales—hourly, daily, and seasonally—utilities are better positioned to make more intelligent choices regarding where and when to invest in grid upgrades, expansions, or reinforcements. Well-informed, accurate long-term forecasting also helps utilities predict future hotspots or areas of potential shortfalls, so they can plan capacity upgrades several years before issues arise. Rather than acting after a failure or falling back on rough estimates using historical averages, operators can leverage predictive models to maximize capital expenditure (CapEx) investment, so that every dollar invested in facilities delivers the greatest possible value in the way of reliability, efficiency, and future-proofing the grid. Predictive analytics also informs the strategic placement of energy storage technologies and microgrids, so decentralized resources are deployed where they will deliver the highest resiliency payback.

Additionally, machine learning supports a transition from reactive to proactive approaches to maintenance, significantly improving infrastructure longevity and dependability. Predictive maintenance software, based on real-time operating data and future usage projections, can detect future sites of failure before outages. By incorporating consumption projections and models of machinery breakdown due to wear and tear, utilities can design more savvy maintenance schedules that target high-risk assets while avoiding the inefficiencies of time-based patterns of routine maintenance. This reduces unplanned failures, streamlines labor deployment, and overall OpEx. When done large scale, these methods help enable an agile, responsive grid that can meet changing needs without the traditional burdens of large-area maintenance or emergency repair. Overall, infrastructure optimization using predictive forecasting not only protects the physical assets of the grid but also preserves its scalability, sustainability, and quality of service amid 21st-century energy challenges.

6. Discussion and Future Directions

Interpretability of Results

An important consideration for deploying machine learning models for the application to energy forecasting within smart grid management is that the outputs are both interpretable and highly reliable. For this research, significant attention was given to examining the contribution of individual attributes like time of day, temperature, weekday/weekend, and seasonality to overall model performance. Feature importance analysis provided evidence that temporal factors—in particular, time of day and weekday/weekend differences—consistently exerted significant influence across models, reflecting longstanding patterns of behavior within energy consumption. Sensitivity testing also showed that even small errors introduced within input features, such as misreadings of temperature, propagated throughout the model and heavily biased forecasted outputs, thus underscoring the need for high-quality, robust data inputs. Overall reliability of models, however, was high, regardless of multiple random train-test splits or predictive horizons. This highly transparent performance not only enhances confidence for deployment operations but also informs on the potential for interventions where anomalous consumption patterns are identified.

Nonetheless, seeking out interpretability always requires trade-offs between complexity and understandability. Whereas easier tree-based models like Random Forests enabled easy visualization of the path to a decision and ranking of feature importance, more advanced models like ensemble gradient boosting models created layers of abstraction that, though enhancing performance, made it more difficult to intuitively explain an individual prediction. Achieving this trade-off between predictive performance and explainability is an important consideration for incorporating machine learning within high-stakes applications like smart grids, where stakeholders not only need high-performing predictions but also actionable information and justifiable decisions. Future research can further leverage explainable artificial intelligence (XAI) methods like SHAP values, LIME, and attention mechanisms to bridge the gap so that even the most advanced models are capable of providing understandable output that informs sensible operating decisions and builds trust among grid operators, regulators, and end-users.

Limitations

Despite the encouraging results shown in this study, several limitations need to be recognized to give an objective overview of the feasibility of deploying machine learning models for predictive smart grid forecasting. A major obstacle faced was data quality: inconsistencies, missing values, and sporadic anomalies within collected energy consumption data necessitated heavy cleaning and preprocessing. Missing IoT sensor readings, communications failures, and human input errors occasionally generated misleading trends, marking the vulnerability of data pipelines within actual smart grid deployments. Additionally, the changing behavior of users, due to factors such as the adoption of new appliances, working patterns (e.g., remote working trends), and new demand-response participation, introduced non-stationarity to the dataset. These changes created major obstacles for models that relied heavily on historical data since they risked decreasing the performance of models upon deviation from future consumption patterns from those deployed previously.

One significant limitation was that of class imbalance, especially so for the task of predicting energy usage classes using classification models. High-load events, while important for grid stability, were comparatively infrequent compared to so-called 'normal' or low-load periods and created datasets skewed towards the majority classes. While methods including oversampling, under-sampling, and weighted loss functions were utilized to counter this imbalance, maintaining an ideal balance without compromising on the generalizability of the models was very difficult. Moreover, short-range forecasting further contributed to this imbalance by generating high instances of 'normal' periods to very low instances of 'critical' peaks. This imbalance not only created challenges for the models' accuracy but also raised issues related to the sensitivity of the resulting alarms or control actions triggered automatically. Overcoming these limitations will necessitate continued advances in data collection processes, creating models that can learn adaptively from changing behavioral baselines, and using more advanced methods to address the imbalanced datasets without adding bias or instability to the forecasting system.

Future Work

In the future, various opportunities exist for enhancing the functionality of machine learning-based energy forecasting for the smart grid. First, incorporating more contextual sources of data, including online meteorological information, IoT sensor streams (e.g., appliance usage patterns, occupancy levels), and local grid status monitors, holds the potential to add to the feature inventory available to models. These integrations would allow for subtler and contextually aware prediction, one that would reflect the effects of dynamic external factors such as sudden temperature shifts, extreme weather events, or sudden spikes in local energy usage. Second, the use of real-time feedback loops—where models repeatedly learn and adapt from outcomes and usage—would result in a half-autonomous forecasting system that stays robust and resilient against changing consumption patterns. Feedback functions would use online learning methods or reinforcement learning frameworks to iteratively enhance the accuracy and reliability of models, thereby offering a responsive and self-adjusting smart grid environment. An additional direction of future research of paramount importance is the more in-depth analysis of advanced deep learning techniques, especially developed for time-series forecasting challenges. Recurrent Neural Networks (RNNs), and more advanced variants thereof, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), hold significant potential for capturing long-range dependencies, complex seasonality, and sudden changes in energy consumption patterns.

While conventional models are limited to sequential data with no memory and thus need major feature engineering to learn both short-term fluctuations and long-term tendencies, LSTMs and GRUs learn from both short-term and long-term patterns, leveraging memory within the models. Researching hybrid models that combine convolutional layers for feature extraction and recurrent layers for temporal modeling, or even exploring transformer-based models renowned for their high ability to model long-range dependencies, can further advance predictive performance. All these innovations, accompanied by ongoing pressures towards explainability, good quality data, and real-time learning, will be essential to developing the next-generation intelligent, robust, and sustainable smart grid prediction systems.

7. Conclusion

The chief objective of this research project was to leverage the potential of machine learning algorithms to maximize the precision and flexibility of energy demand prediction within smart grid networks. The data used for this study consisted of high-resolution, time-stamped energy consumption data captured at 15-minute intervals for two years, including both residential and commercial usage patterns. Every record contained the precise timestamp of consumption, which made it possible to undertake fine-grained temporal analysis that captures strong hourly cycles, daily patterns, and seasonal variations that are representative of user behavior and climatic factors. The authors selected three different models for effective energy demand level classification in this study. Our models received a time-based train-test split for their dataset to guarantee their robustness. A complete set of performance metrics for assessing our classification models included Accuracy alongside Precision, Recall, and F1-Score, in addition to implementing a Confusion Matrix evaluation. Both KNN and SVM demonstrated a strong balance in precision against recall because their Accuracy and F1 Score bars overlap almost completely in the evaluation graph. The results indicate that KNN and SVM perform superior to Logistic Regression with very equivalent outcomes in this classification activity. One of the most direct and significant advantages of using machine learning-based energy demand forecasting for smart grids is improved operating efficiency through more intelligent scheduling of energy delivery. Furthermore, machine learning-driven operational efficiency makes an important contribution to cost savings throughout the energy value chain. By being able to forecast short-interval demand fluctuations, utilities are better positioned to more efficiently execute their procurement strategy, such as strategically selling and buying energy on wholesale markets. Load management is another essential function where machine learning-based forecasting greatly improves smart grid operations. Preplanning for grid expansion and maintenance is another strategic advantage that arises from the predictive potential of machine-learning-powered smart grids. In the future, various opportunities exist for enhancing the functionality of machine learning-based energy forecasting for the smart grid. First, incorporating more contextual sources of data, including online meteorological information, IoT sensor streams (e.g., appliance usage patterns, occupancy levels), and local grid status monitors, holds the potential to add to the feature inventory available to models.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Aderibigbe, A. O., Ani, E. C., Ohenhen, P. E., Ohalete, N. C., & Daraojimba, D. O. (2023). Enhancing energy efficiency with ai: a review of machine learning models in electricity demand forecasting. Engineering Science & Technology Journal, 4(6), 341-356.
[2] Albogamy, F. R., Hafeez, G., Khan, I., Khan, S., Alkhammash, H. I., Ali, F., & Rukh, G. (2021). Efficient energy optimization day-ahead energy forecasting in smart grid considering demand response and microgrids. Sustainability, 13(20), 11429.
[3] Akter, R., Nasiruddin, M., Anonna, F. R., Mohaimin, M. R., Nayeem, M. B., Ahmed, A., & Alam, S. (2023). Optimizing Online Sales Strategies in the USA Using Machine Learning: Insights from Consumer Behavior. *Journal of Business and Management Studies*, *5*(4).
[4] Anonna, F. R., Mohaimin, M. R., Ahmed, A., Nayeem, M. B., Akter, R., Alam, S., ... & Hossain, M. S. (2023). Machine Learning-Based Prediction of US CO2 Emissions: Developing Models for Forecasting and Sustainable Policy Formulation. *Journal of Environmental and Agricultural Studies*, *4*(3), 85-99.
[5] Biswal, B., Deb, S., Datta, S., Ustun, T. S., & Cali, U. (2024). Review on smart grid load forecasting for smart energy management using machine learning and deep learning techniques. *Energy Reports*, *12*, 3654-3670.
[6] Cebekhulu, E., Onumanyi, A. J., & Isaac, S. J. (2022). Performance analysis of machine learning algorithms for energy demand–supply prediction in smart grids. Sustainability, 14(5), Chou, J. S., & Truong, D. N. (2021). Multistep energy consumption forecasting by metaheuristic optimization of time-series analysis and machine learning. International Journal of Energy Research, 45(3), 4581-4612.2546.
[7] Chouksey, A., Shovon, M. S. S., Tannier, N. R., Bhowmik, P. K., Hossain, M., Rahman, M. S., ... & Hossain, M. S. (2023). Machine Learning-Based Risk Prediction Model for Loan Applications: Enhancing Decision-Making and Default Prevention. *Journal of Business and Management Studies*, *5*(6), 160-176.
[8] Dieudonné, N. T., Armel, T. K. F., Hermann, D. T., Vidal, A. K. C., & René, T. (2023). Optimization of Short-Term Forecast of Electric Power Demand in the city of Yaoundé-Cameroon by a hybrid model based on the combination of neural networks and econometric methods from a designed energy optimization algorithm. Technological Forecasting and Social Change, 187, 122212.
[9] Hasan, M. R. (2024). Revitalizing the electric grid: A machine learning paradigm for ensuring stability in the USA. *Journal of Computer Science and Technology Studies*, *6*(1), 141-154
[10] Hrnjica, B., & Mehr, A. D. (2020). Energy demand forecasting using deep learning. Smart cities performability, cognition, & security, 71-104.
[11] Javanmard, M. E., & Ghaderi, S. F. (2023). Energy demand forecasting in seven sectors by an optimization model based on machine learning algorithms. Sustainable Cities and Society, 95, 104623.
[12] Jui, A. H., Alam, S., Nasiruddin, M., Ahmed, A., Mohaimin, M. R., Rahman, M. K., ... & Akter, R. (2023). Understanding Negative Equity Trends in US Housing Markets: A Machine Learning Approach to Predictive Analysis. *Journal of Economics, Finance and Accounting Studies*, *5*(6), 99-120.

[13] Mostafa, N., Ramadan, H. S. M., & Elfarouk, O. (2022). Renewable energy management in smart grids by using big data analytics and machine learning. Machine Learning with Applications, 9, 100363.

[14] Nazir, A., Shaikh, A. K., Shah, A. S., & Khalil, A. (2023). Forecasting energy consumption demand of customers in smart grid using Temporal Fusion Transformer (TFT). Results in Engineering, 17, 100888.

[15] Oprea, S. V., & Bâra, A. (2019). Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions. IEEE Access, 7, 177874-177889.

[16] Rahman, M. S., Bhowmik, P. K., Hossain, B., Tannier, N. R., Amjad, M. H. H., Chouksey, A., & Hossain, M. (2023). Enhancing Fraud Detection Systems in the USA: A Machine Learning Approach to Identifying Anomalous Transactions. *Journal of Economics, Finance and Accounting Studies*, *5*(5), 145-160.

[17] Rana, M. S., Chouksey, A., Das, B. C., Reza, S. A., Chowdhury, M. S. R., Sizan, M. M. H., & Shawon, R. E. R. (2023). Evaluating the Effectiveness of Different Machine Learning Models in Predicting Customer Churn in the USA. *Journal of Business and Management Studies*, *5*(5), 267-281.

[18] Rawal, K., & Ahmad, A. (2022, March). A comparative analysis of supervised machine learning algorithms for electricity demand forecasting. In *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)* (pp. 1-6). IEEE.

[19] Saglam, M., Spataru, C., & Karaman, O. A. (2023). Forecasting electricity demand in Turkey using optimization and machine learning algorithms. Energies, 16(11), 4499.

[20] Solyali, D. (2020). A comparative analysis of machine learning approaches for short-/long-term electricity load forecasting in Cyprus. Sustainability, 12(9), 3612.

[21] Sizan, M. M. H., Das, B. C., Shawon, R. E. R., Rana, M. S., Al Montaser, M. A., Chouksey, A., & Pant, L. (2023). AI-Enhanced Stock Market Prediction: Evaluating Machine Learning Models for Financial Forecasting in the USA. *Journal of Business and Management Studies*, *5*(4), 152-166.

[22] Syed, D., Abu-Rub, H., Ghrayeb, A., Refaat, S. S., Houchati, M., Bouhali, O., & Banales, S. (2021). Deep learning-based short-term load forecasting approach in smart grid with clustering and consumption pattern recognition. *IEEE access*, *9*, 54992-55008.

[23] Wang, Z., Yang, C., & Bozkurt, A. (2024). An advanced framework for net electricity consumption prediction: Incorporating novel machine learning models and optimization algorithms. *Energy*, *296*, 131259.