Electricity Forecasting Software for Microgrid Energy Management System

Yuliia Parfenenko [1], Vira Shendryk [1], Yevhen Kholiavka [1], Oleh Miroshnichenko[1]

[1]*Sumy State University, 116 Kharkivska Street, Sumy, 40007, Ukraine*

Abstract

This article outlines the development of a forecasting software tailored for managing energy in microgrids. It discusses the creation of forecasting models using linear regression and random forest techniques. The software's purpose is to employ these models, handle input data, and store forecast results in a structured manner within a database. This meticulous organization enables efficient processing and timely decision-making in microgrid energy management.

The study primarily focuses on building a power demand forecasting subsystem utilizing random forest and linear regression models. It covers various stages, including project planning and execution, with special emphasis on designing a robust database structure for effective data management. Additionally, the implementation involves an activation module crucial for analyzing and forecasting power grid parameters, enhancing the forecasting process's efficiency.

Through rigorous analysis, the Random Forest Model emerges as the primary forecasting tool. The trained forecasting models are stored in S3 cloud storage. Following model selection, the Forecasting Software's architecture were designed, incorporating essential components for seamless microgrid energy management system operation.

The developed electricity forecasting software is used for activating pre-trained models, processing input data, and logging forecast results into the database. It ensures smooth functioning and facilitates the integration of forecasted data into decision-making processes within power grid management systems.

Keywords

Forecasting, electricity, random forest, linear regression, machine learning, energy management

1. Introduction

Hybrid energy networks are becoming a key element of the transition to a more sustainable and efficient energy future [1]. Hybrid systems combine various energy sources, such as renewable (solar, wind, hydropower, etc.) and traditional energy sources, forming complex networks that meet the needs of consumers. However, solving the problems associated with managing such complex systems requires advanced methods and tools, particularly software modules for decision support.

The management of the hybrid energy network is faced with a number of complex tasks, including forecasting production [2] and energy consumption, optimization of system operation modes, and ensuring the stability of the energy balance. One of the key aspects of management is the ability to accurately forecast changes in energy production and consumption based on available data.

The relevance of developing a software module to support decision-making in managing a hybrid power grid is obvious due to the increasing complexity and volume of processed data. Using forecasting models and analytical tools [3] can greatly facilitate the process of making optimal decisions, increasing energy systems' efficiency and stability.

One of the key aspects of managing such a system is accurate forecasting of energy consumption based on available data. In the context of this study, the software will be responsible

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia , Ukraine
yuliya_p@cs.sumdu.edu.ua (Yu. Parfenenko); v.shendryk@cs.sumdu.edu.ua (V. Shendryk); yevhen.kholyavka@itp.sumdu.edu.ua (Ye. Kholiavka);
0000-0003-4377-5132 (Yu. Parfenenko); 0000-0001-8325-3115 (V. Shendryk); 0000-0002-9841-7909 (Ye. Kholiavka); 0009-0005-6761-6269 (O. Miroshnichenko)

for launching a system for forecasting electricity consumption volumes, which is a critical element for effective hybrid power grid management.

Forecasting models for decision support systems (DSS) should cover various time intervals, as decisions may be made both operationally and during the planning process. This is important considering the need for forecasts over short, medium, and long time periods. One of the pertinent challenges is the storage and activation of necessary models within specified time frames to obtain relevant forecasts timely.

This study aims to develop software for electricity demand forecasting, which can be used as a subsystem in a hybrid energy grid management decision support system.

Tasks include analyzing management issues in hybrid energy, selecting and developing appropriate forecasting models, integrating them into the software module, creating the module's architecture, and testing its effectiveness in practice. This approach will contribute to improving the quality of management in hybrid energy and fostering the development of more resilient and cost-effective energy systems.

This study introduces a novel approach by developing specialized software for forecasting electricity demand within hybrid energy grids. By addressing the complexities of managing such systems through advanced forecasting models and decision support tools, it aims to enhance operational efficiency and system stability. The focus on covering various time intervals in forecasting and integrating these models into a cohesive software module represents a pioneering effort towards optimizing hybrid energy management.

2. Literature review

Before delving into assessing the effectiveness of forecasting models for electricity consumption and developing an activation module, it is essential to evaluate various models applicable to the forecasting process thoroughly. Article [2] delves into selecting an optimal model for forecasting hourly electricity consumption, emphasizing a significant dataset. Various models, including autoregressive and machine learning models, underwent examination. The study focused on comparing model performance, particularly emphasizing autoregressive models and deep learning models like LTSM.

Ultimately, the results indicate that the LTSM model displayed superior forecast performance for large datasets compared to the autoregressive SARIMAX model. Thus, it underscored the effectiveness of using deep learning models for forecasting electricity consumption when dealing with substantial data volumes.

Article [11] underscores the importance of selecting the appropriate method for forecasting microgrid load. It suggests that using random forest (RFR) with feature selection enhances forecast accuracy, considering crucial characteristics affecting electricity consumption and providing reliable load forecasts.

The development of accurate forecasting models necessitates carefully considering different methods, particularly those accounting for time dependence and data seasonality, such as random forest and linear regression. However, the suitability of each algorithm for specific time series forecasting tasks must be meticulously assessed.

The forecasting software development is an active area of research. The studies presented in this review demonstrate significant progress in this field. The appropriate architecture, framework, and deployment tool selection depend on your specific requirements and resources.

Article [5] presents an overview of the MLflow open platform aimed at managing the life cycle of machine learning. It provides tools and capabilities for tracking experiments and managing and deploying models.

The authors systematically analyze various aspects related to MLflow and consider their impact on machine learning processes. Special attention is paid to the possibilities of registration and tracking of parameters and results of experiments, which allows proper analysis and comparison of different approaches and models. In addition, model versioning tools are considered, which is important to ensure stability and reliability in development and deployment.

The paper also discusses aspects of model deployment, which is a key stage in the machine learning lifecycle. MLflow provides a user-friendly interface for deploying and managing models in a production environment, facilitating rapid implementation and use of trained models in practical applications.

Kubeflow is an open-source platform designed to create and deploy machine learning pipelines on Kubernetes. In this article, the authors thoroughly describe Kubeflow's architecture and illustrate how the platform enables developers to build and scale machine learning pipelines using the various tools available in its ecosystem.

Kubeflow architecture includes a number of components that are designed for different aspects of machine learning, such as data preparation, model training, validation of results, and deployment of models to a production environment. The article's authors examine these components and their interaction, explaining how they can be used to create complete machine-learning pipelines.

One of the main advantages of Kubeflow is its integration with Kubernetes, which allows you to automate the processes of deployment and management of computing resources. This makes Kubeflow a powerful tool for developing and implementing complex and large-scale machine-learning pipelines.

The article [7] presents TorchServe as a framework for deploying PyTorch models. The authors describe in detail the capabilities of this framework, including the scaling, monitoring, and model management features that make it an attractive solution for deploying models in a production environment.

TorchServe provides developers and researchers with a convenient tool to deploy and manage their PyTorch models in a production environment. The framework automates the deployment process, enabling efficient scaling, and provides real-time model monitoring and management.

One of the key strengths of TorchServe is its flexibility and ability to integrate with other tools and environments, making it a versatile solution for deploying models in various scenarios and requirements.

While TorchServe offers advantages, potential drawbacks include configuration complexity, limited model compatibility, resource costs, security concerns, and the need for ongoing support and updates.

Overall, the paper provides a detailed overview of the functionality and benefits of TorchServe and demonstrates its potential importance in the field of deploying machine learning models in production using the PyTorch framework.

Article [8] examines Amazon SageMaker Serverless Endpoint as a Serverless solution for deploying machine learning models. The authors describe in detail the advantages of using serverless approaches in the context of deploying models and provide practical examples of using Amazon SageMaker Serverless Endpoint.

The article examines the key functionalities of Amazon SageMaker Serverless Endpoint, including its ability to automatically scale as needed, monitor and manage models. The authors highlight the importance of these capabilities in ensuring efficient and reliable deployment of machine learning models in a real production environment.

In addition, the article provides practical examples of using Amazon SageMaker Serverless Endpoint, which helps readers better understand how this solution can be applied in specific machine-learning scenarios. This allows readers to evaluate the potential benefit and suitability of this solution for their own projects and tasks.

Despite its advantages, some drawbacks of Amazon SageMaker Serverless Endpoint may include limited configurability, AWS service costs, and restricted flexibility for certain complex model configurations.

The article [9] discusses TensorFlow Serving is a framework for deploying TensorFlow models to a production environment. The authors describe in detail the functionality of TensorFlow Serving, focusing on its ability to scale, monitor, and manage models.

The article provides a detailed overview of TensorFlow capabilities Serving in the context of deploying models in a production environment. Special attention is paid to scaling tools that allow easy handling of large workloads and high availability of models. In addition, the authors consider monitoring tools that help track the effectiveness and reliability of models in a production environment.

Additionally, the article also discusses the model management process, including versioning and updating. This is important to ensure stability and reliability in a production environment and to implement the latest model versions.

TensorFlow's functionality Serving, making it a valuable resource for those seeking to understand and implement efficient deployment models in a production environment.

In the context of deploying forecasting models, model activation plays a key role in ensuring their effectiveness and use in a production environment. Software activation modules ensure the fast and reliable launch of forecasting models, allowing them to be used in real-time to make important management decisions.

The work of forecasting software usually includes several main stages aimed at effectively using models for forecasting. The initial stage is the management of the activation process, where a request for a forecasting model is generated based on the user's needs. This request initiates a request to activate the model, which is then activated for forecasting.

Next, the forecasting model acquisition module executes a request to obtain the required model using the S3 API or other appropriate mechanisms [8]. After receiving the model, it is transferred to the activation module, where the activation process itself takes place, which includes internal procedures for modeling and calculating forecasts.

After activating the model, the received forecasting results are stored in the database. This step involves saving the forecast data in the appropriate database tables for later use. In addition, the recording results in the database can be subjected to additional analytical operations or used for reporting and analysis [9].

The absence of explicit drawbacks mentioned in the provided text makes it difficult to pinpoint specific limitations of TensorFlow Serving or the described forecasting model deployment process. However, potential challenges could include issues related to scalability, complexity of model management, and integration with other systems. Additionally, ensuring the reliability and accuracy of forecasting models in real-time production environments may pose technical difficulties. It's imperative to address these challenges to ensure the successful deployment and operation of the forecasting models, particularly in dynamic and high-demand scenarios. This may involve optimizing TensorFlow Serving for scalability, streamlining model management processes, enhancing integration capabilities, and implementing robust monitoring and validation mechanisms to maintain model accuracy and reliability in real-time settings.

Based on the analysis of literature sources and consideration of possible applications and frameworks for deploying forecasting models, it is noted that various tools can be utilized to address the direct task of electricity consumption forecasting. However, due to the necessity of further integrating forecasting results into a decision support system (DSS), it is expedient to develop specialized software rather than using what already exists. This will provide the integration of forecasting results into the decision support system and ensure high operational efficiency in addressing complex tasks related to energy resource management.

3. Forecasting Methodology

To collect and process data for electricity consumption forecasting, a technique to obtain a dataset from a private enterprise was used. The specified dataset contained indicators of changes in electricity consumption by the enterprise, recorded with a frequency of every hour.

The data collection method included obtaining information from the enterprise, which included the use of contractual agreements and the exchange of data between the parties. As an entity, the enterprise ensured the provision of data, including time indicators of electricity consumption.

After receiving the data, they were processed. This process involved identifying and correcting possible anomalies or errors in the data, as well as transforming and aggregating the data into a format suitable for further analysis and modeling. An analysis of the absence of missing values and their processing in case of detection was also carried out.

Subsequently, the data were subjected to a process of exploratory analysis to identify patterns, trends, seasonal variations, and other characteristics (Figure 1) that may affect electricity consumption. This analysis made it possible to better understand the internal relationship of the data and contributed to the selection of the most suitable forecasting models.

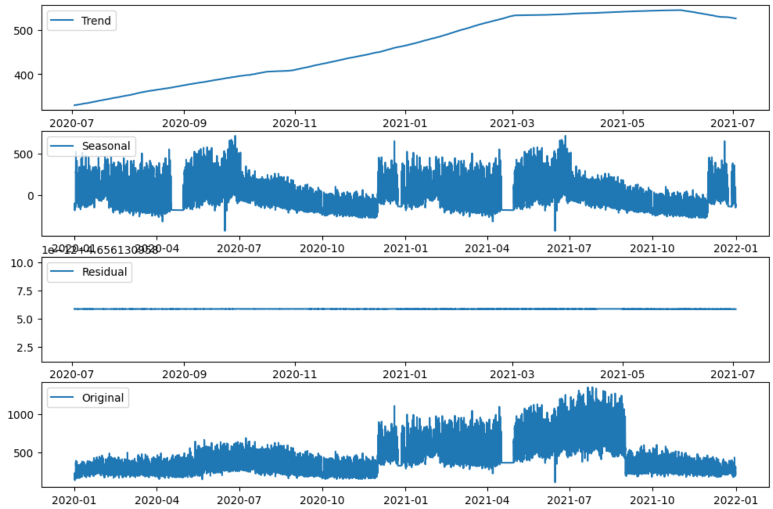


Figure 1: Time series decomposition

Observing the growing trend, seasonality, and character of the residual, which is similar to a straight line, will significantly contribute to understanding the data dynamics. The described data structure can occur in the context of various domains, including economic, technical, or social domains, where certain patterns and trends are present.

An increasing trend indicates a systematic increase in the values in the time series during the observation period. This may indicate economic development, increased demand, or other factors leading to gradual growth.

Seasonality reveals regular cyclical fluctuations in the data that repeat with the same or nearly the same amplitude and shape. However, the straight line-like nature of the residual may indicate a high level of randomness or uncertainty in the data that cannot be explained by trend or seasonality.

This scenario of data dynamics can be interpreted as the discovery of certain regularities in the development of the phenomenon [11], however, with significant uncertainty or randomness of influence in the specified time series. Such an analysis is important for understanding and forecasting the further development of data dynamics in the relevant context.

In the course of the study, a comparative analysis of the effectiveness of two different machine learning methods – linear regression and the random forest method for forecasting electricity consumption was carried out. The data used covered the period from November 2020 to May 2021 and contained information on various parameters such as year, month, week, day of the week, day, hour, and day length, as well as electricity consumption.

First, the data were prepared by dividing it into training and test sets. The training data were used to train the models, while the test data remained to evaluate their performance.

During the research, electricity demand was forecasted for different periods of time, which is determined by the "forecast" parameter. In this case, such forecast periods as 6, 12, 24, 72, and 168 hours were used.

Each of these periods has its own importance and application in the real world. For example, short-term forecasts, such as a 6-hour or 12-hour forecast, can be useful for real-time optimization of power generation or for managing household energy consumption. Long-term forecasts, such as a 24-hour, 72-hour, or 168-hour forecast, can be useful for planning and managing energy resources, such as scheduling scheduled repairs or optimizing the energy use of large industrial enterprises.

Based on the literature analysis, two main models were selected for regression analysis: linear regression and the random forest method. These models were chosen due to their widespread use, flexibility, and effectiveness in handling various data types.

Random forest is a powerful machine learning algorithm that uses an ensemble of decision trees to solve classification and regression problems [12]. One of its main advantages is the ability to automatically model complex nonlinear dependencies and consider the importance of features.

The formula for forecasting a time series using a random forest can be expressed as follows [12]:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} f_i(x), \quad (1)$$

where: $\hat{y}$ – the forecasting value, N - the number of trees of the random forest, $f_i(x)$ – the output of the model for the ith tree.

Linear regression is a classic analysis method that uses a linear function to forecast response values based on input features [13]. It is a simple and interpretable model that is often used for econometric and forecasting problems.

The formula for linear regression in the form of a time series forecast can be presented as [13]:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n, \quad (2)$$

where: $\hat{y}$- forecasting value, $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$– regression coefficients, $x_1, x_2, \cdots, x_n$– features of the model.

First, using the random forest method, 100 trees with a random state of 42 were generated, and then the model was trained on the training data. After that, the forecasts that were obtained were compared with the actual values of the test data. The same process was carried out for the linear regression model.

After obtaining the forecasts, both models were evaluated using various quality metrics such as mean squared error (MSE), mean absolute error (MAE), and others. These metrics made it possible to make an objective analysis of the effectiveness of each model in forecasting electricity consumption.

In measuring the effectiveness of forecasting models, such as Linear Regression and Random Forest, various metrics are used that determine the degree of their accuracy and adaptability to data [14]. The main metrics include root mean square error (Mean Squared Error, MSE), coefficient of determination (Coefficient of Determination, R- squared), average absolute error (Mean Absolute Error, MAE), and root mean square error (Root Mean Squared Error, RMSE).

The root mean square error [15] is one of the most common metrics for evaluating the accuracy of regression models. It is calculated as the average value of the squares of the difference between the observed and forecasting value. The smaller the MSE value, the better the model forecasts the data.

The coefficient of determination (R-squared) [15] measures the proportion of variation in responses explained by the model. It takes a value between 0 and 1, where 1 indicates a perfect model that fully explains the variation in the data. Using R-squared, you can compare different models and determine their ability to explain the data.

The mean absolute error measures the mean absolute value of all the differences between the observed and forecasting values. It allows you to get an idea of the average accuracy of the model, regardless of whether the forecasting values are exceeded or underestimated.

The root mean square error (RMSE) is an interpreted version of MSE, which is calculated as the square root of the root mean square error. It measures the average error in the same units as the original data, making it more interpretable.

These metrics help evaluate the performance and accuracy of Linear Regression and Random Forest models. The use of the most suitable metric depends on the specific context of the task and the requirements of the researcher [14, 15].

Based on the results obtained (Figure 2), it is possible to draw a conclusion about which model is better suited for forecasting electricity consumption in this context, which can be useful for further analysis and decision-making in the field of energy.

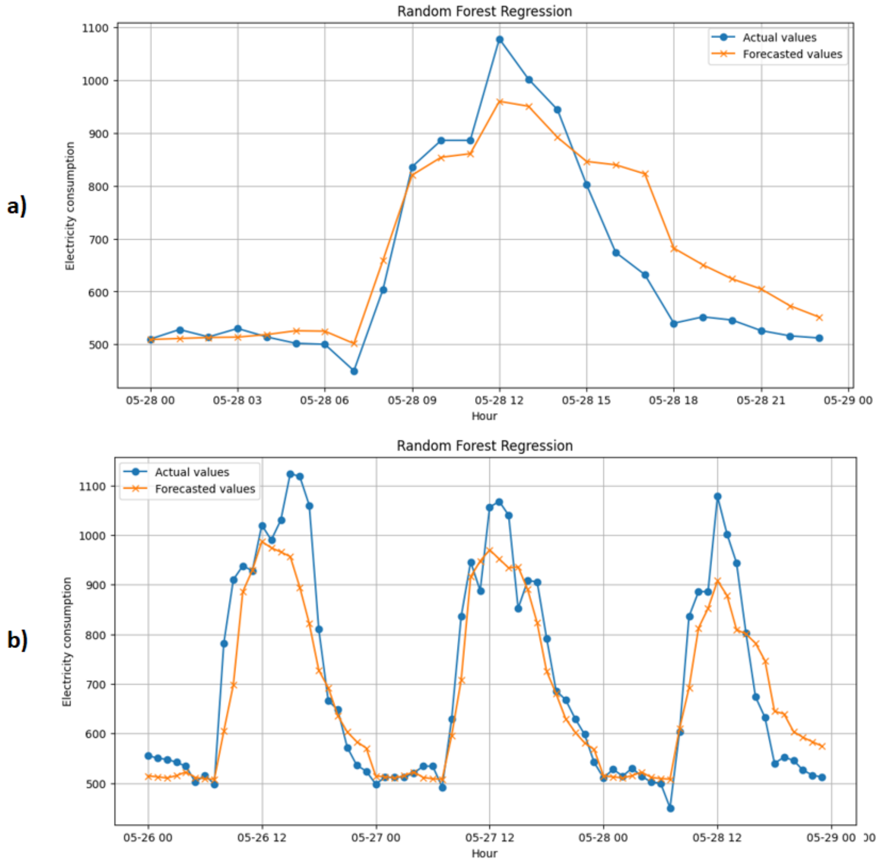


Figure 2 : a) forecasted and actual values for 24-hour-ahead forecasting using the RFR model; b) forecasted and actual values for forecasting 72 hours ahead using the RFR model

As a result of forecasting electricity consumption using random forest (RFR) and linear regression (LR) models, there is a difference in forecasting accuracy for different forecasting periods from 6 to 168 hours.

The random forest model shows low values of mean absolute error (MAE), root mean square error (RMSE), and mean percentage absolute error (MAPE) for short-term forecasts (6 and 12 h), indicating its good accuracy. However, when forecasting for longer periods (24, 72, and 168 hours), these metrics increase, indicating poorer performance of the model at longer time intervals.

Compared to the random forest model, linear regression exhibits higher values for all three metrics regardless of the forecast period. In addition, there is a tendency for errors to increase with the length of the forecast period, which may indicate the limited ability of linear regression to accurately forecast over a long period of time.

In general, the results (Table 1) indicate the superiority of the random forest model for short-term forecasting, while for long-term forecasting, it may be useful to use other models or approaches since both considered models show certain limitations in accuracy at long distances in time.

Table 1
The forecast results

Model	Metrics	6 hours	12 hours	24 hours	72 hours	168 hours
RFR	MAE	14.47	30,138	59.5	61,42	67,106
	RMSE	15,978	36,261	79.48	90,096	86,77
	MAPE	4,995	10.89	9.59	7.89	9.235
LR	MAE	84,689	164,235	131.62	173.04	177.69
	RMSE	85,225	222,036	193,419	242.64	245.41
	MAPE	18,180	20.58	16.78	20,233	21,211

In conclusion, the developed models tailored to various time intervals will be utilized to construct a subsystem for launching forecasting models within the decision support system (DSS), ensuring effective management of electricity consumption and facilitating informed decision-making processes.

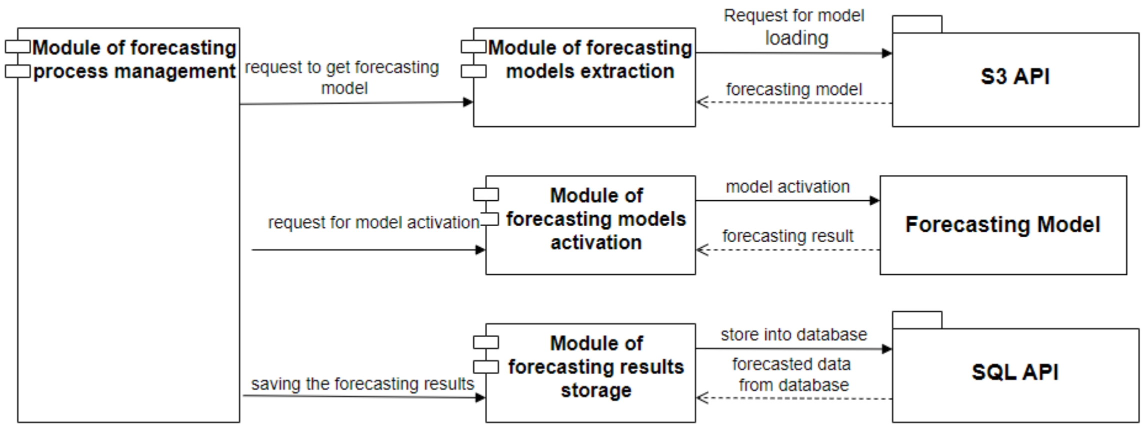
4. Forecasting software development

The forecasting software is designed to automatically run forecasting models to provide stable and accurate predictability at set time intervals, which is critical to ensuring continuity of electricity supply and efficient resource allocation.

This architecture is a complex system of management and application of forecasting models aimed at efficiently using resources and optimizing the forecasting process. In this system, there is interaction between several key modules, each of which performs a specific function within the forecasting process.

In general, this forecasting software architecture (Figure 4) is designed with the optimal use of resources in mind and provides an efficient management process and use of forecasting models. Its structured organization allows for reliable and fast implementation of forecasting with minimal loss of resources.

Figure 4 : Forecasting software architecture



The first module, namely the "Module of the forecasting process management", is responsible for coordinating and controlling the activation process of forecasting models. This module functions as an entry point for model activation requests, sending them to the forecasting model activation module.

The second "Module of forecasting models extraction" performs the task of getting the required forecasting models using the S3 API. It acts as an intermediary between the stored models and other modules of the system, providing access to the models using a defined protocol.

The "Module of forecasting models activation" is responsible for the activation process of the models obtained through the previous module. This module coordinates the initialization and loading of models for further use in the forecasting process.

The "Module of forecasting results storage" provides interaction of the system with the database, where forecasting results and other important data are stored. It writes and reads data from the database using the SQL API, ensuring reliable and efficient work with information.

The software was developed using the Python programming language due to its versatility and a rich arsenal of available libraries that facilitate the development process. Amazon S3 API ensures storage and efficient management of forecasting models, providing reliable storage and fast access to models. Using containerization via Docker provides a standardized environment for the module, while using Linux Cron allows you to automate the running process on a schedule. It actively interacts with the model repository hosted on Amazon S3 to obtain the required forecasting models.

After the forecasting models are activated and executed, the forecasting results or error information are written to the database. This process, which includes acquiring the models, running them, and storing the results or activation errors, is critical to ensuring the accuracy and reliability of the forecasting.

In the process of developing a system for activating forecasting models for managing a hybrid power grid, special attention was paid to the structuring and organization of the database. This database provides reliable and efficient storage of important data associated with each model run. After analyzing literary sources, it was decided to create two main tables in the database.

The first table, Model List, contains information about each moddwdel, including its S3 storage location, test data ID, and launch interval. The second table, Model Activation Results, is designed to track each model run and contains details of the execution process, including status, error descriptions, and results obtained.

5. Experiments

Before testing the developed software for the models, which were created in section 3 and stored in *.py files on an S3 Bucket, it is worth considering the process of automatic activation of models, which occurs through the configuration of tasks in the Linux Cron system.

Automatic activation is provided to run forecasting models, which is carried out by configuring tasks in the Linux Cron system. According to these tasks, the activation module runs at the specified time schedule, initiating a request to the model repository hosted on S3. At this stage, the necessary models are selected for further execution.

After receiving the appropriate models, the Module of forecasting models activation runs them to perform forecasting based on the incoming data. The module then processes the obtained results, including forecasted data and possible performance metrics, for further analysis.

The process of activating models includes not only starting and processing data but also registering these results in a database. Other modules of the system for analysis and development of optimal power grid management strategies subsequently use this database.

In case of errors detected during the activation of models, such errors are recorded for further analysis and possible optimization of the process in the future. This contributes to the continuous improvement of the management system and ensures its optimal functioning.

The obtained results (Figure 5, 6), stored in the database, serve as a basis for the operation of the subsequent modules of the system. This data is analyzed and used to make accurate and efficient decisions regarding power grid management. Thus, the activation module of forecasting models plays a key role in the operation and optimization of the power grid management system.

Id	ModelId	Dataset	Status	LastError	Results
15	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
16	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
17	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
18	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
19	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
20	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
21	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
22	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
23	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
24	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZp
25	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu
26	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu
27	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu
28	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu
29	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu
30	2	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu
31	1	NULL	COMPLETE	NULL	c3RyID0gciIiIuKCrATigKJEICAgICAgIMWSHWtlcmFzLnNyYy5zYXZpbmcu

Figure 5: Results of forecasting models activation

Based on the testing results, errors occurred when the model files were missing and when there was a lack of access to the necessary resources required for forecasting. However, in cases where the models were stored in the repository, they were all successfully launched, confirming the effectiveness of the automatic model activation process.

Begin Transaction | Text | Filter | Sort | Columns

id	model_id	Date	Forecast
1	2	2021-05-28 00:00:00	509,16
2	2	2021-05-28 01:00:00	511,08
3	2	2021-05-28 02:00:00	512,88
4	2	2021-05-28 03:00:00	513,68
5	2	2021-05-28 04:00:00	518,44
6	2	2021-05-28 05:00:00	525,7
7	2	2021-05-28 06:00:00	524,94
8	2	2021-05-28 07:00:00	501,54
9	2	2021-05-28 08:00:00	659,16
10	2	2021-05-28 09:00:00	820,26
11	2	2021-05-28 10:00:00	854
12	2	2021-05-28 11:00:00	860,72
13	2	2021-05-28 12:00:00	959,88
14	2	2021-05-28 13:00:00	950,54
15	2	2021-05-28 14:00:00	891,6
16	2	2021-05-28 15:00:00	846,06

Figure 6: The forecasting results stored in the database

The developed software is used as a subsystem in the decision support system of microgrid energy management. The forecast data stored in the database have timeframes for convenient usage and analysis, enabling timely and effective decision-making regarding electricity management. This facilitates informed actions in the field of electric power engineering.

Conclusion

As a result of the study the software for electricity consumption was developed. It is designed to activate forecasting models that have been previously trained and stored in S3 storage. The developed forecasting software, integrated as a forecasting subsystem within the decision support system of microgrid energy management, showcases exceptional accuracy and efficiency in predicting electricity demand. The comparison between random forest method and linear regression highlights the former's superiority in forecasting accuracy. Computational experiments conducted over various time intervals demonstrate the robustness of the developed models.

In conclusion, while the successful implementation of the forecasting software and rigorous testing of forecasting models are noteworthy achievements, it's imperative to elucidate the practical applications and potential scalability beyond the S3 storage framework. The developed subsystem stands as a critical component of the decision support system, enriching decision-making processes in electricity management and offering promising avenues for optimizing energy network operations and ensuring reliable power grid management. To extend its utility to broader domains, the software can be adapted to different sectors such as city-wide power networks, industries, or residential communities. Integration with real-time data from smart meters and IoT devices can enhance performance. Scalability, renewable energy integration, and decision support capabilities are key areas for further development, ensuring effective energy management and sustainability across diverse contexts.

References

[1] Bankruptcy Proceedings for an International Doctorate Contract within the Scope of the WEC4Ports Project: A Hybrid Wave Energy Converter for Ports. Notice (excerpt) no. 20317/2023. Official Gazette of Ukraine, Series II. October 24, 2023. URL: https://files.diariodarepublica.pt/2s/2023/10/206000000/0017800178.pdf (Accessed: March 18, 2024).

[2] Shendryk, V., et al. "Modeling Techniques of Electricity Consumption Forecasting." Presented at the 9th International Conference of the Indonesian Chemical Society (ICICS) 2021: Toward a Meaningful Society, Mataram, Indonesia, 2022. URL: https://doi.org/10.1063/5.0100123 (Accessed: March 18, 2024).

[3] Chen, Q., et al. "Monthly Electricity Consumption Forecasting." In Data Analytics in Power Markets. Singapore, 2021. Pages 43–69. URL: https://doi.org/10.1007/978-981-16-4975-2_4 (Accessed: March 18, 2024).
[4] Shendryk, V., et al. "Short-term Solar Power Generation Forecasting for Microgrid." Presented at the 2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, October 4–7, 2022. URL: https://doi.org/10.1109/saic57818.2022.9922982 (Accessed: March 18, 2024).
[5] Ventriglia, R. M., et al. "Applying Machine Learning to the Fuel Theft Problem in Pipelines." Journal of Pipeline Systems Engineering and Practice, vol. 14, no. 2, 2023. URL: https://doi.org/10.1061/jpsea2.pseng-1374 (Accessed: March 18, 2024).
[6] Ye, X., Liu, R., Yue, Z. "Structured Knowledge Base Q&A System Based on TorchServe Deployment." Journal of Physics: Conference Series, vol. 2078, no. 1, 2021, p. 012017. URL: https://doi.org/10.1088/1742-6596/2078/1/012017 (Accessed: March 18, 2024).
[7] "ECMC Rule-based Software Module for Defect Prediction using Support Vector Balanced Data." International Journal of Recent Technology and Engineering, vol. 8, no. 6, 2020, pp. 5219–5223. URL: https://doi.org/10.35940/ijrte.f1103.038620 (Accessed: March 18, 2024).
[8] Gadban, F., Kunkel, J. "Analyzing the Performance of the S3 Object Storage API for HPC Workloads." Applied Sciences, vol. 11, no. 18, 2021, p. 8540. URL: https://doi.org/10.3390/app11188540 (Accessed: March 18, 2024).
[9] Karunanidhi, V. "Understanding the TensorFlow Serving API: Deploying TensorFlow Models to a Web Application." Berkeley, CA, 2020. URL: https://doi.org/10.1007/978-1-4842-6699-1_7 (Accessed: March 18, 2024).
[10] Canela, M. Á., Alegre, I., Ibarra, A. "Trend and Seasonality." In Quantitative Methods for Management. Cham, 2019. Pages 93–101. URL: https://doi.org/10.1007/978-3-030-17554-2_10 (Accessed: March 18, 2024).
[11] Zheng, D., et al. "Short-term Renewable Generation and Load Forecasting in Microgrids." In Microgrid Protection and Control, 2021, pp. 57–96. URL: https://doi.org/10.1016/b978-0-12-821189-2.00005-x (Accessed: March 20, 2024).
[12] Mantas, C. J., et al. "A Comparison of Random Forest-Based Algorithms: Random Credal Random Forest vs. Oblique Random Forest." Soft Computing, vol. 23, no. 21, 2018, pp. 10739–10754. URL: https://doi.org/10.1007/s00500-018-3628-5 (Accessed: March 18, 2024).
[13] Fahrmeir, L., et al. "Generalized Linear Models." In Regression. Berlin, Heidelberg, 2021, pp. 283–342. URL: https://doi.org/10.1007/978-3-662-63882-8_5 (Accessed: March 18, 2024).
[14] Damanhuri, A., Mukarram, A. "Ideal Data to Determine Accurate Fajr Time." Jurnal Mathematics MANTIK, vol. 8, no. 1, 2022, pp. 28–35. URL: https://doi.org/10.15642/mantik.2022.8.1.28-35 (Accessed: March 18, 2024).
[15] Kuhlgatz, D. A., et al. "Development of Predictive Models for Boar Semen Quality." Theriogenology, vol. 134, 2019, pp. 129–140. URL: https://doi.org/10.1016/j.theriogenology.2019.05.024 (Accessed: March 18, 2024).
[16] Barrak, A. F., Petrillo, F., Jaafar, F. "Serverless Machine Learning: A Systematic Mapping Study." IEEE Access, 2022, p. 1. URL: https://doi.org/10.1109/access.2022.3206366 (Accessed: March 18, 2024).
[17] Parfenenko, Y. V., et al. "Comparison of Short-term Forecasting Methods of Electricity Consumption in Microgrids." In Radio Electronics, Computer Science, Control, 2023, Issue 1, p. 14. URL: https://doi.org/10.15588/1607-3274-2023-1-2 (Accessed: March 18, 2024).
[18] Karyeyen, A., Çetinkaya, N. "Ensuring Energy Balance for Sudden Demand Changes in Smart Grids." International Journal of Energy Studies, 2023. URL: https://doi.org/10.58559/ijes.1266296 (Accessed: March 18, 2024).
[19] Zaharia, M. "Designing Production-Friendly Machine Learning." Proceedings of the VLDB Endowment, vol. 14, no. 13, 2021, p. 3420. URL: https://doi.org/10.14778/3484224.3484241 (Accessed: March 18, 2024).
[20] Siegel, A. F., Wagner, M. R. "Chi-Squared Analysis." In Practical Business Statistics. 2022, pp. 531–547. URL: https://doi.org/10.1016/b978-0-12-820025-4.00017-8 (Accessed: March 18, 2024).