ENHANCED SMART GRID STABILITY PREDICTION USING HYBRID MACHINE LEARNING TECHNIQUES

A PROJECT REPORT

submitted by

CB.EN.U4EEE23108	**Bhagyashree.M**
CB.EN.U4EEE23126	**Ponabirami.A**
CB.EN.U4EEE23142	**Varshini.G.S**
CB.EN.U4EEE23155	**Harshita.V.P**

in partial fulfillment of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE- 641112

April 2025

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE CAMPUS

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**ENHANCED SMART GRID STABILITY PREDICTION USING HYBRID MACHINE LEARNING TECHNIQUES**" submitted by

CB.EN.U4EEE23108	**Bhagyashree.M**
CB.EN.U4EEE23126	**Ponabirami.A**
CB.EN.U4EEE23142	**Varshini.G.S**
CB.EN.U4EEE23155	**Harshita.V.P**

in partial fulfillment of the requirements for the subject **23EEE213 - INTRODUCTION TO MACHINE LEARNING** in **Bachelor of Technology** in **ELECTRICAL & ELECTRONICS ENGINEERING** is a bonafide record of the work carried out under my guidance and supervision at the Department of Electrical and Electronics Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore Campus.

Mr. Nakkeeran M
Assistant Professor (OC),
Department of Electrical and Electronics Engineering,
Amrita School of Engineering,
Coimbatore - 641112.

This project report was evaluated by us on (date)

INTERNAL EXAMINER EXTERNAL EXAMINER

Contents

List of Figures

List of Tables

Abstract

With increasing renewable energy integration all around the world, maintaining grid balance and preventing fluctuations to ensure optimized energy distribution is very crucial. A stable grid helps in reducing large-scale blackouts and enhances overall resilience. However, the existing Machine Learning (ML) models, despite offering high accuracy, lack interpretability, robustness, and real-time adaptability. The main focus of this project is to provide an efficient smart grid stability prediction model using hybrid ML techniques. The model aims to forecast problems that can occur in the grid when renewable energy is used and electricity production does not meet the consumer's needs. The study compares the use of a Stacking model with XGBoost, LightGBM, and CatBost with other existing ML techniques and provides a reliable and efficient prediction of the stability of smart grids. Multiple ML techniques like Adaptive Boosting with Random Forest as the base estimator, basic XGBoost with hyperparameter tuning, XGBoost with LightGBM, XGBoost with CatBoost, and many other models were employed. Among these, the Stacking model of XGBoost (with hyperparameter tuning), LightGBM, and CatBoost with 150 estimators achieved an effective accuracy of 96.35%. To further improve the model reliability, ensemble techniques were combined with hyperparameter tuning and the SMOTE-Tomek method for class balancing. This approach further improves the stability prediction and model efficiency compared to conventional ML models. Future scopes can include the real-time deployment of the model along with Deep Learning (DL) techniques that can improve the grid stability predictions. This study explains how machine learning techniques could improve the smart grid energy management efficiently.

Keywords: Electric power transmission networks; Forecasting; Learning algorithms; Machine Learning; Stability; Grid Stability; Power stability; Smart grid

Chapter 1

Introduction

1.1 Background

Over the past few years, the world has seen a significant transition from traditional power grids to smart grids, which has led to improvements in grid optimization and energy management. A smart grid is an advanced energy network that integrates digital communication, automation, and advanced control technologies into the conventional power system. This system can be used for real time monitoring as well as respond to dynamic electricity demand much faster compared to the traditional grids. These advantages of smart grid make it important for us to maintain its stability in this modern world.

1.1.1 Need for Smart Grid Stability

The major reason for the transition from traditional power grids to the smart grids, at present, is the integration of renewable energy sources (RES) such as solar, wind and hydroelectric power. Due to the decentralized power generation and dynamic load variations, maintaining the power grid stability has become a complex challenge. With high amounts of carbon emissions from fossil fuels all around the world, the use of renewable energy provides a cleaner and more sustainable alternative. According to Paraschiv & Paraschiv (2020), between 1960 and 2018, the European Union (EU) experienced a sharp increase in CO_2 emissions, with certain member states like Italy and Spain seeing a rise of over 200 Mt CO_2 due to fossil fuel combustion[1]. It is difficult for smart grids to balance fluctuations in the energy sources due to their voltage and frequency variations with the changing environment. Thus, integrating renewable energy sources in power grids leads to grid instability, voltage fluctuations, and frequency imbalances. The consumer demand pattern also could be seen to vary, which increases the grid complexity. Thus, maintaining smart grid stability and providing an efficient and stable electricity to the consumers have become more challenging. Ensuring a stable smart grid prevents large-scale blackouts, equipment failures an insufficiencies in power generation.

1.1.2 Challenges in management of grid stability

The complexity of modern power systems is increasing day by day, due to which grid stability faces various challenges. The below mentioned are a few of them:

Unpredictability of Renewable Energy: The renewable energy resources like solar, wind, and hydroelectric power are highly unpredictable and affect the power generation efficiency [2]. Solar

power is dependent on sunlight availability, and fluctuations can occur in case of rainy or cloudy conditions. Wind energy also can lead to unpredictable changes in energy output due to the varied wind speed at different times. Since these energy sources lack consistency, smart grids must be able to respond to dynamic energy demand effectively.

Decentralized Power Generation: Unlike traditional power grids which are centralized, smart grids integrate distributed energy resources (DERs), such as solar panels in the rooftops of residential buildings and battery storage systems used for energy backup. This decentralized approach makes it difficult to optimize grid stability [2].

1.2 Problem Statement

As mentioned in the earlier part of this paper, the modern power generation and consumption trend has created significant challenges in maintaining grid stability and providing stable electricity to the consumers. Since smart grids have a decentralized power distribution strategy, it is difficult to manage the demand-supply balancing. Existing statistical techniques often fail to adapt to the dynamic nature of energy generation and consumption patterns. Thus, implementing a Machine Learning algorithm to predict the instabilities in smart grid power generation can serve as a useful technique to manage energy demand. This paper presents an efficient approach to handle data imbalances and demonstrate improvement in stability prediction accuracy compared to the already existing basic models.

1.3 Objectives

This project employs a hybrid ensemble ML model to enhance the accuracy and efficiency of smart grid stability prediction. The primary goals of this project are:

Improve Prediction Accuracy – The methodology seeks to enhance the prediction accuracy of electrical grid stability through the application of several machine learning methods that are capable of processing large and high-dimensional data with much efficiency. The model analyses the available grid stability data, finds patterns, and notifies authorities in cases of imminent risk. The use of ensemble learning approaches will minimize mistakes and enhance the entire performance of the model. This model would be extremely likely to get calibrated in a way to enable dynamic distribution for smart grids and offer true predictions as far as grid stability is concerned.

Compare ML Algorithms – This paper also compares the performance of different machine learning models to determine the optimal way to use in grid stability prediction. Based on the following section of the paper, no machine learning models for predicting smart grid stability are available with the inclusion of utilizing multiple models and data imbalance. Therefore, the proposed solution employs ensemble learning techniques using hybrid ML models to decide on the best approach for obtaining enhanced accuracy and efficiency.

1.4 Research gaps

Existing research has explored various machine learning (ML) models, including Support Vector Machines (SVM), Random Forest, and XGBoost, to predict grid stability. While these approaches have shown promising results, several research gaps remain, limiting their effectiveness in real-world applications. The major drawback of existing studies is the use of single ML models like Support Vector Machines (SVM), Random Forest, XGBoost, or low-level models without using the benefits of ensemble learning. As a result, they have not provided the best accuracy and generalization for dynamic grid conditions. The following are the key research gaps and challenges in the existing studies:

Lack of Model Optimization: Many of the studies use default ML models without tuning the parameters, which limits their performance and provides lower accuracy. Poor hyperparameter tuning can lead to longer training times and inefficient computation [3-5].

Class Imbalance Issues: The major challenge in grid stability datasets is the imbalance between stable and unstable grid conditions. Generally, a large set of data points is observed for stable conditions, while unstable conditions are represented by a minimal proportion of data. Present studies have failed to use data balancing techniques, leading to lower prediction metrics and struggles with real-life fluctuations in power systems [5-6].

Lack of interpretability in Deep Learning Algorithms: Deep Learning (DL) models, like LSTMs and CNNs, function as black-box models. They do provide high accuracy; however, data scientists cannot adopt such techniques that do not offer insights into their decision-making process. [7-8]

Limited Real-Time Adaptability: Real-world smart grid data are dynamic and require a model that can adapt to varying voltage and frequency conditions in smart power grids. However, existing models are trained on static datasets and are thus less adaptable to real-time power fluctuations.[9]

By addressing all these research gaps, we can provide a more accurate, interpretable, and robust model for smart grid stability prediction.

1.5 Contributions

The proposed model uses the Stacking Classifier by integrating XGBoost, LightGBM, and CatBoost, which can ensure improvement in accuracy and generalization. When compared to the single-classifier methods, this stacking approach can ensure better decision-making and strengthen model performance. A Logistic Regression meta-model is used as the final estimator to improve grid optimization and efficiency. The model also uses hyperparameter tuning, which helps in selecting optimal conditions for better smart grid efficiency and improves model performance compared to the traditional methods of using default parameters. This can help in adapting to the dynamically varying grid conditions. One of the major challenges faced is the imbalance in the

dataset. Thus, this is addressed by using the SMOTE-Tomek sampling technique, which reduces model bias. The balancing technique ensures that the classifier does not favor the majority class and improves overall model performance[4-5]. Polynomial Feature Engineering is used, additionally, to enhance model learning. Here, the relationship between the features is enhanced, further boosting model accuracy. The stacking classifier also ensures boosting the decision-making ability of the model compared to the Deep Learning (DL) techniques used in existing papers. The DL models often function as black-box models and hence do not provide transparency in decision-making to data scientists[6-12]. The integration of ensemble learning, hyperparameter tuning, and class balancing has ensured that the proposed model can outperform the already existing ML approaches while maintaining efficiency. The key research gaps in the current studies have been addressed, and a more scalable and robust ML algorithm has been proposed in this paper.

1.6 Paper Organization

Chapter 2 of this paper provides a comprehensive review of works related to Smart Grid Stability Prediction. It primarily discusses previous research and its drawbacks compared to our study. The chapter also highlights the similarities and differences between the existing studies and the proposed work. Various gaps in current studies are identified, and justification for the necessity of the proposed approach is presented.

Chapter 3 details the proposed approach in the study. Flowcharts and pseudocodes are provided for an easier understanding of the methodology used. The dataset details are included, along with a detailed explanation of the features used. This section explains the overall approach of the ML model, which has achieved better accuracy compared to existing models.

Chapter 4 describes the methodology used in the study. Detailed information about processes like data cleaning, data preprocessing, and machine learning models employed is provided. The algorithms for the various models tested are also included. A step-by-step implementation of the algorithm is discussed in detail, along with the evaluation metrics used to measure and validate the results.

Chapter 5 explains the experimental setup, tools, and techniques used in the study. Data preprocessing details and train-test split information are also provided. Around six algorithms tested during the study are presented, and the obtained results are displayed in the form of tables and graphs. A detailed analysis of the results is interpreted, and the performance of the proposed model is compared with existing techniques.

Chapter 6 provides deeper insights into the results and discusses the practical and theoretical implications of the study. Various limitations of the study are also addressed in this chapter.

The final chapter (**Chapter 7**) explains the major findings of the study and its contributions. It highlights the key takeaways and future scopes in the current work. This chapter also outlines challenges that future researchers might face.

Chapter 2

Literature Review

2.1 Existing Works

This section explores recent research on the prediction of smart grid stability using ML models, highlighting their limitations, advantages, and possibilities for further development.

Our base study [13], conducted by Hachefi Fatima Ezzahra et al. (2022), uses the Electrical Grid Stability Simulated Dataset provided by the Karlsruhe Institute of Technology. It compares the performances of different Machine Learning models like the Support Vector Machine (SVM), Random Forest, Extra Trees, AdaBoost, and XGBoost, which were used to classify the grid conditions as stable or unstable. The maximum accuracy of 95.23% was obtained using the XGBoost model with f_classif feature selection. The study majorly explains how machine learning models can help in improving smart grid stability, thus preventing large-scale blackouts and increase reliability in energy supply. It also explains the necessity for real-time flexibility of power systems with the growing integration of renewable energy sources. By understanding the limitations of this study, our proposed approach integrates a Stacking Classifier that combines XGBoost, LightGBM, and CatBoost to enhance prediction accuracy and generalization. SMOTE-Tomek technique, and hyperparameter tuning are used in balancing the data and get the optimal values for model parameters. This hybrid stacking model has better decision-making capabilities and has an improved performance compared to the individual models, that were discussed in our base paper.

Support Vector Machine (SVM) algorithm is implemented in the study by Singh et al. (2024) [14] for improving the efficiency of smart grids. This paper majorly explains the importance of integration of machine learning techniques in energy management and power forecasting. The algorithm improves the grids' stability immensely, improves energy generation scheduling, and reduces reliance on external power sources. The study has referred to an increase of 10% in supply-demand ratio, a drop of 15% in the peak demand of load, and an increase of 12% in the adoption of renewable power sources in recent years. It also demonstrates the reduction in overall operating costs to about 8.4%, emphasizing the importance of predictive analysis in energy management. The study aims to explain how SVM remains a robust ML technique, when compared to the traditional methods like Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), and Reinforcement Learning (RL). SVM has the ability to handle non-linear relationships and outliers effectively. In contrast to the methodology proposed in Singh et al. (2024), our model integrates a more comprehensive machine learning pipeline that improves forecasting accuracy and energy

management efficiency. The proposed model in this paper ensures to balance the data with SMOTE technique, and the stacking classifier with hyperparameter tuning ensures better accuracy. This model promises to improve energy forecasting efficiency, making it superior for real-world energy management.

The study conducted by Yerrolla and Ramesh (2023) [15] presents several ML techniques to identify the most efficient technique to make the inference about the grid stability from an oscillating dataset. Random Forest, SVM, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Artificial Neural Networks (ANN-MLP), and Naïve Bayes were evaluated in the work. Additionally, their proposed method also explores Partial Least Squares (PLS) regression, neural networks, and gradient boosting models. Out of all these, Random Forest provided a maximum accuracy rate of 93.5% and an F1-score of 0.948. The study attributes the said enhanced performance to ensemble learning, good feature choice, low bias-variance tradeoff, and the ability of Random Forest to work with unbalanced datasets. All these advantages encouraged our present research to explore an even more advanced ensemble learning approach—a Stacking Classifier based on XGBoost, LightGBM, and CatBoost—to further enhance model accuracy and robustness. However, their model does not account for class imbalances in the dataset. The minority over-sampling technique used in our work would help in unbiased predictions. The ensemble learning methods used in our work have also helped in providing better accuracy and good efficiency.

The paper by Alimi et al. (2020) [16] provides a comparison between some of the ML techniques such as Artificial Neural Networks (ANN), SVM, Decision Trees (DT), and Reinforcement Learning (RL), which is based on their application towards power system security and stability. The paper focuses on power quality disturbances, transient stability assessment (TSA), and voltage stability assessment (VSA). The paper emphasizes the main issues such as data availability and computational complexity, using the argument of the requirement for more efficient and scalable ML models. Some limitations of the model in the paper by Alimi et al. include methodologies without addressing class imbalance, hyperparameter tuning, and ensemble learning for better predictive accuracy. Compared to this, our study promises to enhance the performance metrics and thus is a more robust approach. While previous work has laid a strong foundation in ML-based power system stability analysis, our methodology provides a more sophisticated and effective framework.

The study by Asiri et al. (2024) [17] introduces a hybrid deep learning approach for short-term load forecasting in smart grids. Their methodology is termed as LFS-HDLBWO technique and it employs a combination of Convolutional Bidirectional Long Short-Term Memory with an Autoencoder (CBLSTM-AE) and the Beluga Whale Optimization (BWO) algorithm for hyperparameter tuning. The study also uses feature extraction and hyperparameter tuning. So, there is an improved accuracy in their paper. However, even if their model promises higher performance metrics, they use Deep Learning approaches, which require substantial computational power and may lack interpretability. This apasiriproach has minimized error boundaries in load demand forecasting, which is proof of the effectiveness of hybrid deep learning for application in smart

grids. It is this paper that motivated us to use hybrid models for our study. As mentioned in the later part of this paper, our proposed model aims to use a stacking classifier that would help in improvising the predictive performance by integrating multiple models. SMOTE technique and hyperparameter tuning further improve the metrics.

Arzamasov et al. (2020) [18] contribution lies in the description of grid stability in terms of compact models with data-driven approaches. This study examines the Decentral Smart Grid Control (DSGC) system with demand response under price electrification with grid frequency scenario. Through simulation, the study examines behavior in the system with different conditions of input to a decision tree model that the researchers employ in coming up with simplifiable yet understandable stability-deciding rules. Although research offers a fresh paradigm for dynamic study of DSGC, there are many limitations to research. Being reliant on decision trees, as useful as it is for interpretability, may not be as effective at capturing complicated nonlinear relationships as ensemble learning algorithms might have been. Furthermore, the research does not account for possible class imbalance in the data set, which can cause bias in stability prediction. Besides, the simulation-based approach, as useful as it is, is not grounded on actual experiments and, therefore, its applicability to real grid stability monitoring is questionable. By incorporating ensemble learning, data balancing, and hyperparameter tuning, our method shows improved accuracy and stability over the decision tree-based method in this current research. Further, the capacity of our model to address intricate real-world grid stability problems makes it more relevant.

Abu Al-Haija et al. (2021) [19] have proposed a highly efficient machine learning model for the prediction of smart grid stability. It could be noted that the paper uses seven different machine learning models, which includes Optimizable Support Vector Machine (SVM), Decision Trees (DTC), Logistic Regression (LGC), Naïve Bayes (NBC), k-Nearest Neighbour (KNN), Linear Discriminant Classifier (LDC), and Ensemble Boosted Classifier (EBC). Unlike other studies, which had only used accuracy as the metrics to evaluate the predictive performance of the grid, this paper by Abu Al-Haija et al. uses several performance indicators like the accuracy, precision, and recall. The research, however, lacks ensemble learning methods for enhancing robustness of predictions. Additionally, hyperparameter tuning is limited to some models with fewer chances for optimal settings to be achieved by different machine learning platforms. These limitations are addressed and an approach with a better performance is proposed in our study.

“Smart Grid Stability Prediction: A Comparative Analysis of Machine Learning Models” [20] presents a comprehensive approach for smart grid prediction. This paper has used machine learning techniques like Random Forest, XGBoost, Support Vector Machines, Logistic Regression, and Artificial Neural Networks. All the models are evaluated using accuracy, precision, F1 score, recall, and AUC-ROC curve. This paper also highlights the importance of feature selection and data preprocessing, and it highlights ANN as the best in predicting the smart grid stability, as it effectively handles the non-linear relationship among the data. This uses individual models, but our paper includes an ensemble model and stacking classifier, which helps in improving the overall accuracy. Individual models might have their own drawbacks, which can be resolved using a

stacking classifier, as proposed in our study. Both projects aim to enhance grid stability, but our method could improve the overall accuracy and enhance the prediction. Additionally, our study includes SMOTE (Synthetic Minority Over-Sampling Technique), which helps in balancing the imbalanced dataset. When a dataset is not balanced, it might lead to a bias towards the majority class, so SMOTE creates some synthetic data points in the minority class, thereby maintaining a balance between both the majority and minority classes. By this method, the model becomes robust towards bias and error.

The paper titled "On Data-Driven Modeling and Control in Modern Power Grid Stability: Survey and Perspective" [21] provides insights on handling non-linear data, handling uncertainties, and time-varying conditions due to renewable energy. This paper explores a variety of machine learning techniques to find an approach for smart grid prediction. This paper focuses on supervised learning, unsupervised learning, and reinforcement learning. It uses PL and learning-based methods to optimize power grid stability. However, this paper is about optimizing the parameters that influence power grid stability. It uses special and smart meters to record the dataset effectively and in a balanced way. Whereas our paper includes SMOTE to balance the imbalanced dataset and also involves techniques like ensemble learning to improve and enhance the overall accuracy.

2.2 Gap Analysis

Various researchers have previously evaluated various ML models, such as Support Vector Machines (SVM), Random Forest, and XGBoost, for predicting grid stability. Despite these methods proving to be very promising, there are a few limitations associated with them that disqualify their practical implementation. Most of the work has been on one ML algorithm like SVM, Random Forest, or XGBoost that, although robust to a certain degree, may not necessarily provide the best accuracy and generalization. One of the greatest drawbacks of such algorithms is that they include default ML model hyperparameters that result in inefficient performance and increased training time. Current research also does not provide a solution to the inherent class imbalance of grid stability data where instabilities within the grid will generally be larger than stable ones. Class imbalance can produce biased estimates and, therefore, undermine the reliability of the stability predictions. Interpretability is also a deficiency in current research that comes with high-accuracy models. Though deep learning-based models have improved the accuracy of predictions, they are black-box models and it is difficult for power system engineers to comprehend and rely on their decision-making mechanism. These research constraints must be overcome in developing a more accurate, interpretable, and resilient smart grid stability prediction model.

2.3 Justification

Old models of ML predictive frameworks of smart grid stability are of no use in many ways. SVM, Decision Trees, and Random Forest models are not useful and adaptive to learn as it's a need in sophisticated power systems. Random Forest and Decision Trees are based on bagging

(bootstrapping samples) where a lot of individual trees are trained and aggregated results come out. Models are not learnable from earlier mistakes. SVM selects the optimal hyperplane for classification but does not seek to maximize the misclassified points[7]. Boosting-based models (XGBoost, LightGBM, and CatBoost) use gradient boosting, where the next tree is learned based on the residual errors of the existing trees, leading to much enhanced performance and accuracy[10]. Stacking XGBoost, LightGBM, and CatBoost under a Stacking Classifier gives us high training accuracy and testing accuracy, improved generalization, and improved prediction[11]. Feature learning, ensemble approach, and real-time adaptability combined is a robust solution for the prediction of smart grid stability.

Chapter 3

Proposed Approach

3.1 Process Flow Diagram

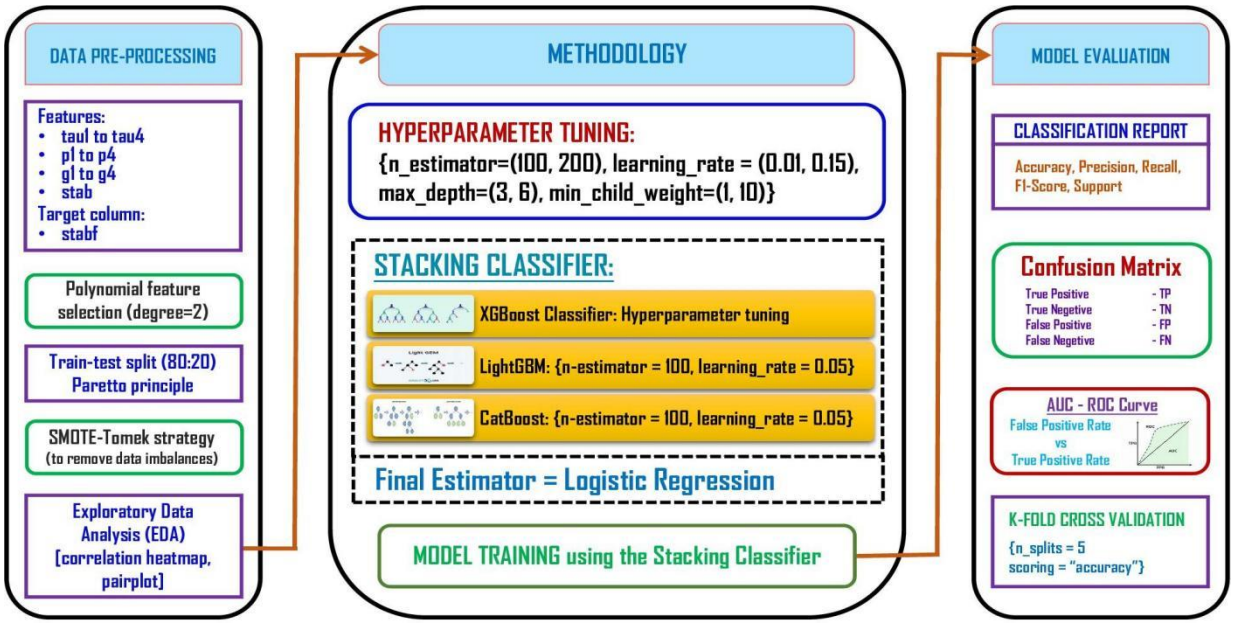


Figure 3.1: Process Flow Diagram of the proposed model

3.2 Dataset Details

The dataset [22] consists of simulated data of power systems and smart grids. It has been used to forecast the stability of the smart grid, and through this model, we can enhance the stability of the grid. The data measurement and collection are performed by using smart energy meters and smart power meters. It is based on simulations done on a 4-node architecture: one main producer providing electricity to three consumers. It has 10,000 samples and 14 attributes. Table 1 contains the details of the dataset and its description, which includes:

The first four columns named tau1, tau2, tau3, and tau4 are about the reaction time of the consumers and the producer. tau1 is the reaction time of the producer, while the other three columns give us the reaction time of the consumers. A longer reaction time from the producer side means it takes more time to respond to changes in demand. Also, if the consumers adjust their power consumption slowly, they may not effectively respond to supply variations, which may lead to power imbalances. Thus, longer reaction times increase the risk of instability, especially in

dynamic energy markets. Reactive power is a component of electricity that does not perform any useful work. Reactive power plays a major role in grid stability, and when there is any insufficiency in the reactive power, it may lead to voltage fluctuations or blackouts. The next three columns are about the nominal power generated and power absorbed. Nominal power is the rated power capacity of equipment. A stable grid should have balanced nominal power with load demand. The next four columns are about the price elasticity coefficient, which measures the change in demand for a product as a result of a change in price. So, whenever the price increases, the usage of the grid might get reduced, which might reduce the stress on the grid. g1 represents the price elasticity coefficient for the producer, while g2, g3, and g4 refer to the price elasticity coefficient of the three consumers. Higher elasticity means consumers respond to price changes significantly. Thus, moderate elasticity can promise a balance between supply and demand in energy and reduce instabilities.

Table 3.1: Dataset features and description

FEATURE COLUMN	DESCRIPTION
tau1-tau4	Reaction time within the network. tau1:reaction time of the producer. tau2-tau4:reaction time of three consumers.
p1-p4	Positive nominal power generated or negative nominal power absorbed. p1=p2+p3+p4.
g1-g4	Price elasticity coefficients g1:producer g2-g4:consumer
stab	Grid stability measure
Stabf (target variable)	Stable or unstable

The column named stab represents the largest real part of the root of the characteristic differential equation, which acts as the stability measure of the grid. The target column is stabf, which indicates whether the grid is stable or not. The target column is stabf, which indicates whether the grid is stable or not. Here in the target column it is given as stable or unstable so we need to convert it to numerical column before creating a model.With the help of exploratory data analysis we observe how each parameter contributes to the stability of the smart grid.

3.3 Approach towards dataset

Our process starts with data preprocessing, which includes removing the missing values and converting the target column to numerical value. Polynomial feature engineering is used to improve the model's ability to capture the non-linear relation,which improves the prediction accuracy even in complex grid conditions. As the dataset is imbalanced,balancing techniques like SMOTE-Tomek resampling is used to check whether an equal spread of data is there for stable and unstable grid conditions. This step is important in predicting grid stability, as it prevents the model from being falsely biased towards the majority class, this improves the overall prediction performance. For implementing the model, we have used a Stacking Classifier that use XGBoost, CatBoost, and LightGBM. By stacking these models, we can predict the data effectively, because the data patterns

related to smart grid are sometimes complex and highly variable due to fluctuation in renewable energy sources, and varying load conditions. We use Optuna hyper parameter tuning function, which adjusts the parameters automatically to get the perfect parameters for tuning. Finally, the performance of the model is evaluated using evaluation metrics which includes accuracy, precision, recall, F1-score, and confusion matrix. These metrics are important in the smart grid domain, where false positives and false negatives can lead to operational errors, higher costs, or even grid failures. These techniques are used to determine the stability of the smart grid and its efficiency.

Chapter 4
Methodology

4.1 Techniques

A stack classifier is a machine learning technique which combines multiple models to create an accurate model [10-12]. Here the base learners are CatBoost, LightGBM, XGBoost and logistic regression is used as the final meta learner. XGBoost focuses on both speed and performance. It is used for handling overfitting, as it has its own inbuilt regularisation. CatBoost works efficiently for categorical features without any pre-processing work. LightGBM is used and it works well for large datasets and it handles it effectively. And finally a classifier is used as a final meta learner which means the predictions of all the base models are combined and used as input features for the final classifier which is Logistic regression.
And our methodology includes:

Polynomial Feature Engineering is a technique which is used to enhance the model's ability to capture non-linear relations in the dataset. Here in terms of smart grid, the relationship between voltage and power demand are not linear, so by this method we can improve the model performance. It involves declaring new features by raising existing features to n power (n=1,2,3..) and combining with others. This captures more complex relationships. Variables like power consumption and reactive power interact in non-linear ways, so polynomial feature engineering with degree=2 contributes and helps the model to capture the non-linear relationship in our dataset.

SMOTE-Tomek Resampling is a hybrid resampling technique which is used for balancing the imbalance dataset. It is most commonly used in binary classification. Whenever the dataset is imbalanced, it may lead to biasing towards the majority class. Class imbalance happens when the number of data in one class is significantly more than another class. Whenever a model is trained on an imbalanced dataset, it may tend to get biased towards the majority class, which may lead to poor performance of the model. SMOTE, i.e., Synthetic Minority Over-Sampling Technique, addresses the imbalance by generating synthetic data points for the minority class rather than duplicating existing data. It works by creating synthetic data points by interpolating the minority class samples. SMOTE selects the data point from the minority class and randomly chooses one of the nearest neighbour from the same class. A new synthetic point is created by interpolating the feature value between the two points. Tomek links are used to clean the data by removing noisy examples from the majority class. By using SMOTE-Tomek, it is ensured that the data points related to stable and unstable grid conditions are adequately represented, allowing the model to learn the datasets equally.

Ensemble Learning -

CatBoost is a gradient-boosting algorithm used to handle categorical features in datasets. Additionally, CatBoost can process non-numeric data (such as labels or categories) without manual intervention with conversion into numerical form. This reduces preprocessing steps and preserves data quality. The process starts with building symmetric decision trees, where each level of the tree grows evenly. This structure allows the model to make faster and more accurate predictions. It also uses a unique technique called ordered boosting, which helps prevent overfitting. Moreover, CatBoost performs well with imbalanced datasets, where certain classes appear much more frequently than others. Hence, this is useful in smart grid applications, where detecting rare faults or instabilities is important.

LightGBM (Light Gradient Boosting Machine) is a high-speed, high-performance boosting algorithm designed to work with large datasets. It uses a leaf-wise tree growth strategy, which means it expands the branch that results in the highest improvement in model accuracy. LightGBM uses leaf-wise tree growth, which always splits the leaf with the highest potential to reduce error. This helps the model focus on the most informative parts of the data, enabling faster and more accurate learning compared to traditional level-wise methods. To further improve speed , LightGBM uses a histogram-based technique, where continuous data values are grouped into bins. It also supports parallel training and GPU acceleration, making it ideal for real-time applications.LightGBM is suitable for smart grid systems because it can deal with vast datasets that are generated by sensors and meters. It helps in tasks such as energy demand forecasting, usage pattern analysis, and anomaly detection with high speed and accuracy.This real time adaptabiliy adds value to smart grid stability.

Hyperparameter Tuning is the process of selecting the optimal set of parameters for the model to improve its performance. Tuning of hyperparameters makes the model accurate and allows it to perform well for unseen data[11]. Hyperparameters are external settings that must be tuned either externally or with the help of an algorithm. Some of the hyperparameters include learning rate, number of trees, max_depth, and so on. Proper hyperparameter tuning helps the model to capture the data effectively and avoid overfitting and underfitting. Instead of training the models individually, one can stack XGBoost, LightGBM, and Logistic Regression in a stacking classifier such that model vulnerabilities do not have to be tackled individually, leading to better overall performance.

4.1.1 Algorithms

ALGORITHM 1: Pseudocode for AdaBoost with Random Forest Base Estimator

INPUT : Preprocessed Training Data (X_train, y_train), Test Data (X_test, y_test),
Base Estimator: Random Forest (),
AdaBoost Parameters (n_estimators=150, learning_rate=0.01)

OUTPUT : Trained AdaBoost Model, Predictions (y_pred), Evaluation Metrics (Accuracy, F1-Score)

BEGIN

1. Initialize base_estimator ← RandomForestClassifier()
2. Initialize adaboost_model ← AdaBoostClassifier(
   base_estimator=base_estimator,
   n_estimators=150,
   learning_rate=0.01
)
3. Train model: adaboost_model.fit(X_train, y_train)
4. Predictions: y_pred ← adaboost_model.predict(X_test)
5. Evaluate: Calculate Accuracy and F1-Score using (y_test, y_pred)

END

ALGORITHM 2: Pseudocode for XGBoost with Hyperparameter Tuning

INPUT : Preprocessed Training Data (X_train, y_train), Test Data (X_test, y_test),
Hyperparameter Grid [n_estimators=(10, 150), max_depth=(3, 10), learning_rate=(0.01, 0.2)]

OUTPUT : Best XGBoost Model, Optimized Hyperparameters, Predictions (y_pred),
Evaluation Metrics (Accuracy, F1-Score)

BEGIN

1. Initialize model ← XGBClassifier(**params, random_state=42)
2. Define param_grid ← {
   'n_estimators': (10, 150),
   'max_depth': (3, 10),
   'learning_rate': (0.01, 0.2)
   }
3. Initialize Optuna ← optuna.create_study(direction='maximize')
   study.optimize(objective, n_trials=30)
4. Get best_xgb ← XGBClassifier(**study.best_params, random_state=42)
5. Predictions: y_pred ← best_xgb.predict(X_test)

6. Evaluate: Calculate Accuracy and F1-Score using (y_test, y_pred)

END

ALGORITHM 3: Pseudocode for CatBoost

INPUT : Preprocessed Training Data (X_train, y_train), Test Data (X_test, y_test), CatBoost Parameters (n_estimators=100, learning_rate=0.05, verbose=0)

OUTPUT : Trained CatBoost Model, Predictions (y_pred), Evaluation Metrics (Accuracy, F1-Score)

BEGIN

1. Initialize catboost_model ← CatBoostClassifier(
   n_estimators=100,
   learning_rate=0.05,
   verbose=0
)
2. Train model: catboost_model.fit(X_train, y_train)
3. Predictions: y_pred ← catboost_model.predict(X_test)
4. Evaluate: Calculate Accuracy and F1-Score using (y_test, y_pred)

END

ALGORITHM 4: Pseudocode for LightGBM

INPUT : Preprocessed Training Data (X_train, y_train), Test Data (X_test, y_test), LightGBM Parameters (n_estimators=100, learning_rate=0.05)

OUTPUT : Trained LightGBM Model, Predictions (y_pred), Evaluation Metrics (Accuracy, F1-Score)

BEGIN

1. Initialize lgbm_model ← LGBMClassifier(
   n_estimators=100,
   learning_rate=0.05,
)
2. Train model: lgbm_model.fit(X_train, y_train)
3. Predictions: y_pred ← lgbm_model.predict(X_test)
4. Evaluate: Calculate Accuracy and F1-Score using (y_test, y_pred)

END

ALGORITHM 5: Pseudocode for final Stacking Classifier Model

START

1. **Load and Clean the Dataset**
2. **Define Features and Target**
   Feature columns are selected:
   ["tau1", "tau2", "tau3", "tau4", "p1", "p2", "p3", "p4", "g1", "g2", "g3", "g4"]
   target column: "stabf"
3. **Data Pre-processing (EDA and PolynomialFeatures)**
4. **Train-Test split (80-Train and 20-Test)**
5. **SMOTETomek to resample**
6. **Define Hyperparameter Optimization Function -**
   OPTUNA - n_estimators=(100, 200),
   max_depth=(3, 6),
   learning_rate(0.01, 0.15),
   min_child_weight=(1, 10)
7. **Run Optuna for Hyperparameter Tuning**
8. **Define the Best XGBoost Model**
9. **Define Stacking Classifier**
   Initialize a stacking classifier with: XGBoost, LightGBM, CatBoost, Logistic Regression as the final meta-model.
10. **Train the Stacking Model and make predictions for test data**
11. **Evaluate the Model (Both training and testing metrics)**
12. **Confusion Matrix, ROC curve, Precision-Recall Curve, Residual Distribution, Learning Curve**
13. **K-fold cross validation**

END

4.2 Dataset Source

The dataset that is used here is [22] provided by the Karlsruher Institut für Technologie and is based on simulations done on a 4-node architecture: one main producer providing electricity to three consumers. It has 10,000 samples and 12 attributes.

4.3 Implementation Steps

The dataset is loaded using pandas. Missing values are removed using dropna() with the help of the numpy library. The feature columns and target column are chosen, and the target column is converted to numeric values of 1 or 0. The polynomial feature method is applied to ensure that the model captures high-order relationships. Here, a degree of 2 is used in polynomial engineering to capture complex and non-linear data. Exploratory data analytics is done for the

feature and target columns, and the best correlation is observed with the help of a heatmap (correlation matrix) and pair plot. Then, the obtained data is converted into a dataframe using pandas. The dataset is split according to the Pareto principle (80:20). SMOTE is used to maintain the balance in the dataset by increasing the minority class through the creation of synthetic data points. By using Tomek, redundant samples are removed. The hyperparameters for XGBoost are optimized using Optuna. A stacking classifier is created by combining XGBoost, LightGBM, and CatBoost as base classifiers, and the final estimator is logistic regression. Test data predictions are made, and the model's performance is assessed. Finally, a confusion matrix is generated to verify how the model performs under each class, which includes True Positive, True Negative, False Positive, and False Negative.

4.4 Evaluation Metrics

Evaluation metrics is used to evaluate the performance of a model. These metrics help in measuring accuracy, support, recall, precion and F1-score. A confusion matrix is used to observe how the data perform under each class. It consists of 4 components, which include True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

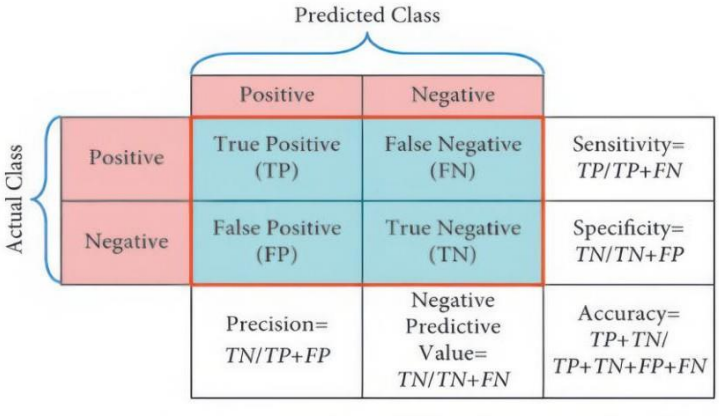


Figure 4.1: Confusion Matrix representation

Accuracy: Accuracy is the ratio of correctly predicted values to total instances and indicates the overall accuracy of the model.

$$\text{Accuracy} = \frac{\text{Number of Correct Prediction}}{\text{Total Number of Prediction}}$$

Precision: Precision measures the accuracy of positive predictions it indicates the proportion of true positive predictions among all the values which are predicted as positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: Recall measures the ability of the model to capture all positive instances, representing the ratio of true positives to the total actual positive instances.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1-score: The F1 Score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives.

$$\text{F1- score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}}$$

Support: Support refers to the number of actual occurrences of each class in the dataset during model evaluation.

Chapter 5

Implementation Details & Results

5.1 Setup

The dataset used here, provided by the Karlsruher Institut für Technologie, is based on simulations done on a 4-node architecture: one main producer providing electricity to three consumers. It has 10,000 samples and 12 attributes.

Tools Used:

Scikit-Learn was used for applying different traditional machine learning algorithms like XGBoost, LightGBM, CatBoost, and Stacking Classifier. It also included facilities for train-test split, SMOTE-Tomek resampling, and was employed for measuring performance through accuracy, precision, recall, and F1-score.

SciPy was used for mathematical computation and statistical analysis. This also includes handling complex optimization tasks with hyperparameter tuning and Optuna to improve the model's performance and minimize overfitting.

Pandas is used for loading, cleaning, and preprocessing the dataset. It also helps in handling large datasets efficiently. It assists in operations like data manipulation, preparing the data for train-test splits, and feature engineering.

NumPy is used for numerical computations and operations with arrays. It also provides support in handling large matrices during SMOTE-Tomek resampling and polynomial feature transformation.

Matplotlib and **Seaborn** libraries were used for data visualization and also helped in generating correlation heatmaps, ROC curves, precision-recall curves, learning curves, and residual plots, which were essential for interpreting the model's performance.

Google Colab is the cloud software where the entire code was able to run and served as the background for the project. It helped in preprocessing and executing large datasets seamlessly. It also facilitated the visualization of results for better analysis. This platform provided all the above facilities to perform the experiment efficiently.

5.2 Data Pre-processing

5.2.1 Synthetic Minority Over-Sampling Technique (SMOTE)

Before applying SMOTE, the distribution of the target variable was imbalanced, due to which the model had a high bias that affected the model's performance. The model was biased towards the majority class. Here, the training data was largely affected, which also resulted in low precision and recall. After applying SMOTE (Synthetic Minority Over-Sampling Technique), which is used to handle imbalanced data by oversampling the minority data and under-sampling the majority data, the model leads to higher precision, F1 score, and recall.

5.2.2 Train-Test Split

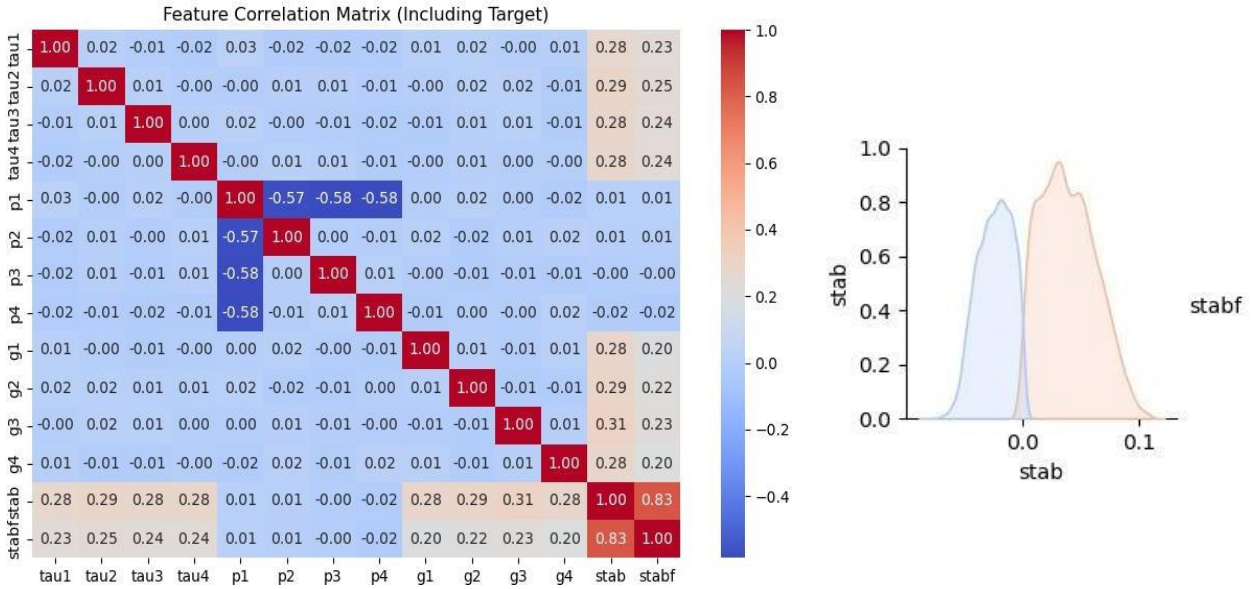
The train-test split is done to divide the entire dataset into a training set, which would be used to train the model, and a testing set, using which the performance of the trained model can be evaluated. This process is done to ensure that the model does not overfit the data and ensures generalization. According to the Pareto principle (80/20 rule), 80% of the effects come from 20% of causes. Thus, in our study, the data is split into 80% training and 20% testing while keeping the class distribution the same using stratified sampling. This prevents bias in imbalanced datasets. The random_state=42 ensures the split is reproducible, and stratify=y maintains class balance in both sets. This helps the model learn effectively and perform well on new data.

5.2.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the most important step in data science to perform initial investigations about the dataset, understand anomalies as well as patterns, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations. This step would help us visualize the dataset in the form of graphs, charts and maps, which gives the overall idea on what type of model the researcher needs to use. EDA helps to spot any unusual data or outliers and is usually done before starting more detailed statistical analysis or building models.

This study involves two major types of data analysis before proceeding to the Machine Learning (ML) model implementation. Firstly, the correlation between all the features and the target variable is analysed with the help of the correlation heatmap. From the generated correlation map, shown in Figure 5.1(a), the relationship between input and target variables is observed. There is a strong positive correlation of 0.83 between the "stab" and "stabf" features, which indicates that these are closely related. Among the input features, tau1, tau2, tau3, and tau4 show a moderate positive correlation, ranging from 0.24 to 0.29, with the target "stabf". Similarly, g1, g2, g3, and g4 features also exhibit a weaker positive correlation, ranging from 0.20 to 0.2, with the target. In contrast, the features p1, p2, p3, and p4 show a negligible correlation of about 0.01 to 0.02, indicating their poor contribution in predicting the target variable. The matrix also shows a strong negative correlation of -0.57 to -0.58 between certain pairs of features, for example, between p1

and p3, p1 and p4, and p2 and p3. Hence, due to the stronger association with the target variables, the tau and g features should be prioritized for better optimization of the model.



(a) **(b)**

Figure 5.1: (a) Correlation heatmap to depict the relationship between features (b) Pair plot between stab and stabf columns

The second type of EDA used is the pair plot or the density plot, as shown in the Figure 5.1(b). It compares the distribution of "stab" and "stabf" features. Both distributions exhibit a similar shape but show a slight shift from each other. The "stab" feature represents the blue-colored curve and has its center towards the low values, while the orange-colored curve represents the "stabf" feature and is shifted towards the right. We see that "stabf" takes higher values compared to "stab" from the shift. Despite this offset, the distribution of both features has similarities, which indicates the strong positive correlation of 0.83 as observed from the correlation matrix. The overlap region signifies that these two curves have features related to each other but are not identical.

5.3 Procedure

1. Get the preprocessed dataset in CSV format.
2. Splitting the dataset by defining the features and target. The train-test split is to be done such that training data has 80% and test data has 20% of the full dataset.
3. Applying Synthetic Minority Over-sampling Technique (SMOTE) to balance the training dataset.
4. Training the dataset with different models.
5. Training AdaBoost with RandomForest as the Base Estimator. Also analyzing its accuracy and evaluating its confusion matrix.
6. Training XGBoost model. Also analyzing its accuracy and evaluating its confusion matrix.

7. Using Optuna to find the best hyperparameters for XGBoostClassifier. Also analyzing its accuracy and evaluating its confusion matrix.
8. In Stacking Classifier (Ensemble Learning), we combine multiple models—XGBoost, LightGBM, and CatBoost—to improve performance.
9. Performance on each model is evaluated. Then, accuracy, F1 score, precision, and recall are calculated.
10. The performance is compared among these models, and the one with the best accuracy is displayed.

Table 5.1: Comparison between various algorithms used in the study

MODEL	PARAMETER	VALUES EXPLORED	BEST VALUE	ACCURACY
AdaBoostClassifier+(baseclassifier=RandomForestClassifier)	Number of estimators	100	100	91.95%
	Learning rate	0.01	0.01	
XGBoostClassifier	Oversampling strategy	SMOTE	None	93%
	Number of estimators	100	100	
	Learning rate	0.05	0.05	
XGBoostClassifier(Hyperparameter tuning)	Oversampling strategy	SMOTE	None	95.25%
	Number of estimators	10-150	140	
	Learning rate	0.01-0.2	0.1480	
	Maximum depth	3-10	4	
	Minimum child weight	1-10	6	
XGBoostClassifier(Hyperparameter tuning)+ CatBoost(n_estimator = 100, learning_rate = 0.05)	Oversampling strategy	SMOTE	None	96.35%
	Number of estimators	10-200	131	
	Learning rate	0.01-0.1	0.0897	
	Maximum depth	3-10	9	
	Minimum child weight	1-10	6	
XGBoostClassifier(Hyperparameter tuning)+ LightGBM(n_estimator=100, learning_rate = 0.05)	Oversampling strategy	SMOTE	None	95.65%
	Polynomial Features	2	2	
	Number of estimators	10-200	171	
	Learning rate	0.01-0.1	0.0559	
	Maximum depth	3-10	9	
	Minimum child weight	1-10	3	
XGBoostClassifier(Hyperparameter tuning)+ Stackingclassifier(XGBoost,LightGBM, CatBoost)	Oversampling strategy	SMOTE	None	96.55%
	Polynomial Features	2	2	
	Number of estimators	100-200	184	
	Learning rate	0.01-0.15	0.1107	
	Maximum depth	3-6	6	
	Minimum child weight	1-10	5	

The Table 5.1 consists of all the models that we used in our code and their respective parameters, values explored, best value, and accuracy. The parameters include an oversampling strategy, polynomial features, number of estimators, learning rate, maximum depth, and minimum child weight. Among the models evaluated, the AdaBoost classifier with a RandomForest base class achieved an accuracy of 91.95%, which has n_estimators=100 and learning rate=0.01 and has

the best value of the same. Secondly, the XGBoost model was used and improved the accuracy to 93%. Here, SMOTE was used as an oversampling strategy with n_estimators=100 and learning rate=0.05. Significant gains were observed when hyperparameter tuning with the XGBoost model was introduced to the dataset, and the resulting accuracy increased to 95.25%. The parameters observed for this case have its best n_estimators at 140 between the range of 10 to 150. Also, the learning rate has its best value at 0.1480 between the range of 0.01 and 0.2. The maximum depth had a best value of 4, and a minimum child weight of 6. Introducing hybrid models led to even better results by combining XGBoost with CatBoost using optimized hyperparameters, pushing the accuracy to 96.35%. The parameters used for this case have its best n_estimators at 131 between the range of 10 to 150. Also, the learning rate has its best value at 0.0897 between the range of 0.01 and 0.1. The best values for maximum depth and minimum child weight are 9 and 6, respectively. Further enhancement occurred when LightGBM was integrated, yielding 95.65% accuracy. The parameters used for this case have its best n_estimators at 171 between the range of 10 to 200. Also, the learning rate has its best value at 0.0559 between the range of 0.01 and 0.1. Additionally, maximum depth has its best value at 9, and minimum child weight has its best value at 3. Finally, the Stacking Classifier, incorporating XGBoost, LightGBM, and CatBoost, achieved the highest accuracy of 96.55%. Here, SMOTE was used as an oversampling strategy, and polynomial features of 2nd degree were used. The other parameters include n_estimators between the range of 10 to 200, with its best value at 184. The learning rate has its best value at 0.1107 between the range of 0.01 and 0.1. The best values for maximum depth and minimum child weight are 6 and 5, with their ranges between 3 to 9 and 1 to 10, respectively. This tabulation highlights the effectiveness of ensemble learning and hyperparameter tuning in improving classification accuracy. The Stacking Classifier achieved the best performance, suggesting that combining diverse models enhances predictive power. Figure 5.2 shows how the accuracy varies with increase in the number of n_estimators from 50 to 300.

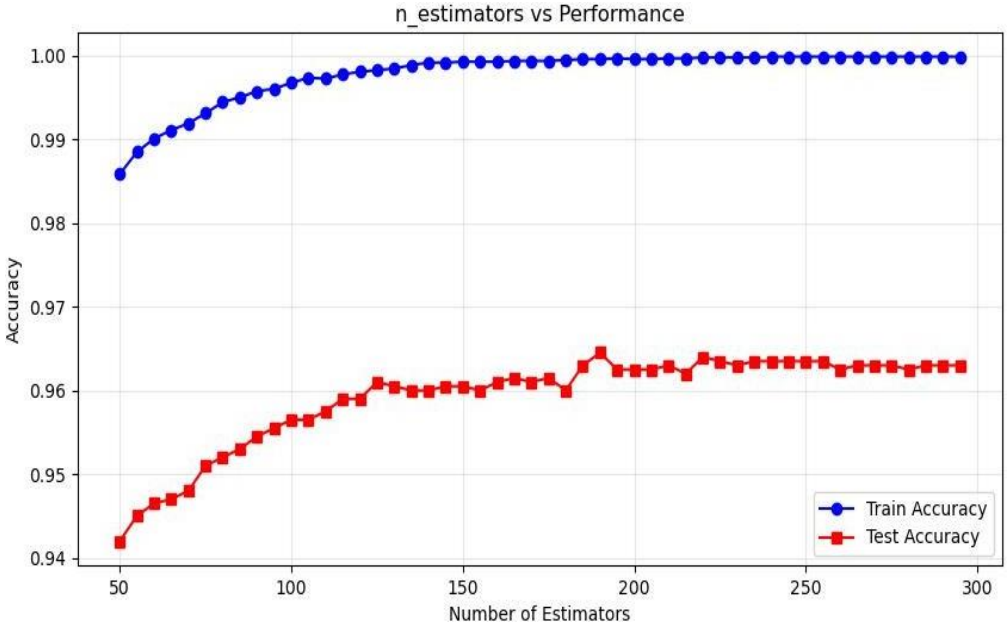


Figure 5.2: Plot between number of n_estimators and accuracy

5.4 Results

Confusion Matrix

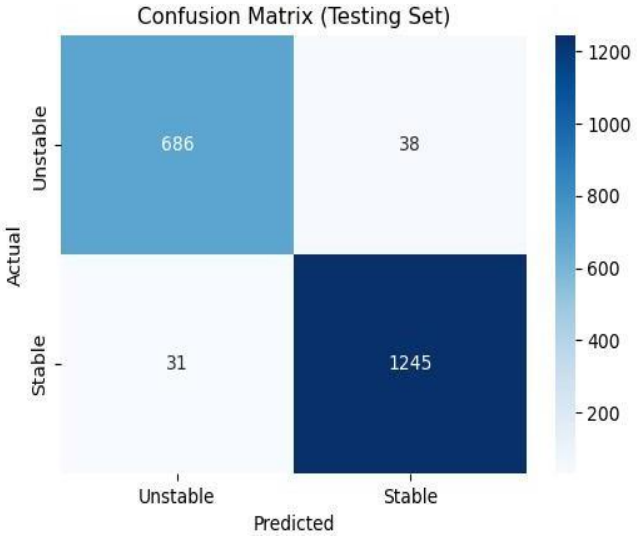
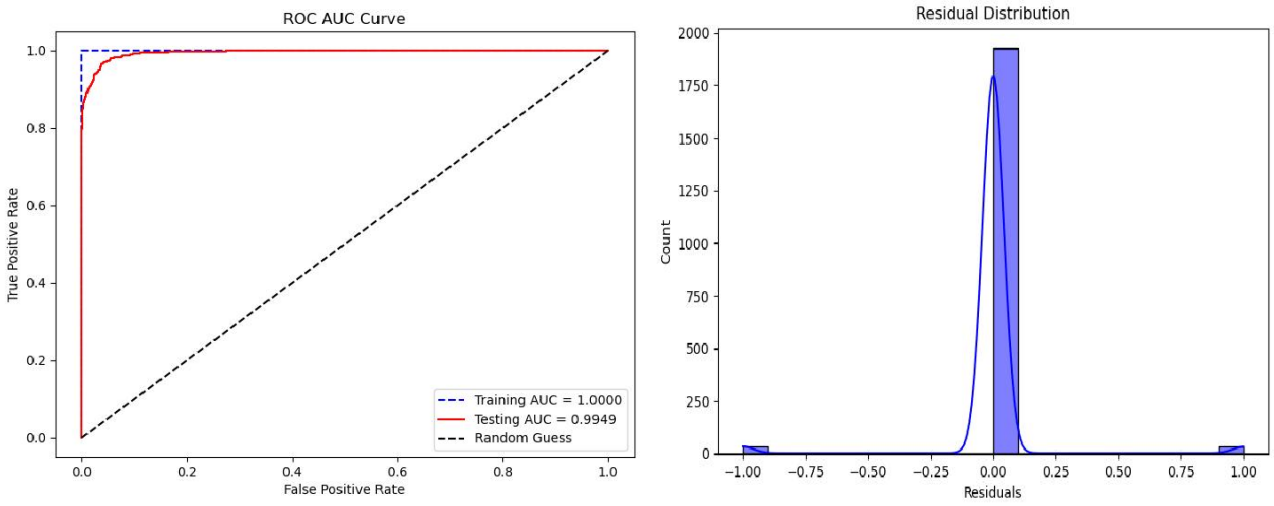


Figure 5.3: Confusion Matrix

The confusion matrix in Figure 5.3 gives an idea of the proportion of stable and unstable data and also about the positive and negative parameters effectively. Out of the actual 686 unstable cases, the model correctly classified 686 as unstable (true negatives) and incorrectly labeled 38 as stable (false positives). Out of the 1276 stable cases, the model accurately predicted 1245 as stable (true positives) but incorrectly labeled 31 as unstable (false negatives). This indicates that the model performs well in distinguishing between stable and unstable classes at all times. The low rates of false positives and false negatives suggest high accuracy and well-balanced performance across both classes. The confusion matrix indicates how the model performs well in error minimization and demonstrates good predictive performance during the test.



(a) **(b)**

Figure 5.4: (a) ROC AUC Curve (b) Residual Distribution

ROC-AUC Plot

The Receiver Operating Characteristic (ROC) plot, shown in Figure 5.4(a) examines the discriminative ability of the model to distinguish between stable and unstable classes. True Positive Rate (sensitivity) versus False Positive Rate is plotted, resulting in data as to whether there is a satisfactory trade-off between correctly classifying and avoiding false alarms. Training AUC (Area Under the Curve) is 1.0000, which implies perfect discrimination in training between both classes. The AUC for Testing is 0.9949, and it demonstrates that the model predicts very well on unseen data, which suggests that the model generalizes extremely well without extreme overfitting. The value of AUC should be as close to 1 as possible. The closer to 1 the AUC value is, the better the model will be in classifying between classes. Both the curves are near the upper-left part, indicating a high true positive rate and a low false positive rate. The random guess line (AUC = 0.5) is the baseline, and the model performs much better than that. That is a great and solid classification system.

Residual Distribution

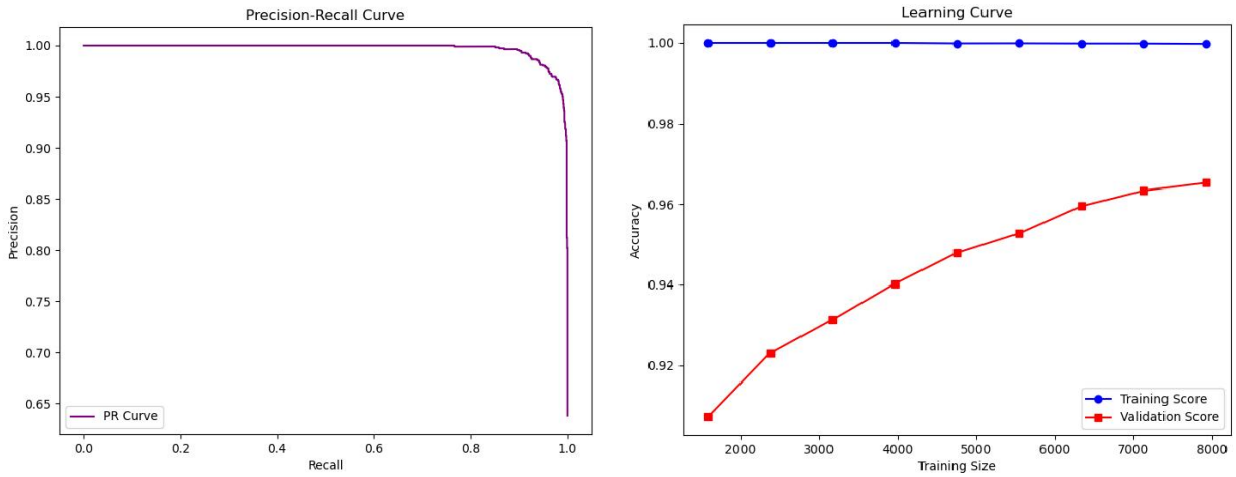
The residual plot in Figure 5.4(b) displays the difference between real and predicted values. A random and evenly spread residual pattern will be an indication that the model picks up on the underlying pattern without systematic errors. The absence of identifiable patterns will mean that the model is free of bias or underfitting. The fact that residuals are centered on zero will be a sign of a well-calibrated model. This analysis confirms the effectiveness of using SMOTE and polynomial feature engineering, which reduces errors and improves the model's capacity to predict complex, non-linear relationships in the smart grid dataset.

Precision-Recall Curve

The precision-recall curve [Figure 5.5(a)] shows the trade-off between precision and recall (sensitivity). The shape of the curve shows a high area under the precision-recall curve, which signifies a strong performance on imbalanced data. The model has high precision as well as a high recall, which implies that it finds unstable grid conditions very well without any large increase in the number of false positives. Hence, it could be highly reliable for our smart grid applications where accurate identification of unstable conditions is crucial for operational safety and reliability.

Learning curve

Learning curve in Figure 5.5(b) shows performance of model with respect to increasing training dataset. Here, accuracy on the training set is very flat across and representative that the model is fitting extremely to training data. Validation accuracy flattens at comparatively lower points but tracks the training curve very accurately, which exhibits very little signal of overfitting. This crossing implies that the model is able to generalize to novel data and future training data increments will only be in a state to provide incremental value. This is an indicator of the fact that ensemble strategy with hyperparameter tuning is strong and demonstrates reliable performance even when there is an increased amount of data.



(a) **(b)**

Figure 5.5: (a) Precision-Recall Curve (b) Learning Curve

```
Cross-Validation Accuracy Scores: [0.96115035 0.95961636 0.96415952 0.95608279 0.96314992]
Mean Accuracy: 0.9608
Standard Deviation: 0.0028
```

5.5 Analysis

Firstly, the Adaboost Classifier with Random Forest as the base classifier was used to train the model with the given dataset. It was observed that the model provided an accuracy of 91.95%. Then, XGBoost was tried with n_estimator = 100, and the accuracy increased to 93%. The above table shows that XGBoost with hyperparameter tuning gives a very good result of 95.1%. It also shows the need for polynomial feature selection, where a combination of features is also correlated and then the model is trained. This has improved the accuracy of the model. Using stacking models with XGBoost, LightGBM, and CatBoost has also shown a significant improvement in model accuracy. When the n_estimator of 100 gives a much better result than other values, increasing the accuracy to 96.45%. Further improvements can be made by including feature selection techniques.

Table 5.2: Classification Report for Training data

-	*Precision*	*Recall*	*F1-Score*	*Support*
0	*1.00*	*1.00*	*1.00*	*4953*
1	*1.00*	*1.00*	*1.00*	*4953*
Accuracy			*1.00*	*9906*
Macro Average	*1.00*	*1.00*	*1.00*	*9906*
Weighted Average	*1.00*	*1.00*	*1.00*	*9906*

Table 5.3: Classification Report for Testing data

-	*Precision*	*Recall*	*F1-Score*	*Support*
0	*0.96*	*0.94*	*0.95*	*724*
1	*0.97*	*0.98*	*0.97*	*1276*
Accuracy			*0.96*	*2000*
Macro Average	*0.96*	*0.96*	*0.96*	*2000*
Weighted Average	*0.96*	*0.96*	*0.96*	*2000*

5.6 Comparison

Various machine learning techniques like XGBoost, Random Forest, AdaBoost, SVM, and Gradient Boosting were explored in many papers to enhance smart grid stability. XGBoost without hyperparameter tuning had an accuracy of 95%. We improved this by using hyperparameter tuning, resulting in 96.45% accuracy, surpassing the base paper by 1.45% using a stack classifier by stacking CatBoost, LightGBM, and XGBoost. Additionally, the SMOTE-Tomek method balances the dataset and improves the model's performance characteristics. It also reduces bias toward stable conditions and enhances the detection of unstable classes. Using feature engineering, we can effectively capture the data's non-linear relationships. Then, Optuna tuning furthe helps to enhance the performance of XGBoost.

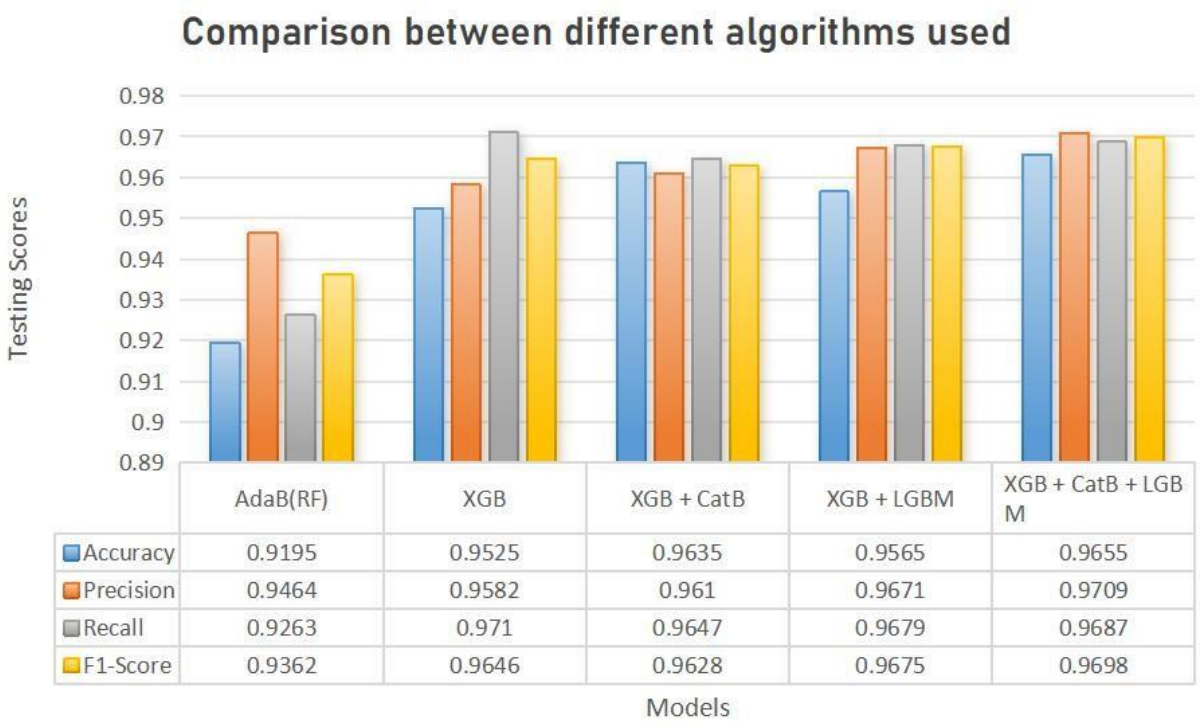


Figure 5.6: Comparison between the performance metrics of different algorithms used

Chapter 6

Result Discussion

6.1 Insights

This study gives an important thought on using advanced methods, combining models, and checking the stability of smart grids. The major finding here is that the combination of models, called the Stacking Classifier, is better than using Support Vector Machine (SVM) and Random Forest techniques. This hybrid approach helps us achieve higher accuracy and reliability. By adjusting the model settings and accuracy, we improved from 93% with basic XGBoost settings to 96.35% with a tuned stacking classifier, highlighting the importance of tweaking models for optimal results. This study also emphasizes the importance of Optuna in hyperparameter optimization, which helps models tackle complex data and minimize errors like overfitting or underfitting. Another important aspect is dealing with imbalanced datasets, which is a common issue in assessing smart grid stability. Using SMOTE-Tomek resampling helps balance data, resulting in faster predictions, but it can also introduce artificial noise, impacting prediction reliability. This study also explores other methods for balancing classes, such as generative models or adaptive synthetic sampling, which can enhance the model's generalization.

6.2 Implications

6.2.1 Practical Implications

In this project we have enhanced grid stability predictions using ensemble learning and hyperparameter tuning which will be improving accuracy and also on how to handle data that is not stable from renewable energy sources. We know that black-box deep learning models are not suitable for it since there is no transparency and data is also not safe here, so we include other methods and so it offers high accuracy with transparency and secure the data perfectly which will be making our project acceptable for grid engineers. Our project is also scalable and also able to do real-time analysis which can prevent blackouts and voltage issues which can be ensuring good energy distribution to the users.

6.2.2 Theoretical Implications

In this project by using stacking classifiers like (XGBoost, LightGBM, CatBoost) we can improve the power system stability of the system. We, in this project, tried to improve accuracy than the traditional methods. It will tackle the class which is not balanced by using SMOTE-Tomek

resampling and also by improving the predictions by polynomial feature engineering. Optuna tuning corrects the model performance which is a shift toward hybrid AI-ML models for smart energy management and real-time grid monitoring.

6.3 Limitations

Some of the limitations of this study include the following:

Time-consuming Stacking Classifier model: This study has used a Stacking Classifier using XGBoost, LightGBM and CatBoost. The proposed model shows higher accuracy and efficient grid optimization. However, the training process involved is time-consuming. The power networks meet huge demands every day and thus, they require faster decision-making to predict the stability of these smart grids. So, less time-consuming models could be implemented in future research.

Model Overfitting and real-time adaption: The cases of unstable state are rare in smart grids. So, the model may fail to capture genuine patterns in the data. This study uses SMOTE-Tomek and hyperparameter tuning. But, these techniques alone cannot improve the model's accuracy in real-world scenarios. The model must have the ability to learn from real-time data and future improvements need to focus on this aspect.

Adaption to dynamic grid conditions: This model is trained over the past data. Optuna helps the model in testing different configurations. However, this requires high computing power to find the best parameters. Additionally, the model requires adjustments for better performance in dynamic environments. Thus, this proposed approach is limited to specific conditions and cannot be completely accepted as a universal solution.

Chapter 7

Conclusion and Future Works

7.1 Summary

Our project is developing a high accurate model which is used for understanding and finding smart grid stability using machine learning techniques. By adding XGBoost, LightGBM, and CatBoost, we have achieved this model with 96.45% accuracy, output which is far better than older methods like SVM and Random Forest. The important changes done in this project to make it better include better feature selection, class balancing by using SMOTE-Tomek, and Optuna-based tuning is used to prevent overfitting. Another important feature which adds polynomial feature engineering which helps the model to understand hard patterns which makes it more comfort. It is not like the deep learning models, this new approach is balancing accuracy and interpretability which is making it useful for real-world power grids. The model is scalable, can be used in any environment, and able to real-time usage which is the main reason to help maintain grid stability and prevent power wastage.

7.2 Contributions

We develop our project tittled smart grid stability prediction using machine learning in which we achieve at the rate of 96.45% accuracy by using a stacking classifier XGBoost, LightGBM, and CatBoost. It overcomes the older methods namely SVM, Random Forest, and AdaBoost by addressing class imbalance with SMOTE-Tomek, ensuring fair and accurate predictions of grid instability. The main techniques used here is Optuna tuning for hyperparameter optimization, reducing overfitting and improving adaptability to new data. Polynomial feature engineering is one of the model which helps to identify complex power system relationships thus enhancing decision-making. This research shows ensemble learning is more effective than standalone models. It supports real-time grid monitoring, AI-driven energy management, and future advancements like IoT-based smart meters and reinforcement learning for better power system optimization.

7.3 Key Takeaways

Now let us see the key takeaways on how machine learning improves grid stability prediction by achieving 96.45% accuracy with the help of stacking classifier – XGBoost, LightGBM, and CatBoost which is better than the SVM, Random Forest, and AdaBoost. Optuna tuning give good performance and also SMOTE-Tomek handles class imbalance for accurate

classification, and polynomial feature engineering captures complex grid relationships. We also used ensemble learning to enhance reliability, making this model useful for energy distribution, grid monitoring, and also provides trust to the grid users.

7.4 Improvements

For future enhancements,several ways can be used to optimize the model's performance.Methods such as pruning, quantization, and knowledge distillation can make the model faster and more efficient. We can also use techniques such as streaming data processing and incremental learning and also approaches like Bayesian optimization, genetic algorithms, or reinforcement learning to make the model perform even more better. We can also use selection by using methods like PCA, autoencoders, or SHAP analysis to improve predictions.

7.5 Extensions

Our study tells the importance of how machine learning models can efficiently predict smart grid stability.Presently, the model uses historical grid data, but changes will be found in the actual power systems. In the future, we should also focus on systems that can learn and update with real-time data. It can also be extended by adding features like IoT-based smart meters and edge computing, which would allow collective live grid stability data, making the system easier to adapt to power supply and demand changes.Even though the stacking classifier boosts accuracy, it is very hard to use due to its high computation time.Future aspects should use AI decisions enabling operators to act with confidence. Another research area is hyperparameter tuning, which is more efficient. It uses Optuna for automated tuning, which is very time-consuming and costly. Further studies can try faster methods like Bayesian optimization.The changes made in the logistic regression meta-learners to algorithms like Random Forest, XGBoost, or MLP neural networks will also enhance performance. We can also add cloud-based distributed computing, and federated learning could help scale the model to multiple grids, ensuring data privacy and security.

7.6 Challenges

While doing this project we face some of the challenges in smart grid stability prediction using machine learning. We found that this may require high computational costs which makes real-time use difficult, requiring techniques like model compression to optimize performance. Class imbalance from SMOTE-Tomek resampling may cause overfitting, which can be improved with hybrid resampling and anomaly detection. Some methods like Hyperparameter tuning with Optuna are resource-intensive so that Bayesian optimization and reinforcement learning could enhance efficiency. The stacking model could not explain, which can be covered with XAI, SHAP, and LIME. We found that the Ensemble models are not able to meet with the time-series patterns, and deep learning could help but demand large datasets and processing power.

References

[1] Spiru Paraschiv, Lizica Simona Paraschiv, Trends of carbon dioxide (CO2) emissions from fossil fuels combustion (coal, gas and oil) in the EU member states from 1960 to 2018, Energy Reports, Volume 6, Supplement 8, 2020, Pages 237-242, ISSN 2352-4847, https://doi.org/10.1016/j.egyr.2020.11.116.

[2] Luis Corona, Asuncion Mochon, Yago Saez, Electricity market integration and impact of renewable energy sources in the Central Western Europe region: Evolution since the implementation of the Flow-Based Market Coupling mechanism, Energy Reports, Volume 8, 2022, Pages 1768-1788, ISSN 2352-4847, https://doi.org/10.1016/j.egyr.2021.12.077.

[3] Zaid Allal, Hassan N. Noura, Ola Salman, Khaled Chahine,Leveraging the power of machine learning and data balancing techniques to evaluate stability in smart grids,Engineering Applications of Artificial Intelligence,Volume 133,Part C,2024,108304,ISSN0952-1976,https://doi.org/10.1016/j.engappai.2024.108304.(https://www.sciencedirect.com/science/article/pii/S0952197624004627)

[4] Seong Gyu, Yang & Kim, Beom & Son, Seung-Woo & Kim, Heetae. (2021). Power-grid stability prediction using transferable machine learnings. 10.48550/arXiv.2105.07562.

[5] Raza, Muhammad Qamar & Nadarajah, Mithulananthan & Li, Jiaming & Lee, Kwang. (2018). Multivariate Ensemble Forecast Framework for Demand Prediction of Anomalous Days. 10.48550/arXiv.1811.09339.

[6] Boutahir, Mohamed Khalifa & Hessane, Abdelaaziz & Farhaoui, Yousef & Azrour, Mourade. (2023). An Effective Ensemble Learning Model to Predict Smart Grid Stability Using Genetic Algorithms. 10.1007/978-3-031-25662-2_11.

[7] Gupta, Sudha & Kambli, Ruta & Wagh, Sushama & Kazi, Faruk. (2015). Support-Vector-Machine-Based Proactive Cascade Prediction in Smart Grid Using Probabilistic Framework. Industrial Electronics, IEEE Transactions on. 62. 2478-2486. 10.1109/TIE.2014.2361493.

[8] Aliyeva, Laman & Abdullayev, Nihat. (2024). Hybrid Deep Learning Approach Towards Smart Grid Stability Prediction. 1-5. 10.1109/ENERGYCON58629.2024.10488774.

[9] Singh, Arvind & Seshu Kumar, Rangu & Bajaj, Mohit & Khadse, Chetan & Zaitsev, Ievgen. (2024). Machine learning-based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources. Scientific Reports. 14. 10.1038/s41598-024-70336-3.

[10] Ramasamy, Karthikeyan & Sundaramurthy, Arivoli & Velusamy, Durgadevi. (2023). Assessment and classification of grid stability with cost-sensitive stacked ensemble classifier. Automatika. 64. 783-797. 10.1080/00051144.2023.2218164.

[11] Massaoudi, Mohamed & Abu-Rub, Haitham & Chihi, Ines & S. Refaat, Shady & Oueslati, Fakhreddine. (2021). An Effective Ensemble Learning approach-Based Grid Stability Assessment and Classification. 10.1109/KPEC51835.2021.9446197.

[12] Shi, Zhongtuo & Yao, Wei & Li, Zhouping & Zeng, Lingkang & Zhao, Yifan & Zhang, Runfeng & Tang, Yong & Wen, Jinyu. (2020). Artificial intelligence techniques for stability analysis and control in smart grids: Methodologies, applications, challenges and future directions. Applied Energy. 278. 115733. 10.1016/j.apenergy.2020.115733.

[13] H. F. Ezzahra, M. Aatila, M. Lachgar and A. Abdali, "Predicting smart grid stability using machine learning algorithms," 2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC), El Jadida, Morocco, 2022, pp. 1-6, doi: 10.1109/ISIVC54825.2022.9800733.

[14] R. Singh, A., Kumar, R.S., Bajaj, M. et al. Machine learning-based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources. Sci Rep 14, 19207 (2024). https://doi.org/10.1038/s41598-024-70336-3

[15] "Smart Grid Stability Prediction with Machine Learning", DOI: 10.37394/232016.2022.17.30

[16] O. A. Alimi, K. Ouahada and A. M. Abu-Mahfouz, "A Review of Machine Learning Approaches to Power System Security and Stability," in IEEE Access, vol. 8, pp. 113512-113531, 2020, doi: 10.1109/ACCESS.2020.3003568

[17] M. M. Asiri, G. Aldehim, F. A. Alotaibi, M. M. Alnfiai, M. Assiri and A. Mahmud, "Short-Term Load Forecasting in Smart Grids Using Hybrid Deep Learning," in IEEE Access, vol. 12, pp. 23504-23513, 2024, doi: 10.1109/ACCESS.2024.3358182.

[18] V. Arzamasov, K. Böhm and P. Jochem, "Towards Concise Models of Grid Stability," 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, Denmark, 2018, pp. 1-6, doi: 10.1109/SmartGridComm.2018.8587498.

[19] Qasem Abu Al-Haija & Abdallah A. Smadi & Mohammed F. Allehyani, 2021. "Meticulously Intelligent Identification System for Smart Grid Network Stability to Optimize Risk Management," Energies, MDPI, vol. 14(21), pages 1-19, October.

[20] Smart Grid Stability Prediction: A Comparative Analysis of Machine Learning Models by Vishal Chaudhary1, Deepansh Kulshrestha1, Kunal Bharadwaj1

[21] Xun Gong, Xiaozhe Wang, Bo Cao,On data-driven modeling and control in modern power grids stability: Survey and perspective, Applied Energy, Volume 350,2023,121740,ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2023.121740.

[22] Arzamasov, V. (2018). Electrical Grid Stability Simulated Data [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5PG66.