Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2697303, 15 pages
https://doi.org/10.1155/2022/2697303

Research Article

A Machine Learning-Based Model for Stability Prediction of Decentralized Power Grid Linked with Renewable Energy Resources

Muhammad Ibrar,[1] Muhammad Awais Hassan,[1] Kamran Shaukat,[2,3] Talha Mahboob Alam,[4] Khaldoon Syed Khurshid,[1] Ibrahim A. Hameed,[5] Hanan Aljuaid,[6] and Suhuai Luo[2]

[1] *Department of Computer Science, University of Engineering and Technology, Lahore 58590, Pakistan*
[2] *School of Information and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia*
[3] *Department of Data Science, University of the Punjab, Lahore 54890, Pakistan*
[4] *Department of Computer Science and Information Technology, Virtual University of Pakistan, Lahore 58590, Pakistan*
[5] *Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Trondheim, Norway*
[6] *Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), P.O. Box 84428, Riyadh 11671, Saudi Arabia*

Correspondence should be addressed to Kamran Shaukat; kamran.shaukat@uon.edu.au,
Talha Mahboob Alam; talhamahboob95@gmail.com, and Ibrahim A. Hameed; ibib@ntnu.no

Received 6 June 2022; Revised 6 July 2022; Accepted 28 July 2022; Published 23 August 2022

Academic Editor: Farman Ullah

A decentralized power grid is a modern system that implements demand response without requiring major infrastructure changes. In decentralization, the consumers regulate their electricity demand autonomously based on the grid frequency. With cheap equipment (i.e., smart meters), the grid frequency can be easily measured anywhere. Electrical grids need to be stable to balance electricity supply and demand to ensure economically and dynamically viable grid operation. The volumes of electricity consumed/produced (p) by each grid participant, cost-sensitivity (g), and grid participants' response times (tau) to changing grid conditions affect the stability of the grid. Renewable energy resources are volatile on varying time scales. Due to the volatile nature of these renewable energy resources, there are more frequent fluctuations in decentralized grids integrating renewable energy resources. The decentralized grid is designed by linking real-time electricity rates to the grid frequency over a few seconds to provide demand-side control. In this study, a model has been proposed to predict the stability of a decentralized power grid. The simulated data obtained from the online machine learning repository has been employed. Data normalization has been employed to reduce the biased behavior among attributes. Various data level resampling techniques have been used to address the issue of data imbalance. The results showed that a balanced dataset outperformed an imbalanced dataset regarding classifiers' performance. It has also been observed that oversampling techniques proved better than undersampling techniques and imbalanced datasets. Overall, the XGBoost algorithm outperformed all other machine learning algorithms based on performance. XGBoost has been given an accuracy of 94.7%, but while combining XGBoost with random oversampling, its accuracy prediction has been improved to 96.8%. This model can better predict frequency fluctuations in decentralized power grids and the volatile nature of renewable energy resources resulting in better utilization. This prediction may contribute to the stability of a decentralized power grid for better distribution and management of electricity.

1. Introduction

The majority of the global energy supplies for electricity generation are nonrenewables (oil, gas, coal, etc.) [1]. These nonrenewable energy resources are depleting quickly. They are polluting the environment and causing global warming due to the emission of various greenhouse gasses [2]. Due to these limitations of nonrenewable resources, the world's energy policies are shifting towards renewable resources for clean and sustainable energy [3]. Globally, the current share of renewable energy in electricity generation is 24%, which is expected to grow by 44% by 2030 [4, 5]. Pakistan has enormous potential for renewable energy generation due to long sunshine hours, and its coastal belt has promising wind speeds. According to the Alternate and Renewable Energy Policy 2019 approved by Pakistan, the country has planned to grow its share of renewable energy in electricity generation from 4% to 30% by 2030, excluding hydropower [5]. However, the volatile nature of many renewable energy sources is a well-known challenge [6, 7]. A more flexible approach is required to balance energy demand and supply linked with renewable energy resources since renewable energy resources are more susceptible to fluctuations than nonrenewable energy resources [8, 9]. Various approaches to managing supply and demand have been presented for such a fluctuating power grid. The various smart grid concepts' core idea is to manage consumer demand which is a significant paradigm shift from current grid operating schemes [7].

A decentralized approach means a resource is self-dispatched with rubrics that can be defined in isolation of other resources or in coordination with them. In decentralization, the consumers regulate their electricity demand autonomously based on the grid frequency [6, 10]. The decentralized grid approach was first suggested a few years ago but only recently received much attention by implementing demand response without major infrastructural changes. Electrical grids need to be stable to balance energy supply and demand to ensure economically and dynamically viable grid operation [11]. During periods of excess power, the frequency increases but decreases during periods of underproduction. Grid frequency monitoring is a low-cost and easy way to determine grid stability. The grid's frequency changes when there is an undersupply or oversupply of electricity in the grid [12]. With cheap equipment (i.e., smart meters), consumers can easily measure the grid frequency anywhere.

ML techniques have widely used in the domain of education [13, 14], software measurement [15–17], decision support system [18, 19], social sciences [20, 21], healthcare [22–24], and disease diagnosis [9, 25]. Numerous computational methods were used in the renewable energy domain [26–30]. These include the prediction of the decentralized grid using a decision tree (DT) [11] and Hybrid Kernel Ridge Regression-Extreme Gradient Boosting (KRR-XGBoost) for distributed power systems [31]. The artificial neural network (ANN), support vector regression (SVR), and regression trees (RT) were used for the power output of photovoltaic (PV) systems [32]. For electricity load prediction, ehanced multilayer perceptron (MLP), enhanced support vector machine (SVM), and enhanced logistic regression (LR) were used [33]. Gradient boosted decision tree (GBDT), LR, random forest (RF), and MLP classifier were applied for smart grid stability prediction [34]. ANN, RF, and SVM were used for voltage stability forecasting in transmission systems [35]. ANN, linear discriminant analysis (LDA), and Naïve Bayes (NB) predicted solar power ramp events [36]. Ensemble diverse-extreme learning machine was applied [37] by sampling key variables over a broad feasible solution space. All these studies were not able to achieve effective results. Few studies utilized a large number of features [32, 35, 37] and employed a limited amount of data for modelling purposes [33, 37]. Previous research did not work on class imbalance which caused insufficient results.

Resampling techniques have been used in this study in combination with ML algorithms to predict the stability of a decentralized electricity grid. This study tests the hypothesis that ML algorithms combined with resampling techniques can provide highly accurate predictions for the stability of decentralized electricity grids. ML algorithms can detect trends and anomalies in datasets and thus help grid system operators to make real-time decisions for better distribution of available electricity [38]. Different approaches were used for the stability prediction of power grids, but effective results were not achieved [11, 31–37]. To the best of our knowledge, the available literature concluded that resampling techniques were not used to balance the data for grid stability prediction. Furthermore, only accuracy was used as an evaluation metric in previous studies, but other important metrics like precision, recall, and receiver operating characteristic (ROC) curve were not evaluated.

Contributions were as follows: this research possesses various contributions to decentralized power grid stability prediction.

(1) Latest dataset from the University of California Irvine (UCI) Machine Learning repository has been used to build a ML model for decentralized power grid stability prediction

(2) The data imbalance issue has been explored by comparing the different resampling techniques and evaluating the performance that which resampling technique has given efficient results with a ML classifier

(3) Lastly, our proposed model may help better prediction of the stability of a decentralized electricity grid, which may ultimately help in better distribution. Frequency fluctuations in the decentralized grid due to renewable energy resources can be predicted with the proposed model for better utilization of these renewable energy resources. To test the effectiveness of our proposed technique, it has been verified on the Electricity Grid Dataset. It can be applied to any real-time dataset related to decentralized grid frequency

The rest of the paper is structured as follows:

The existing techniques for grid stability prediction are analyzed in Section 2. Section 3 includes the proposed solution. The evaluation metrics are described in Section 4. The results of our study have been discussed and analyzed in Section 5. The research is summed up, and the problem definition is restated in Section 6. The study's conclusion, challenges, and limitations are described, and suggestions for future improvements are discussed in section 7.

2. Literature Review

Previous studies have mostly worked on conventional centralized grids with few frequency fluctuations [33]. However, decentralized power grids connected with renewable energy resources involve strong fluctuations on varying time scales, including seasonal, intraday, and short-time fluctuations [7]. The previous studies used imbalanced data to predict the grid stability [11, 31–37]. Abu Al-Haija et al. [39] proposed a system using various ML models to classify stability records in smart grid networks. Seven machine learning architectures are specifically examined, including SVM, DT, LR, NBC, LDC, and GBDT. A recent and substantial dataset for the stability of smart grid networks (SGN Stab2018) was also used to test the system's performance, and it received high marks for classification. Breviglieri et al. [40] studied deep learning models to solve fixed inputs and equality issues in decentralize smart grid control (DSGC) system. By removing those constrictive assumptions on input values, they examined the DSGC system using several optimized deep learning models to forecast smart grid stability.

Massaoudi et al. [41] proposed an accurate stacking ensemble classifier (SEC) for decentralizing smart grid control stability prediction. Using a supervised learning approach, the presented method showed a fantastic ability to categorize the grid instabilities accurately. Numerical findings validate the excellent effectiveness of the suggested model. Arzamasov et al. [11] predicted the results of decentralized grid stability using a DT algorithm. To determine the stability/instability of the grid, they solved a numerical optimization problem called the characteristics roots equation. Positive real numbers indicated instability, and negative real numbers indicated a stable grid state. Yin et al. [31] developed a KRR-XGBoost model to forecast the stability of distributed power systems and provide effective design guidelines and cost optimization for these systems. The grid stability index, the grid stability predictor (stable/unstable), and the factors affecting the grid's stable state covered the data input components. Ali et al. [8] proposed an optimization-based method to smooth voltage. To extend the lifespan of the electric vehicle (EV) battery, EV power fluctuations and their minimum preset state of charge (SOC) are considered in the proposed optimization model.

Different ML models were applied to test their ability to forecast PV power output by Theocharides et al. [32]. SVR, ANN, and RT were explored, each with its own hyperparameters and features. Each model's output power prediction performance was evaluated on real-world PV generation data for one year and compared to a developed persistence model. The basic purpose was to build an association between the input features and their output. Bano et al. [33] utilized ML techniques, i.e., enhanced MLP, enhanced SVM, and enhanced LR, to forecast the electricity load. To forecast New York City's load and price, they used hourly data from 2016 to 2017. Classification and regression tree and recursive feature elimination were used for feature selection. Singular Value Decomposition was used to extract the features. Moldovan and Salomie [34] presented a feature extraction-based ML approach to predict the stability of a smart grid using the Python tsfresh package. They used ML and statistical methods to detect sources of instability and made feature selection before applying classifiers. Their study used four classifiers: LR, GBDT, RF, and MLP. Ali et al. [29] proposed an optimization approach to determine microgrids' optimal locations and sizes of photovoltaic and wind generation systems. They created a bilevel metaheuristic-based method to solve the planning model. Various simulations and study cases are run to evaluate the viability of the proposed model.

Malbasa at el. [35] predicted voltage stability in transmission systems using active machine learning. Their key contribution is applying pool-based active learning techniques to power system measurements like synchrophasor data, a tool for determining voltage stability. Experiments on synthetic data obtained from a complex power system simulation model are used to test their method. The experiments focused on margins of voltage stability prediction in a transmission network using ML techniques. Abuella et al. [36] predicted solar power ramp events to handle high renewable generation ramp rates, energy storage systems, energy management, and voltage regulator settings on distributed generation feeders. They used LDA, ANN, and NB to forecast solar power ramp events. Baltas et al. [37] proposed a response-based model to forecast a benchmark system's stability following a serious disturbance. They used ensemble-based multiple classifiers. Simulated data generated through Spider IDE was used. For the ensemble's final output, three separate approaches were considered. They used a majority voting scheme in their first method. In the second method, a variant of the boosting technique was considered that uses the weight factor and a constant. Finally, in the third approach, an all-or-nothing technique was considered. Ali et al. [30] presented an interval optimization method to schedule EV optimally. The goal was to reduce network active power losses and overall voltage magnitude deviation while considering system-wide restrictions. The best day-ahead scheduling of EV was done using the proposed method on a 33-bus distribution system. Various case studies were conducted to evaluate the viability of the suggested approach. The summary of techniques of related articles is presented Table 1.

3. Methodology

The proposed framework for decentralized grid stability prediction is presented in Figure 1. Our methodology is comprised of the following major steps: (1) dataset selection, (2) data normalization, (3) data resampling, (4) modeling, and (5) evaluation. The data has been pre-processed to obtain effective results. Various resampling techniques have been used to tackle data imbalance issues to get the best

Table 1: A summary of techniques cast-off in related articles.

Reference	Year	ML techniques	Undersampling techniques	Oversampling techniques
[39]	2021	✓	✗	✗
[40]	2021	✓	✗	✗
[41]	2021	✓	✗	✗
[33]	2020	✓	✗	✗
[31]	2019	✓	✗	✗
[34]	2019	✓	✗	✗
[36]	2018	✓	✗	✗
[37]	2018	✓	✗	✗
[11]	2018	✓	✗	✗
[35]	2018	✓	✗	✗
[32]	2018	✓	✗	✗
Proposed model	**2022**	✓	✓	✓

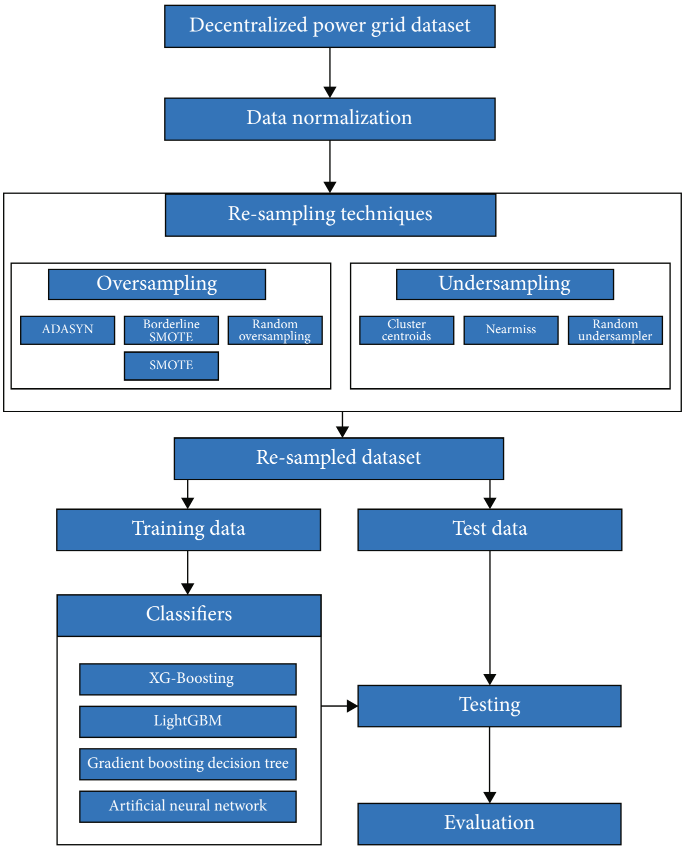

Figure 1: Proposed methodology.

results. After data preprocessing, different ML models were used, and their results were compared.

3.1. Dataset. In this study, the Electricity Grid Simulated Dataset was employed and obtained from the UCI repository. The dataset includes 10,000 instances and 14 features in which 3620 were stable and 6380 were unstable. The dataset has also observed class imbalance issues because unstable instances are far more than stable instances. The dataset contains the reaction time of participants (tau1, tau2, tau3, tau4), nominal power produced/consumed (p1, p2, p3, p4), and gamma coefficient, i.e., price elasticity features (g1, g2, g3, g4) as shown in Table 2. Price elasticity is the measurement of the change in consumption of electricity in response to a change in its price, expressed mathematically as

$$\text{Formula to calculate price elasticity} = \frac{\%\text{Change in electricity quantity demanded}}{\%\text{Change in price}}. \quad (1)$$

Table 2: Feature description used in the dataset.

Feature name	Feature description
tau1	Electricity producer's reaction time in response to price change
tau2	First consumers' reaction time in response to price change
tau3	Second consumer's reaction time in response to price change
tau4	Third consumer's reaction time in response to price change
p1	Nominal power (positive real) produced by the producer (amount of electricity produced)
p2	Nominal power (negative real) consumed by the first consumer (amount of electricity consumed by the first consumer)
p3	Nominal power (negative real) consumed by the second consumer (amount of electricity consumed by the second consumer)
p4	Nominal power (negative real) consumed by the third consumer (amount of electricity consumed by the third consumer)
g1	(Gamma coefficient) proportional to price elasticity of producer
g2	(Gamma coefficient) proportional to the price elasticity of the first consumer
g3	(Gamma coefficient) proportional to the price elasticity of the second consumer
g4	(Gamma coefficient) proportional to the price elasticity of the third consumer
Stab	Target class value real, positive shows instability or negative shows stability
Stabf	Target class label (categorical), i.e., stable or unstable

A numerical optimization problem known as the characteristics roots equation determines the target class value (stab). Positive real numbers indicated instability, and negative real numbers showed the grid's stability. The stability/instability of the system is labeled as stabf (categorical).

3.2. Data normalization. The major issue with various features is that each numerical feature/attribute is represented differently. So, data normalization is an effective data preprocessing technique for tabular data to make comparisons between measurements more accessible while constructing a model. It rescales feature values to confront the standard normal distribution to form new inputs. The maximum and minimum values of various features often vary significantly, like the reaction time of different participants ranges from 0.5 to 0.99, power values of producers range from 1.58 to 5.86, power values of different consumers range from -1.99 to -0.5, and gamma values, i.e., price elasticity of demand of all participants ranges from 0.05 to 0.99. The target variable values resulted from optimizing characteristics roots equations, ranging from -0.08 to 0.10. The Z-score normalization technique has been employed to bring all these features into a specified range. All numerical values have been scaled within the specified range (-1.73 to +1.73). The formula of the Z-score technique is given in equation (2).

$$z = \frac{X - \bar{X}}{S}, \tag{2}$$

where z is the standard score, S is the standard deviation of a sample, X represents each value in the dataset, and $\bar{X}$ is the mean of all values in the dataset. It has been observed that Z-score normalization outperformed other data normalization techniques [42–44].

3.3. Undersampling Techniques. Undersampling is one of the simplest methods for dealing with imbalanced data. This technique undersamples the majority class to balance it with the minority class [45]. The undersampling method can be applied if a sufficient amount of data is collected. This study used three undersampling techniques: near miss, cluster centroid, and random undersampling.

3.3.1. Random Undersampling. This approach is aimed at picking and removing samples from the majority class randomly. Hence, the number of majority class examples is reduced. Due to undersampling, the data set is transformed, with fewer examples in the majority class. This procedure can be replicated until each class has an equal number of examples. This approach proves effective when the minority class has sufficient examples despite the great imbalance. On the other hand, it is always important to consider the risk of losing important data. We remove them randomly from our dataset because we have no means of identifying or maintaining the examples in the majority class rich in information. This method also significantly reduces the size of the training data. As a consequence, it is the most naive method of data undersampling.

3.3.2. Near Miss. This technique was proposed by Mani and Zhang [46] to achieve an equal distribution of classes by removing majority class examples at random. Only the majority class examples in the overlapping region nearest to the minority class examples are retained. When instances of two distinct classes are too close to one another, the majority of class instances are removed to increase the space between these two classes, and this aids in the process of classification. The first process, "NearMiss-1," picks samples from the majority class closest to those from the minority class. This process picks majority class samples with the smallest average distances to the three nearest minority class samples. The second "NearMiss-2" approach picks majority class samples with the smallest average distances to the three farthest minority class samples. For each minority class sample, the third process, "NearMiss-3," takes a fixed number of the nearest majority class samples. Finally, the fourth method, "Most distant" approach, chooses the majority class samples with the highest average distances to the three nearest minority class samples.

3.3.3. Cluster Centroid. One of the main disadvantages of undersampling is that valuable information from the majority class may be lost, resulting in misclassifying samples after classification. This cannot be afforded to build a solid model. As a result, the cluster centroid method was proposed by Yen and Lee [47] to solve this problem. This technique undersamples the majority class instances by replacing majority class instances from clusters with a cluster of centroids by considering the ratio of majority class samples to minority class samples. Undersampling is accomplished using this approach, generating centroids using k-means clustering methods. The data have been grouped based on similarity. The data is equipped with a K-means algorithm, and the level of undersampling determines the number of clusters (k). The sets of cluster centroids from K-means then entirely substitutes for most cluster samples. The most representative combinations of the majority class will be visualized in the middle of the cluster of centroids. This problem was attempted to be solved by under fitting and overfitting the data and their combination. While under fitting the dataset, only the cluster centroids were considered, adapted from [47].

3.4. Oversampling Techniques. When the number of instances in each class is not equal, any dataset can be called imbalanced. Resampling methods add a bias typically to make the dataset balanced. While classifiers may learn from imbalanced datasets, balanced datasets have more efficient results. All resampling techniques resample data until it reaches the required ratio. It also helps compare various resampling methods in the final training set for a given proportion of majority and minority class data points. The data level resampling method (oversampling and undersampling) is the optimal solution for dealing with class imbalance problems. In this study, various resampling techniques have been employed. By replicating or making new minority class samples, oversampling increases the minority class weight. Different oversampling methods are available in the literature. [48]. Four oversampling techniques have been applied in this study: random oversampling (ROS), adaptive synthetic (ADASYN), synthetic minority oversampling technique (SMOTE), and borderline-SMOTE.

3.4.1. Random Oversampling. ROS replicates minority classes randomly, that replication is content identical to the source, and no new contextual variation is added. To balance the classes, this approach creates a minority class set based on the size of the majority class. Overfitting and increased training time may occur when all the training examples are very close, and the classifier correctly classifies them. If a testing example differs marginally from the training examples, the classifier would be unable to correctly classify it, resulting in misclassification for the new examples. This technique is prone to overfitting, but other oversampling methods are also developed based on ROS.

3.4.2. Synthetic Minority Oversampling Technique. Compared to ROS, the SMOTE is a more advanced approach. Chawla et al. [48] state that it oversamples data by generating synthetic examples. New minority instances are synthesized between existing minority instances in SMOTE. It selects the minority class at random and calculates the K -nearest neighbor for that specific point. Finally, it adds synthetic points between that chosen point and its neighbors. The instance of the x_i minority class is chosen as the foundation for creating new synthetic data points. Several nearest neighbors of the same class are selected from the training set based on a distance metric. Finally, a randomized interpolation has been carried out to obtain new instances. An integer value of oversampling total amount N is determined, which can be set up to achieve a 1 : 1 class distribution or discovered using a wrapper method [49]. After that, a multistep iterative process is carried out, which works as below: first, a minority class instance is chosen randomly from the training set. Next, the K-nearest neighbors are then collected. Finally, N of these K instances are randomly selected to calculate new instances via interpolation. To complete this task, the difference between the feature vector (sample) under consideration and each of the selected neighbors is taken. After that, a random number between 0 and 1 is multiplied by the difference and added to the previous feature vector. As a result, a random point along the "line segment" between the features is selected. If there is a case of nominal attributes, one of the two values is chosen randomly.

Consider the sample (6, 4) and its nearest neighbour (4, 3). The sample for which k-nearest neighbors are being identified is (6, 4), and one of its k-nearest neighbors is (4, 3).

$$\begin{aligned} \text{Let} : a_{11} = 6, a_{21} = 4, a_{21} - a_{11} = -2, \\ a_{12} = 4, a_{22} = 3, a_{22} - a_{12} = -1. \end{aligned} \tag{3}$$

The newly generated samples will be as given in equation (4). ran $d(0 - 1)$ creates two random numbers vectors ranging from 0 to 1.

$$a_1', a_2' = (6, 4) + \text{ran } d(0 - 1) \times (-2,-1). \tag{4}$$

3.4.3. Adaptive Synthetic. ADASYN works similar to SMOTE. However, weighting the distance and the linear function creates more minority class synthetic examples. According to He et al. [50], ADASYN concentrates on minority class examples based on their level of learning difficulty. The basic idea behind ADASYN is to use a weighted distribution as a criterion for various minority classes to determine how many synthetic samples are required for each minority class [50]. Equation Equation (5) is used to calculate the weight.

$$W = \frac{\Delta}{K}. \tag{5}$$

The value of W varies between 0 to 1, and Δ is the number of examples in a minority class's K-nearest neighbors, which are members of the majority class.

3.4.4. Borderline-SMOTE. Borderline-SMOTE is a more advanced type of SMOTE that aims to generate synthetic samples by interpolating the k-nearest neighbors of the minority instances close to the border [51]. Since these border instances are more relevant for classification, this technique only extracts synthetic instances for minority samples near the boundary of two classes. At the same time, SMOTE produces new instances for each minority sample. Possible misclassified minority class instances will undergo more training in borderline-SMOTE [52]. It first identifies borderline minority instances; then, it uses them to generate synthetic instances with their chosen k-nearest neighbors. Rather than simply replicating the existing samples, SMOTE creates new synthetic samples along the line between the minority samples and their selected nearest neighbors. However, this increases overlapping between classes since synthetic samples are generated without considering neighboring samples. Many modified techniques have been proposed to solve this constraint, with borderline-SMOTE proving the most effective in most cases. Since samples close or on the borderline are more likely to be misclassified than those farther away, borderline-SMOTE just oversamples and enhances these difficult-to-learn samples.

For clear presentation, considering a given training dataset D, we define subsets $D_{\min} \subset D$ and $D_{\text{maj}} \subset D$, where $D_{\min}$ and D_{maj} are the set of minority and majority class samples in D, and the set of m nearest neighbors are determined for each $x_i \in D_{\min}$ called $D_{i:m-\text{NN}}$. Then, calculate how many of x_i nearest neighbors are members of the majority class, i.e., $|D_{i:m-\text{NN}} \cap D_{maj}|$.Then, select the number of neighbours of x_i to form the set "DANGER", which satisfies

$$\frac{m}{2} \leq \left|D_{i:m-\text{NN}} \cap D_{\text{maj}}\right| < m. \tag{6}$$

Only those in "DANGER" having more majority class neighbors than minority class neighbors are suggested in equation (6). This means that they represent samples from the borderline minority classes, which are the most likely to be misclassified. It is worth noting that other x_i are not operating in the next step. Finally, for each sample x_j in the "DANGER" set, select one of the K-nearest neighbors of x_j at random that have the smallest Euclidian distance to it, multiply the corresponding feature vector difference with a random number between [0,1], and this vector difference is added to x_j

$$x_{\text{new}} = x_j + \delta \times \left(\hat{x}_j - x_j\right), \tag{7}$$

where x_j is the selected minority samples in the "DANGER" set, $\hat{x} \in D_{\min}$ is one of $x_j{}'sK$-nearest neighbors, and $\delta \in [0, 1]$ is a random number. The resulting synthetic sample is, therefore, one point in the line segment between x_j and $\hat{x}_j$ according to equation (7). The newly generated samples are appended to the original set and used to train the classifier. The dataset class distribution before and after applying resampling techniques is presented in Table 3.

3.5. XGBoost. It is a GBDT extension introduced by Chen and Guestrin [53]. It is a boosting algorithm and belongs to the supervised learning algorithms. Boosting is an ensemble technique of sequential learning. In boosting, different models are trained one after another. XGBoosting first creates a base model. We take the average number as the first prediction of the base model (also called model zero M0). Next, model M1 is fitted to minimize errors (the difference between the actual and predicted values). Until this, the procedure is the same as gradient boosting. XGBoost uses regularization parameters to avoid overfitting, and it also uses auto pruning to avoid trees not growing beyond a certain level and handles missing values. It has been used to solve various classification problems in many fields. The XGBoost algorithm assigns various levels of importance to features before deciding the weighted distance for the K-means algorithm. It combines predictions from "weak" classifiers (tree model) to get a "strong" classifier (tree model). It speeds up the learning process allowing for quicker modeling using distributed and parallel computing.

A new tree is generated along the direction of the negative gradient of the loss function. As the number of tree models increases, the loss becomes smaller and smaller. The XGBoost computational process started from equation (8):

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{t-1} + f_t(x_i), \tag{8}$$

where $\hat{\mathrm{y}}_{\mathrm{i}}^{\mathrm{t}-1}$ is the previously generated tree model, $f_t(x_i)$ is the newly generated tree model, $\hat{y}_i^{(t)}$ is the final tree model, and t is the total number of base tree models. Both depths of the tree and the number of trees are essential parameters for the XGBoost algorithm. The problem of determining the best algorithm was changed to finding a new classifier capable of reducing the loss function, with the target loss function given in equation (9).

$$\text{obj}^{(t)} = \sum_{i=1}^{t} (L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_i), \tag{9}$$

where y_i is the actual value; $\hat{y}_i^{(t)}$ is the predicted value; $L(y_i, \hat{y}_i^{(t)})$ is the loss function, and $\Omega(f_i)$ is the regularization term.

Equation (10) could be obtained by substituting equation (8) into equation (9) and then following some deduction steps.

$$\text{obj}^{(t)} = \sum_{i=1}^{t} L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i) + \Omega(f_t) + \text{constant}. \tag{10}$$

Table 3: Dataset class distribution: before and after applying resampling techniques.

	Electricity Grid Simulated Dataset		
	Total instances	Stable instances	Instable instances
Imbalanced dataset			
Imbalanced dataset	10000	3620	6380
Oversampled dataset			
ROS	12760	6380	6380
ADASYN	12760	6380	6380
SMOTE	12760	6380	6380
Borderline-SMOTE	12760	6380	6380
Undersampled data			
Near miss	7240	3620	3620
Cluster centroids	7240	3620	3620
Random undersampling	7240	3620	3620

After that, the final target loss function was transformed to equation (11), used to train the model.

$$obj^{(t)} = \sum_{i=1}^{t} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t), \tag{11}$$

where $g_i = \partial_{y_i(t-1)} l(y_i, y_i^{(t-1)})$ and $h_i = \partial^2_{y_i(t-1)} l(y_i, y_i^{(t-1)})$ are the loss function's first and second-order gradient statistics. Equation (12) calculates the regularization term $\Omega(f_t)$ to reduce the model's complexity and increase its applicability to other datasets.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2, \tag{12}$$

where λ and γ are coefficients with default values set as $\lambda = 1, \gamma = 0$, ω is the weight of the leaves, and T is the number of leaves. Both continuous and discrete variables can be used as inputs to the XGBoost algorithm, but the output variable must be discrete, excluding binary variables.

4. Evaluation Metrics

In every predictive modelling task, model evaluation is critical. It becomes more critical in predictive ensemble modeling, where diversity and models' relative performance must be evaluated thoroughly. Each of the evaluation metrics is based on one of 4 classifications. These classifications are true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). With the aid of the confusion matrix, accuracy is typically used to measure the efficiency of a model [54, 55]. Equation (13) has been used to calculate the model's accuracy.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{13}$$

Precision is about out of the total predicted positives, how many of them are true positives. It means precision measures how many positive instances the classifier said were positive. The model's precision has been calculated using equation (14).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{14}$$

A recall is about out of the total actual positives, how many of them are true positives. Equation Equation (15) has been used to calculate the model's recall.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{15}$$

F-measure is the harmonic mean of precision and recall. Precision and recall are mutually exclusive: Low recall is usually associated with higher precision. Equation Equation (16) has been used to calculate the model's F-measure.

$$F - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{Precision} + \text{recall}}. \tag{16}$$

ROC curve plot is another commonly used metric for evaluating a classifier's efficiency. ROC graph shows the performance of a classification model at all classification thresholds. It plots the false positive rate (x-axis) versus the true positive rate (y-axis) for different candidate threshold values between 0.0 and 1.0. It is used to interpret the prediction of probabilities for binary classification problems.

5. Results

Various classifiers have been employed to evaluate the performance of an imbalanced Electricity Grid Simulated Dataset, as shown in Table 4. The train-test split technique has been used to assess the performance of ML algorithms. The dataset has been split into two subsets, with 70% for training and 30% for testing purposes. The first subset is the training dataset used to fit the model. The second subset is not used to train the model; instead, the model is provided with the dataset's input element. Then, predictions are made, and the results are compared to the expected results.

Table 4: Performance of ML models on imbalanced dataset.

Models	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	ROC (%)
ANN	93.4	97.7	92	94.8	98.7
Averaged perceptron	81.5	84.6	87.5	86	88.9
Bayes point machine	81.3	83.2	89.2	86.1	88.7
Decision forest	89.8	91.9	92.4	92.2	95
Decision jungle	89.5	89.7	94.8	92.2	96
GBDT	94.1	95.2	95.8	95.5	98.8
LightGBM	94.6	94.6	97	95.8	98.9
Locally deep SVM	91.9	93.6	93.9	93.8	97.4
LR	81.5	84.4	87.7	86	89
SVM	80.9	84.3	86.6	85.5	88.5
XGBoost	94.7	94.9	96.9	95.9	98.9

This second dataset is called the test dataset. The performance of the XGBoost method has also been compared with other ML models. The XGBoost outperformed other ML models on an imbalanced dataset. XGBoost algorithm was run in Jupyter Notebook. Table 4 shows the results, with the XGBoost model predicting the highest accuracy of 94.7% on the imbalanced Electricity Grid Simulated Dataset. Tuning parameters is a critical step in improving the efficiency of any ML algorithm. It involves determining a grid with all possible parameters and checking them to find the values that maximize classification performance. The default values are used for the parameters whose values are not defined. Various parameters are listed below:

(i) Eta: It is the learning rate of the model. The default value of eta is 0.3; however, the optimal value of eta used in our experiment is 0.4. The feature weights are shrunk by eta to make the boosting procedure more prudent. Its range is from 0 to 1

(ii) Subsample: It controls the number of samples (observations) supplied to a tree. The default value of the subsample is 1. The optimal value used in our experiment is 0.8. Its range is from 0 to 1

(iii) colsample_bytree: It controls the number of features (variables) supplied to a tree. Its default value is 1, ranging from 0 to 1. The optimal value used in our experiment is 0.9

(iv) N-estimator: The number of trees (or rounds) in an XGBoost model is defined in the n_estimators. 100 is the default value for the n-estimator. Its range is from 1 to infinity. The optimal value used in our experiment is 200

To assess the performance of machine learning algorithms, various classifiers were used on undersampled datasets, as shown in Table 5. ANN proved to be the best algorithm. Classifiers were applied to three undersampled datasets. The results are shown in Table 5, in which random undersampling in combination with ANN has predicted the best accuracy of 94.5%. The random undersampling technique outperformed all other undersampling techniques based on accuracy. Other models have also shown effective results on the undersampled dataset. Further, the performance of classifiers was also better on the cluster centroids based on undersampled datasets than near miss.

Oversampling techniques (ADASYN, borderline-SMOTE, ROS, and SMOTE) have also been employed to improve the performance of the classifiers. Various models were applied to oversampled datasets, as shown in Table 6. The results are presented in Table 6, in which the ROS method in combination with XGBoost has predicted the best accuracy of 96.8%. XGBoost, in combination with borderline-SMOTE and SMOTE, has shown promising results with an accuracy of 96.5% and 96.1%, respectively. ADASYN predicted an outcome of 95.9% with the GBDT. The results showed that the XGBoost outperformed other ML models based on imbalanced and oversampled datasets. XGBoost was tuned further by adjusting the values of a few parameters to improve the results. However, ANN proved to be best on an undersampled dataset. The results of the imbalanced dataset have also been compared with undersampling and oversampling techniques. Results showed that ROS-based methods outperformed all other oversampling and undersampling techniques used—the performance of the XGBoost model, along with other models. The proposed model has outperformed previous studies significantly, as shown in Table 7.

The XGBoost, in combination with ROS, outperformed all other models predicting the best accuracy of 96.8%. However, the accuracy was 94.7% on an imbalanced dataset, as shown in Figure 2. The F-measure and ROC have also shown better results with ROS as 96.7% and 99.6%, respectively, compared to F-measure and ROC on the imbalanced dataset, which showed 95.6% and 98.9%, respectively.

6. Discussion

Effective demand response management and control in decentralized power grids are complex. Because grid participants' consumption and production behaviors are influenced by price signals issued and responded on a seconds scale, key variables influencing the grid's stability are the volumes of electricity consumed/produced (p) by each grid

Table 5: Performance of ML models on an undersampled dataset.

Undersampling technique	Models	Accuracy (%)	Precision (%)	Recall (%)	*F*-measure (%)	ROC
Cluster centroids	ANN	94.4	94.6	94.1	94.4	98.7
	Averaged perceptron	79.1	79.5	78.6	79	88.1
	Bayes point machine	79.1	79.4	78.7	79	88.1
	Decision forest	91.1	93.3	88.6	90.9	97.1
	Decision jungle	88.9	88.8	89.1	88.9	95.8
	GBDT	94.2	94.4	94.1	94.2	98.9
	LightGBM	93.6	94.2	92.8	93.5	98.4
	Locally deep SVM	90.9	91.1	90.6	90.9	97.2
	LR	79.1	79.6	78.4	79	88.1
	SVM	78.5	78.9	77.9	78.4	87.5
	XGBoost	93.7	94.1	93.3	93.7	98.7
Near miss	ANN	92	73.2	89.8	91.9	98.1
	Averaged perceptron	75.2	74	77.9	75.9	83.1
	Bayes point machine	75.1	74.1	77.5	75.7	83
	Decision forest	91.1	93.3	88.6	90.9	97.1
	Decision jungle	85.3	83.7	87.7	85.7	93.1
	GBDT	92.1	90.6	94	92.3	98.3
	LightGBM	91.8	90.6	93.2	91.9	97.9
	Locally deep SVM	88.3	88.3	88.3	88.3	94.6
	LR	75.2	73.9	78	75.9	83
	SVM	74.6	73.2	77.6	75.3	82.3
	XGBoost	92.6	92.1	93.1	92.6	98.8
Random undersampling	ANN	94.5	97.8	91	94.3	99
	Averaged perceptron	80.2	80.1	80.4	80.3	88.9
	Bayes point machine	79.7	79.7	79.9	79.8	88.8
	Decision forest	89	89.1	89	89	95.9
	Decision jungle	89.2	87.8	91.1	89.4	95.9
	GBDT	94.1	94.5	93.7	94.1	98.8
	LightGBM	92.7	93.1	92.1	92.6	98.3
	Locally deep SVM	90	89.8	90.2	90	96.6
	LR	79.8	79.7	80	79.9	88.7
	SVM	80.3	80	80.9	80.4	89
	XGBoost	93.4	93.6	93.1	93.3	98.7

participant and the cost-sensitivity (g), i.e., price elasticity and reaction time (tau) to price signals of the grid participants. A simulation of a decentralized grid applied to a four-node star grid.

In combination with resampling techniques, ML techniques have been used in our study to improve the decentralized grid stability prediction. The simulated four-star node dataset used in this study reflects a simple configuration of a decentralized grid. Here in this study, the grid participants are four and work have been performed on a four-node star architecture grid. The twelve independent variables have been imposed constraints of maximum and minimum values, and an absolute value of power production and consumption has been taken for simulation. As designed, this method successfully explores the feasible solution space with an evaluation of 10,000 cases. However, there are likely stronger correlations between the grid participants in decentralized grids. It has been discovered that the classifier's accuracy is related to class balance. The accuracy increases when the number of minority samples increases in the dataset. As the number of instances increases, the classifier has a greater chance of learning the patterns that differentiate binary classes. The XGBoost model predicted the best accuracy with the ROS method (96.8%), followed by XGBoost with borderline-SMOTE (96.5%). Other combinations predicted slightly lesser accuracy, such as XGBoost with SMOTE-oversampling gave 96.1%, GDBT with ADASYN-oversampling 95.9%, and XGBoost with imbalanced dataset 94.7%. As shown in Table 7, our study outperformed other studies. This can help to avoid power outages and improve grid performance significantly.

Table 6: Performance of ML models on oversample dataset.

Oversampling technique	Models	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	ROC (%)
ADASYN	ANN	95.4	97.5	93.4	95.4	99
	Averaged perceptron	79.2	78.6	81.3	79.9	87.7
	Bayes point machine	79.1	78.4	81.3	79.8	87.8
	Decision forest	91.2	93.7	88.8	91.2	97.2
	Decision jungle	89.5	89.7	89.7	89.7	96.4
	GBDT	95.9	96.5	95.4	96	99.4
	LightGBM	94.7	96.5	94	94.8	99.1
	Locally deep SVM	91.5	91.7	91.6	91.6	97.4
	LR	79.1	78.3	81.5	79.9	87.8
	SVM	78.8	78.5	80.6	79.5	87.6
	XGBoost	95.7	96.4	95	95.7	99.4
Borderline-SMOTE	ANN	95.1	95.8	94.4	95.1	99
	Averaged perceptron	76.9	77.5	75.9	76.7	86.1
	Bayes point machine	76.9	77.4	76.2	76.8	86.1
	Decision forest	90.8	93.5	87.6	90.5	96.8
	Decision jungle	89.3	91.2	87.1	89.1	96
	GBDT	95.4	96.9	93.7	95.3	99.3
	LightGBM	94.5	95.9	92.7	94.3	99
	Locally deep SVM	90.8	92.4	88.8	90.6	97.2
	LR	76.8	77.4	75.8	76.6	86.1
	SVM	76.8	77.6	75.6	76.6	85.9
	XGBoost	96.5	97.7	95.1	96.4	99.5
ROS	ANN	95.7	97.9	93.4	95.6	99.2
	Averaged perceptron	79.2	79.4	78.9	79.2	88.3
	Bayes point machine	79.4	79.7	79.2	79.4	88.4
	Decision forest	91.8	94.9	88.4	91.6	97.7
	Decision jungle	90.5	92	88.8	90.4	97
	GBDT	95.7	96.8	94.4	95.6	99.4
	LightGBM	95.7	97.2	93.9	95.5	99.4
	Locally deep SVM	90.8	92.4	88.8	90.6	97.2
	LR	79.4	79.5	79.3	79.4	88.4
	SVM	78.8	79.1	78.4	78.8	88.1
	XGBoost	96.8	97.5	95.9	96.7	99.6
SMOTE	ANN	95.6	97.8	93.4	95.5	99.3
	Averaged perceptron	80.1	80.7	79.3	80	89
	Bayes point machine	91.1	933	88.6	90.9	97.1
	Decision forest	95.9	97	94.7	95.8	99.3
	Decision jungle	90.4	90.3	90.7	90.5	96.5
	GBDT	80.3	80.9	79.5	80.2	89
	LightGBM	90	90.8	89.2	90	96.8
	Locally deep SVM	94.9	95.8	93.6	94.7	99.2
	LR	80.2	80.8	79.4	80.1	89
	SVM	79.8	80.6	78.5	79.6	88.9
	XGBoost	96.1	96.5	95.5	96	99.4

The different classification techniques on Electricity Grid Dataset showed different performances to identify improvement in grid stability. XGBoost has an accuracy of 94.7%, but when combining XGBoost with ROS, its accuracy improved to 96.8%. LightGBM has an accuracy of 94.6%, but when combined with ROS, its accuracy improved to 95.7%. Similarly, ANN has an accuracy of 93.4%, but when combining it with ROS, its accuracy improved to 95.7%.

Table 7: Grid stability prediction: a comparison of previous work.

Reference	Year	Dataset	Number of features	Number of instances	Results (accuracy)
[33]	2020	New York Independent System Operator Data	9	2832	Enhanced MLP = 65.01% Enhanced SVM = 70.40% Enhanced LR = 67.78%
[31]	2019	Electrical Grid Stability (Simulated Dataset)	14	10,000	KRR − XGBoost hybrid = 91.40%
[34]	2019	Electrical Grid Stability (Simulated Dataset)	14	10,000	MLP = 93.8% GBDT = 88.8 LR = 66.3% RF = 87.8%
[36]	2018	Australian PV Solar Power Data	12	3828	ANN = 85% NB = 83% LDA = 86%
[37]	2018	Simulated Data generated through Spider IDE	4	3000	Ensemble diverse ELM = 94.13%
[11]	2018	Electrical Grid Stability (Simulated Dataset)	14	10,000	DT = 80%
[35]	2018	Dynamic simulation-based power system data	5	10147	RF = 90.01% ANN = 89.73% SVM = 86.7%

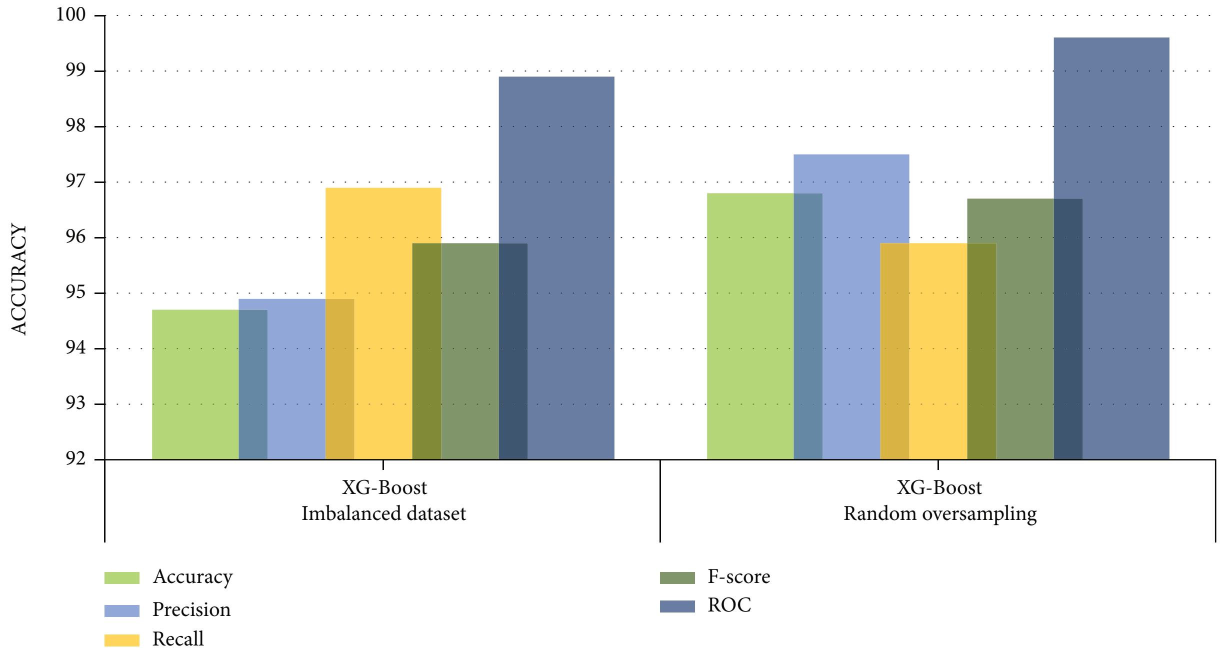

Figure 2: The performance comparison of the proposed framework on imbalance and oversample dataset.

The fourth algorithm, GBDT, gave an accuracy of 94.1%, but while combining it with SMOTE-oversampling and ADASYN-oversampling, its results were improved to 95.9%, as shown in Table 6.

On the contrary, undersampling techniques predicted less accuracy than the oversampling and imbalanced dataset. The ANN with random undersampling gave results of 94.5%, ANN with cluster centroid undersampling gave 94.4%, and XGBoost with near miss undersampling has given results of 92.6%, as shown in Table 5. This model may help better predict the stability of a decentralized electricity grid, which may help better distribute and manage electricity. To test the effectiveness of our proposed technique, it is verified using the Electricity Grid Dataset. It can be used to evaluate any real-time dataset. LightGBM has never been used to predict grid stability to our knowledge. Rather than employing algorithmic level data resampling techniques, only data level resampling techniques have been employed in this research, limiting our work. A detailed comparison of different studies has been shown in Table 7 below for grid stability prediction.

7. Conclusion

A stable power grid is necessary to overcome power outages and a constant electricity supply. There must be a balance between power supply and demand for a grid to remain stable. Due to the volatile nature of renewable energy resources, the decentralized grid often destabilizes. Various ML algorithms were applied in this study to predict the stability of a decentralized power grid. Key input variables influencing the grid stability were the power production or consumption by the grid participants (p), price elasticity, i.e., the cost sensitivity of the grid participants (g) and participants' reaction time against price changes (tau). The simulated data from the UCI machine learning repository predicts the decentralized grid stability. Balanced data predicts better results than imbalanced data; so, different resampling techniques have been used to address the class imbalance issue and obtain better results. Four oversampling techniques (ROS, ADASYN, SMOTE, and borderline-SMOTE) and three undersampling techniques (random undersampling, near miss, and cluster centroid) were used to balance the class distribution in the dataset. After preprocessing the data, different ML models were used for results prediction, and the results were compared. Oversampling techniques predicted the best results in our experiments.

In contrast, undersampling techniques were less accurate than oversampling and imbalanced datasets. It may also imply that if the number of instances increases, the classifier has a greater chance of learning the patterns that differentiate binary classes. The XGBoost algorithm outperformed all other ML algorithms to predict the decentralized electricity grid. XGBoost predicted accuracy of 94.7%, but while combining it with ROS, its accuracy prediction was improved to 96.8%. To boost the model's accuracy, four tuning parameters were applied. More complex decentralized grids can also be explored with more than four grid participants involving multiple prosumers and different grid architectures, i.e., circular and multibranched configurations, to explore the proposed model's performance further.

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request

Conflicts of Interest

The authors declare that they have no conflicts of interest

Acknowledgments

Princess Nourah Bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R54), Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia.

References

[1] M. Usman and M. Radulescu, "Examining the role of nuclear and renewable energy in reducing carbon footprint: does the role of technological innovation really create some difference?," *Science of The Total Environment*, vol. 841, article 156662, 2022.

[2] T. T. Teo, T. Logenthiran, and W. L. Woo, "Forecasting of photovoltaic power using extreme learning machine," in *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*,, pp. 1–6, Bangkok, Thailand, 2015.

[3] B. Schäfer, M. Matthiae, M. Timme, and D. Witthaut, "Decentral smart grid control," *New Journal of Physics*, vol. 17, no. 1, article 015002, 2015.

[4] International Renewable Energy Agency, "A renewable energy roadmap 2030," 2021, http://www.irena.org/-/media/Files/IRENA/Agency/Publication/2014/IRENA_REmap_summary_findings_2014.pdf.

[5] Alternate Energy Development Board Government of Pakistan, "Alternate and renewable energy policy," 2019, http://www.aedb.org/images/Draft_ARE_Policy_2019_-_Version_2_July_21_2019.pdf.

[6] M. Younas, S. Shafique, A. Hafeez, F. Javed, and F. Rehman, "An overview of hydrogen production: current status, potential, and challenges," *Fuel*, vol. 316, article 123317, 2022.

[7] B. Schäfer, C. Grabow, S. Auer, J. Kurths, D. Witthaut, and M. Timme, "Taming instabilities in power grid networks by decentralized control," *The European Physical Journal Special Topics*, vol. 225, no. 3, pp. 569–582, 2016.

[8] A. Ali, D. Raisz, and K. Mahmoud, "Voltage fluctuation smoothing in distribution systems with RES considering degradation and charging plan of EV batteries," *Electric Power Systems Research*, vol. 176, article 105933, 2019.

[9] M. Khushi, K. Shaukat, T. M. Alam et al., "A comparative performance analysis of data resampling methods on imbalance medical data," *IEEE Access*, vol. 9, pp. 109960–109975, 2021.

[10] B. N. Alhasnawi, B. H. Jasim, B. E. Sedhom, E. Hossain, and J. M. Guerrero, "A new decentralized control strategy of microgrids in the internet of energy paradigm," *Energies*, vol. 14, no. 8, p. 2183, 2021.

[11] V. Arzamasov, K. Böhm, and P. Jochem, "Towards concise models of grid stability," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, Aalborg, Denmark, 2018.

[12] M. Timme, L. Kocarev, and D. Witthaut, "Focus on networks, energy and the economy," *New Journal of Physics*, vol. 17, no. 11, article 110201, 2015.

[13] A. Wahab, T. M. Alam, and M. M. Raza, "Usability evaluation of FinTech mobile applications: a statistical approach," in *2021 International Conference on Innovative Computing (ICIC)*, pp. 1–10, Lahore, Pakistan, 2021.

[14] K. Shaukat, S. Luo, N. Abbas, T. Mahboob Alam, M. Ehtesham Tahir, and I. A. Hameed, "An analysis of blessed Friday sale at

a retail store using classification models," in *2021 The 4th International Conference on Software Engineering and Information Management*, pp. 193–198, Yokohama Japan, 2021.

[15] A. Tariq, M. J. Awan, J. Alshudukhi, T. M. Alam, K. T. Alhamazani, and Z. Meraf, "Software measurement by using artificial intelligence," *Journal of Nanomaterials*, vol. 2022, 10 pages, 2022.

[16] T. I. Baig, T. M. Alam, T. Anjum et al., "Classification of human face: Asian and non-Asian people," in *2019 International Conference on Innovative Computing (ICIC)*, pp. 1–6, Lahore, Pakistan, 2019.

[17] K. Shaukat, T. M. Alam, S. Luo et al., "A review of time-series anomaly detection techniques: a step to future perspectives," in *Advances in Information and Communication. FICC 2021*, K. Arai, Ed., vol. 1363 of Advances in Intelligent Systems and Computing, pp. 865–877, Springer, Cham, 2021.

[18] T. M. Alam, K. Shaukat, A. Khelifi et al., "A fuzzy inference-based decision support system for disease diagnosis," *The Computer Journal*.

[19] T. M. Alam, K. Shaukat, A. Khelifi et al., "Disease diagnosis system using IoT empowered with fuzzy inference system," *Computers, Materials and Continua*, vol. 70, no. 3, pp. 5305–5319, 2022.

[20] K. Shaukat, T. M. Alam, M. Ahmed et al., "A model to enhance governance issues through opinion extraction," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 511–516, Vancouver, BC, Canada, 2020.

[21] T. M. Alam and M. J. Awan, "Domain analysis of information extraction techniques," *International Journal Of Multidisciplinary Sciences And Engineering*, vol. 9, no. 6.

[22] Y. Ali, A. Farooq, T. M. Alam, M. S. Farooq, M. J. Awan, and T. I. Baig, "Detection of schistosomiasis factors using association rule mining," *IEEE Access*, vol. 7, pp. 186108–186114, 2019.

[23] T. I. Baig, Y. D. Khan, T. M. Alam, B. Biswal, H. Aljuaid, and D. Q. Gillani, "Ilipo-pseaac: identification of lipoylation sites using statistical moments and general pseaac," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 215–230, 2022.

[24] K. Shaukat, T. M. Alam, I. A. Hameed et al., "A comprehensive dataset for bibliometric analysis of SARS and coronavirus impact on social sciences," *Data in Brief*, vol. 33, article 106520, 2020.

[25] M. Z. Latif, K. Shaukat, S. Luo, I. A. Hameed, F. Iqbal, and T. M. Alam, "Risk factors identification of malignant mesothelioma: a data mining based approach," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–6, Istanbul, Turkey, 2020.

[26] S. Farah, N. Humaira, Z. Aneela, and E. Steffen, "Short-term multi-hour ahead country-wide wind power prediction for Germany using gated recurrent unit deep learning," *Renewable and Sustainable Energy Reviews*, vol. 167, article 112700, 2022.

[27] A. K. Bashir, S. Khan, B. Prabadevi et al., "Comparative analysis of machine learning algorithms for prediction of smart grid stability†," *International Transactions on Electrical Energy Systems*, vol. 31, no. 9, article e12706, 2021.

[28] A. Ali, K. Mahmoud, and M. Lehtonen, "Optimal planning of inverter-based renewable energy sources towards autonomous microgrids accommodating electric vehicle charging stations," *IET Generation, Transmission & Distribution*, vol. 16, no. 2, pp. 219–232, 2022.

[29] A. Ali, K. Mahmoud, and M. Lehtonen, "Optimization of photovoltaic and wind generation systems for autonomous microgrids with PEV-parking lots," *IEEE Systems Journal*, vol. 16, no. 2, pp. 3260–3271, 2022.

[30] A. Ali, D. Raisz, and K. Mahmoud, "Optimal scheduling of electric vehicles considering uncertain RES generation using interval optimization," *Electrical Engineering*, vol. 100, no. 3, pp. 1675–1687, 2018.

[31] D. Yin, Y. Yang, M. Yang, Z. Yang, C. Li, and L. Li, "A new distributed power system for stability prediction and analysis," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 1–4, Beijing, China, 2019.

[32] S. Theocharides, G. Makrides, G. E. Georghiou, and A. Kyprianou, "Machine learning algorithms for photovoltaic system power output prediction," in *2018 IEEE International Energy Conference (ENERGYCON)*, pp. 1–6, Limassol, Cyprus, 2018.

[33] H. Bano, A. Tahir, I. Ali, A. Haseeb, and N. Javaid, "Electricity load and price forecasting using enhanced machine learning techniques," in *Innovative Mobile and Internet Services in Ubiquitous Computing. IMIS 2019*, L. Barolli, F. Xhafa, and O. Hussain, Eds., vol. 994 of Advances in Intelligent Systems and Computing, pp. 255–267, Springer, Cham, 2020.

[34] D. Moldovan and I. Salomie, "Detection of sources of instability in smart grids using machine learning techniques," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP),*, pp. 175–182, Cluj-Napoca, Romania, 2019.

[35] V. Malbasa, C. Zheng, P.-C. Chen, T. Popovic, and M. Kezunovic, "Voltage stability prediction using active machine learning," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 3117–3124, 2017.

[36] M. Abuella and B. Chowdhury, "Forecasting solar power ramp events using machine learning classification techniques," in *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pp. 1–6, Charlotte, NC, USA, 2018.

[37] G. N. Baltas, C. Perales-González, P. Mazidi, F. Fernandez, and P. Rodríguez, "A novel ensemble approach for solving the transient stability classification problem," in *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 1282–1286, Paris, France, 2018.

[38] D. A. Wood, "Predicting stability of a decentralized power grid linking electricity price formulation to grid frequency applying an optimized data-matching learning network to simulated data," *Technology and Economics of Smart Grids and Sustainable Energy*, vol. 5, no. 1, 2020.

[39] Q. Abu Al-Haija, A. A. Smadi, and M. F. Allehyani, "Meticulously intelligent identification system for smart grid network stability to optimize risk management," *Energies*, vol. 14, no. 21, p. 6935, 2021.

[40] P. Breviglieri, T. Erdem, and S. Eken, "Predicting smart grid stability with optimized deep models," *SN Computer Science*, vol. 2, no. 2, pp. 1–12, 2021.

[41] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi, and F. S. Oueslati, "An effective ensemble learning approach-based grid stability assessment and classification," in *2021 IEEE Kansas Power and Energy Conference (KPEC)*, pp. 1–6, Manhattan, KS, USA, 2021.

[42] S. Shabbir, M. S. Asif, T. M. Alam, and Z. Ramzan, "Early prediction of malignant mesothelioma: an approach towards non-

invasive method," *Current Bioinformatics*, vol. 16, no. 10, pp. 1257–1277, 2021.

[43] T. M. Alam, M. Mushtaq, K. Shaukat, I. A. Hameed, M. Umer Sarwar, and S. Luo, "A novel method for performance measurement of public educational institutions using machine learning models," *Applied Sciences*, vol. 11, no. 19, p. 9296, 2021.

[44] T. M. Alam, K. Shaukat, I. A. Hameed et al., "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020.

[45] T. M. Alam, M. M. A. Khan, M. A. Iqbal, W. Abdul, and M. Mushtaq, "Cervical cancer prediction through different screening methods using data mining," *International Journal of Advanced Computer Science and Applications*, vol. 10, 2019.

[46] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, Washington DC, 2003.

[47] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5718–5727, 2009.

[48] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[49] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018.

[50] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328, Hong Kong, 2008.

[51] H. Wang and B. Raj, "On the origin of deep learning," 2017, https://arxiv.org/abs/1702.07800.

[52] T. M. Alam, K. Shaukat, I. A. Hameed et al., "A novel framework for prognostic factors identification of malignant mesothelioma through association rule mining," *Biomedical Signal Processing and Control*, vol. 68, article 102726, 2021.

[53] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco California, USA, 2016.

[54] T. M. Alam, M. A. Iqbal, Y. Ali et al., "A model for early prediction of diabetes," *Informatics in Medicine Unlocked*, vol. 16, article 100204, 2019.

[55] T. M. Alam, K. Shaukat, M. Mushtaq et al., "Corporate bankruptcy prediction: an approach towards better corporate world," *The Computer Journal*, vol. 64, no. 11, pp. 1731–1746, 2021.