iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72352

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Decentralized Smart Grid Stability Evaluation using ML and DL Approaches

Maloth Anusha[1], Dr. M. Balasubbareddy[2]
Chaitanya Bharathi Institute of Technology, Hyderabad, India

Abstract: The growing integration of decentralized energy technologies such as distributed generation units, microgrids, and systems with bidirectional power flow has significantly complicated the task of maintaining grid stability. Conventional centralized control strategies often fall short when addressing the localized, rapidly changing dynamics of decentralized electrical networks. This paper explores the use of traditional machine learning technics and advanced deep learning models namely Random Forest (RF), Support Vector Machine (SVM), and Convolutional Neural Networks (CNN) to forecast stability conditions in such distributed systems. A dataset of UCIs from kaggle repository is taken into consideration of simulated measurements like tau[x], p[x], g[x], stab, stabf, and node-based interactions, is employed for model development and evaluation. The RF algorithm proves effective in handling diverse input features and offers interpretable results achieving a peak prediction accuracy of 98.95%, while the SVM model excels at classifying distinct stability states with an accuracy of 96.75%, whereas CNNs classifies confusion matrix with 81% accuracy. The findings demonstrate that decentralized deployment of intelligent, data-driven models can enable autonomous decision-making and enhance system robustness. Overall, this study highlights machine learning as a promising tool for ensuring stability in next-generation decentralized power infrastructures.
Keywords: Decentralized power systems, stability forecasting, machine learning, Random Forest, Support Vector Machine, Convolutional Neural Networks, smart grid resilience.

I. INTRODUCTION

The worldwide transition to renewable energy sources has brought tremendous challenges to ensuring the stability of contemporary power systems. In contrast to traditional centralized systems, electrical smart grids in a decentralized configuration employ heterogeneous and dynamic resources such as solar, wind, and battery storage systems that change in production and consumption behavior. Such variability requires strong, real-time stability analysis and control mechanisms [1][3][7]. One of the new paradigms here is Decentralized Smart Grid Control (DSGC), which ventures into grid frequency and other local measurements-based localized decision-making instead of traditional centralized coordination [2][12]. Although DSGC improves privacy, scalability, and robustness, it also adds complexity through the dependency among distributed nodes. Assumptions such as fixed inputs and symmetric conditions used in simplification within the DSGC model diminish its applicability in real-world scenarios [4][16].To overcome these constraints, Machine Learning (ML) and Deep Learning (DL) methods have emerged as strong contenders for grid stability analysis and forecasting. Data-driven models like these are capable of handling non-linear relationships, dealing with high-dimensional input spaces, and learning dynamic changes in grid behavior without the imposition of tight assumptions [5][6][10]. ML algorithms like Random Forest, Support Vector Machines, and ensemble techniques have achieved high predictive accuracy for numerous smart grid applications like load forecasting and stability analysis[9][11]. More complex DL models—such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks—are specifically well-suited to time-series data common in electrical grids. These models are able to learn temporal patterns and dependencies from grid signals and provide accurate classification of stable versus unstable states even with missing or noisy data conditions [2][8][13]. Recent research has also incorporated optimization techniques, federated learning, and edge computing to improve real-time response and security in decentralized grid systems [17][18][19]. As computational infrastructure has made great progress, ML/DL-based solutions are growing more and more viable for implementation in decentralized environments, making the grid operation more resilient and intelligent [14][20]. This work explores and compares the performances of different ML and DL models in predicting and classifying smart grid stability under decentralized setups. Utilizing real-world data sets and simulation platforms, this work seeks to create a scalable, adaptive framework that will further the future of secure, efficient, and smart energy systems.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

II. LITERATURE SURVEY

Traditional methods for grid stability assessment, such as time-domain simulations and small-signal analysis, are accurate but computationally intensive, limiting real-time applications. With the rise of data from PMUs and SCADA systems, machine learning (ML) has emerged as a viable alternative for fast and reliable stability prediction. Supervised learning models like SVMs and Random Forests have been used effectively to classify system states as stable or unstable. Gradient Boosting techniques (e.g., XGBoost) have shown high performance on large, real-time datasets. Deep learning models, especially LSTMs and CNNs, have proven capable of capturing complex temporal patterns in grid dynamics. Unsupervised methods assist in anomaly detection, while reinforcement learning has been explored for adaptive voltage and frequency control strategies. Hybrid models combining simulation and ML predictions, along with ensemble learning approaches, enhance accuracy and reliability. Despite progress, challenges like data scarcity, model interpretability, and generalization persist. Future work aims to integrate explainable AI, transfer learning, and federated modeling to enhance deployment in real-world power systems.

To address the limitations of input rigidity and inequality in decentralized smart grid control (DSGC) architectures, Breviglieri et al. proposed an enhanced deep learning framework. Their implementation of advanced deep neural network models eliminated the dependency on fixed input values, thereby increasing the system's adaptability and improving stability assessment capabilities. Concurrently, Massaoudi et al. introduced a Stratified Ensemble Classification (SEC) approach specifically designed for forecasting stability in decentralized smart grids. This supervised learning-based method demonstrated superior accuracy in detecting grid instabilities, highlighting its potential for practical deployment in modern smart energy systems.

III. METHODOLOGY

The proposed methodology employs supervised machine learning techniques to predict the stability of an electrical power grid based on real-time system parameters. The workflow involves data preprocessing, model training, evaluation, and deployment readiness analysis. The implementation was carried out using Python in a Google Colab environment, leveraging libraries such as pandas, scikit-learn, and matplotlib.

A. *Classification Approach and Architecture*

The research begins with the analysis of the 4-node original datasets. Various ML and DL algorithms were applied to derive results. The selected algorithms include:

1) Dataset Description

4-Node Original Data: Contains 10,000 records

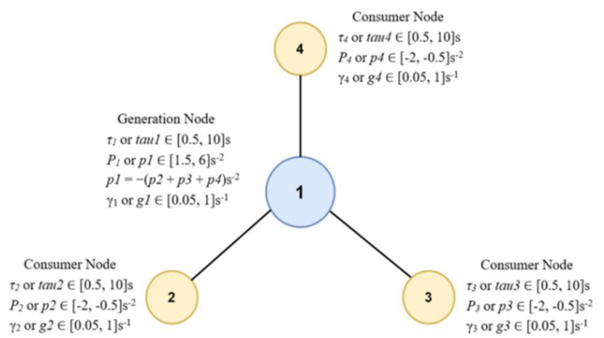


Fig.1. Architecture of the four-node star DSGC system

The dataset used in this work was adopted from UC Irvine Machine Learning Repository. The simulated data consists of 10,000 records of parameters (features) related to grid stability. The number of parameters in each record is 14, with 11 of these parameters are features capturing grid states, 1 being a derived feature from other features, 1 represents the numerical value indicating grid stability, and the last parameter is the record label (i.e., stable/unstable).

Attribute Description: There are 11 predictive variables, 1 auxiliary (non-predictive) variable (`p1`), and 2 target outputs: tau[x]: Denotes participants' response time in seconds between 0.5 and 10. In particular, `tau1` is the response time of the power generator.

p[x]: Refers to nominal power, negative for consumption, positive for production. For consumers, values range between -0.5 and -2 (in s^{-2}).

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

The value of `p1` is found as the absolute sum: `p1 = |p2 + p3 + p4|`. g\[x]: Represents the gamma coefficient, related to price elasticity, whose values range between 0.05 and 1 (in s^{-1}). The producer's coefficient is given as `g1`. stab: This is the target variable which is the greatest real component of the system's characteristic equation roots. A positive value reflects that the system is linearly unstable. Based on the preceding discussion, stabf has been identified as the target variable for the machine learning and deep learning models utilized in this study. The dataset used in this study is a standard open-access dataset for grid stability, which includes features such as voltage, phase angle, and load values. Each instance is labeled as either *stable* or *unstable*. The data consists of numerical attributes, facilitating the use of multiple ML algorithms without the need for encoding categorical variables.

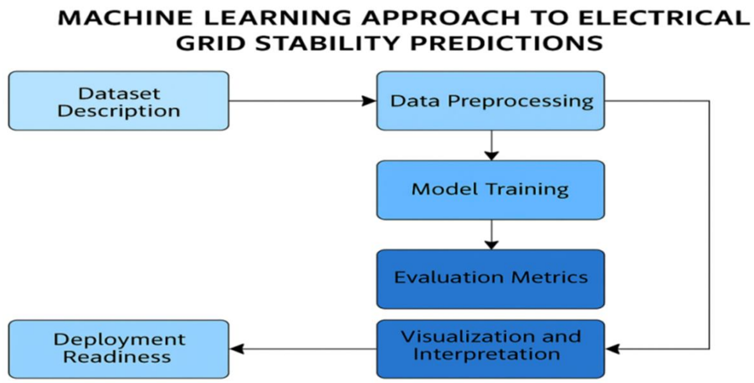


Fig.2. Architecture Framework of Machine Learning

B. Data Preprocessing

Prior to being utilized in the machine learning and deep learning models, the dataset underwent essential pre-processing, which involved three key steps:

1) *Normalization:* All numerical features were scaled to values within the range of 0 to 1 using the min-max normalization technique. This process ensures that no individual feature disproportionately influences the learning algorithm, promoting equal treatment of all numerical inputs.
2) *Encoding the Target Variable:* The categorical target feature, **stabf**, was converted into a numerical format. Specifically, "stable" was mapped to 0, and "unstable" was mapped to 1, enabling the models to interpret the labels effectively.
3) *Splitting the Dataset*: To evaluate model performance on unseen data, the dataset was divided into training and testing sets. The split was chosen to be 80%-20%, meaning that 80% of the available records (that is, 8,000 records) are used for training the models and 20% (that is, 2,000 records) are used for testing. The split was stratified to ensure the original proportion of stable and unstable instances was preserved in both subsets.

C. Model Training

Three classification models were trained and evaluated:

1) *Random Forest Classifier (RFC)* – An ensemble learning method based on decision trees, suitable for handling non-linear relationships.
2) *Support Vector Machine (SVM)* – Applied with an RBF kernel for maximum margin classification in high-dimensional feature space. Or A robust classification model that identifies the optimal hyperplane to separate classes in the feature space, particularly effective in high dimensional spaces.
3) *Convolutional neural network (CNN)* – are deep learning algorithms which learn automatically from structured data by extracting features. In smart grids, they are employed for fault detection and load forecasting by examining time-series patterns.

D. Evaluation Metrics

The models were evaluated using the following metrics:

1) *Accuracy:* This is one of the most commonly used ways to measure how well a model performs. It tells us the percentage of predictions the model got right both when it correctly identified stable and unstable conditions out of all the predictions it made.
2) *Precision:* Precision helps us understand how many of the cases the model predicted as “unstable” were actually correct. It focuses only on the positive predictions and checks how many of those were true, which is especially useful when false alarms are a concern.

3) *Recall:* Recall looks at how good the model is at catching all the actual unstable cases. It measures how many of the real unstable conditions were correctly identified, making it important when missing a true unstable case is risky.
4) *F1-Score:* The F1-score brings precision and recall together into a single number. Since there's often a trade-off between catching all true cases (recall) and avoiding false alarms (precision), the F1-score balances the two, giving a better overall picture of a model's performance.

E. Visualization and Interpretation

Matplotlib and seaborn were used to generate:

- Correlation matrices.
- Confusion matrices.
- Distribution plots of feature importance (for the Random Forest model).
- Histograms.

These visualizations provided insights into model behavior and feature relevance.

F. Deployment Readiness

The trained model exhibiting the best performance (Random Forest) was saved using joblib, making it suitable for deployment in a real-time power grid monitoring system. The model can be integrated with PMU or SCADA data streams for real-time prediction of grid stability status.

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 11, 32)	128
max_pooling1d (MaxPooling1D)	(None, 5, 32)	0
conv1d_1 (Conv1D)	(None, 3, 64)	6,208
max_pooling1d_1 (MaxPooling1D)	(None, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 128)	8,320
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Total params: 14,914 (58.26 KB)
Trainable params: 14,914 (58.26 KB)
Non-trainable params: 0 (0.00 B)

A 1D CNN was constructed using the Sequential API of Keras to classify power grid stability into two classes. It contains two convolutional layers with 32 and 64 filters, respectively, and max-pooling layers to decrease temporal resolution. The feature maps are flattened and fed into a dense layer of 128 units and a dropout layer to increase generalization and avoid overfitting. The output layer contains 2 softmax activated neurons to make predictions for stable versus unstable grid states. The model transforms from sizes like (None, 11, 32) to (None, 2) and contains 14,914 trainable parameters, which makes it appropriate for application in real-time or computationally restricted environments.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

IV. MODEL PERFORMANCE OVER 20 EPOCHS

(Accuracy and Loss on Training and Validation Sets)

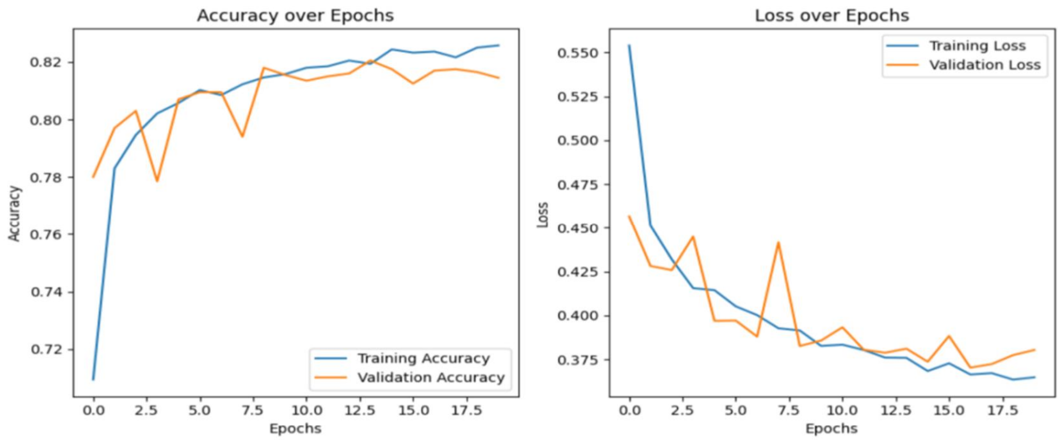


Fig.3. Training and validation accuracy and loss plotted over 20 epochs

The training and validation graphs give us a good sense of how well the model is learning over time. On the left, we can see the accuracy steadily improving for both the training and validation sets across the 20 epochs. The training accuracy increases quickly and eventually levels off around 82%, while the validation accuracy follows a similar trend with some small fluctuations. This suggests the model is learning meaningful patterns without overfitting too much.

On the right, the loss curves show how the model's prediction error decreases over time. The training loss drops consistently, and the validation loss generally decreases as well, although it fluctuates slightly. These ups and downs in validation loss are common and can be due to randomness in the data batches or smaller validation size.

Overall, the curves show that the model is training well; it improves performance without diverging or getting stuck, and the gap between training and validation metrics is small indicating good generalization to unseen data.

V. RESULTS AND DISCUSSION

A. Data Visualization (KDE plot)

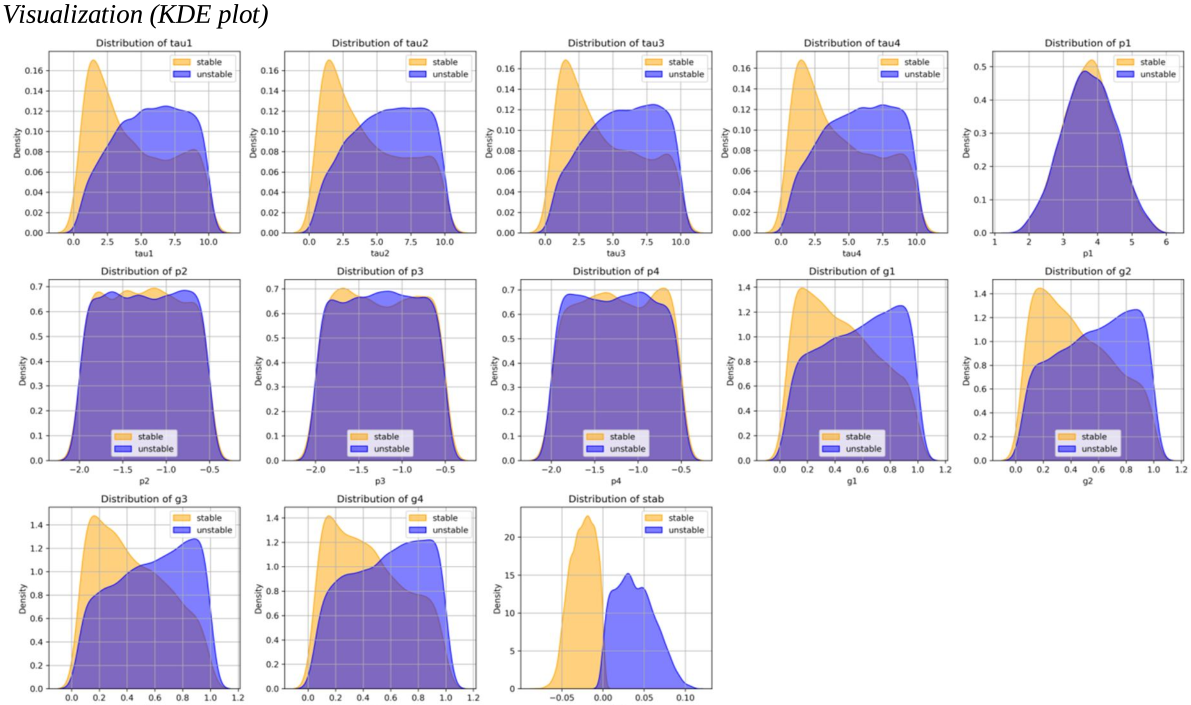


Fig.4. Statistical Distribution of Features across Stability State

Fig.4. shows the distributions of important system features under stable and unstable operating conditions in a decentralized electrical smart grid. Each histogram compares feature values for the two classes, where stable instances are represented in orange and unstable instances in blue. Time constant parameters (tau1 to tau4) show a clear tendency for increased values to be most common among the unstable cases, with a strong relationship to system dynamics during these events. p1 to p4 parameters demonstrate modest differentiation, with stable cases tending to group within well-defined ranges of values and limited but clear discriminative ability. Conversely, gain features (g1 to g4), specifically g3 and g4, show better separation with unstable cases lumped in the higher values. The stab metric shows the most prominent separation between the two states with stable conditions bunched around lower values and unstable ones biased towards higher values, further substantiating its status as a principal indicator of system stability. In general, these distributions of features are helpful in choosing informative variables and in training models for predicting grid stability.

B. Analysis of Heatmap and Correlation

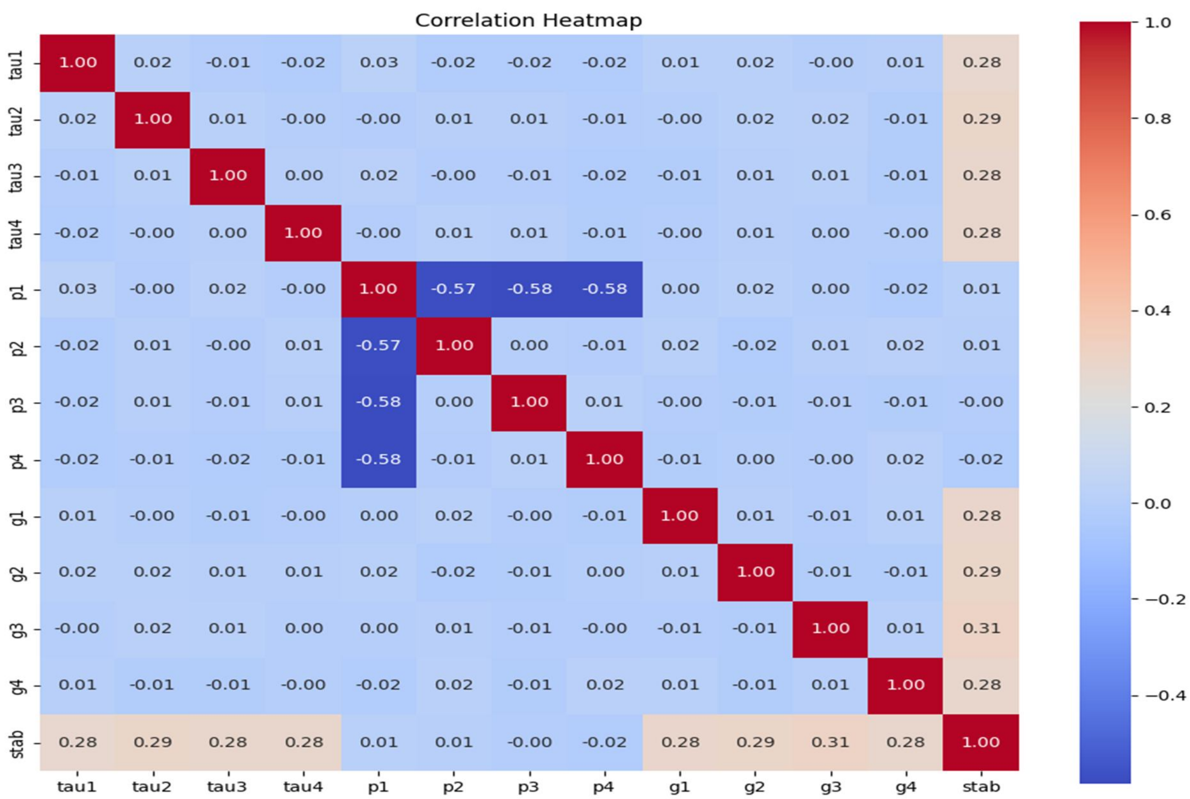


Fig.4. Heat map and Correlation Matrix output

The heatmap shows that the time constants (tau1 to tau4) and generation capacities (g1 to g4) have a moderate positive relationship with the system's stability index (stab), with correlation values around 0.28 to 0.31. Notably, g3 stands out with the strongest link at 0.31. On the other hand, the power measurements (p1 to p4) are highly negatively correlated with one another (roughly -0.57 to -0.58), yet they exhibit little to no direct connection with the stability index. Overall, the results suggest that the time constants and generation capacities play a more significant role in influencing system stability than the power variables.

C. Metrics of Methods Used

STABLE-0
UNSTABLE-1
Table I shows that Random Forest exhibits exceptional performance in predicting smart grid stability, achieving an accuracy of 98.95% along with high precision of 0.98, recall of 0.96, and F1-score of 0.98. Support Vector Machine(SVM) follows with an accuracy of 96.75%. While remaining are comparatively weaker.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Table I. STABLE STATE CONDITION METRICS

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9895	0.98	0.96	0.98
SVM	0.9675	0.95	0.93	0.94
Confusion matrix	0.81	0.73	0.72	0.73
LSTM	0.615	0.41	0.81	0.69

Under unstable grid conditions as shown in Table II, all models showed improved performance. Random Forest achieved the highest accuracy of 98.95% with a precision of 0.99, recall of 0.97 and an F1-score of 0.99. Support Vector Machine (SVM) followed, excelling with precision of 0.96, recall and F1-score of 0.98 respectively.

Table II. UNSTABLE STATE CONDITION METRICS

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9895	0.99	0.97	0.99
SVM	0.9675	0.96	0.98	0.98
Confusion matrix	0.81	0.85	0.86	0.86
LSTM	0.615	0.81	0.54	0.66

D. Model Evaluation and Error Analysis

To compare the accuracy of various machine learning models, three main evaluation metrics were utilized: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These evaluations were applied to gauge the predictive quality of Random Forest, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM) models on a shared test dataset. To facilitate an equitable and consistent comparison, a special Python function was created to decode one-hot encoded outputs into equivalent class labels.

Table III. ERROR METRICS OF MODEL EVALUATION

Model	**MSE**	**RMSE**	**MAE**
Random Forest	0.0005	0.0224	0.0005
SVM	0.0462	0.2150	0.1605
LSTM	0.2501	0.5001	0.5001

Among the three models tested, Random Forest model achieved the most accurate results, showing extremely low MSE, RMSE, and MAE values. The SVM model showed moderately well performance but with higher error rates compared to Random Forest, while the LSTM model had the least accurate predictions, reflected in its high error values. Based on these outcomes, Random Forest appears to be the most appropriate model for this particular dataset and task.

VI. CONCLUSION

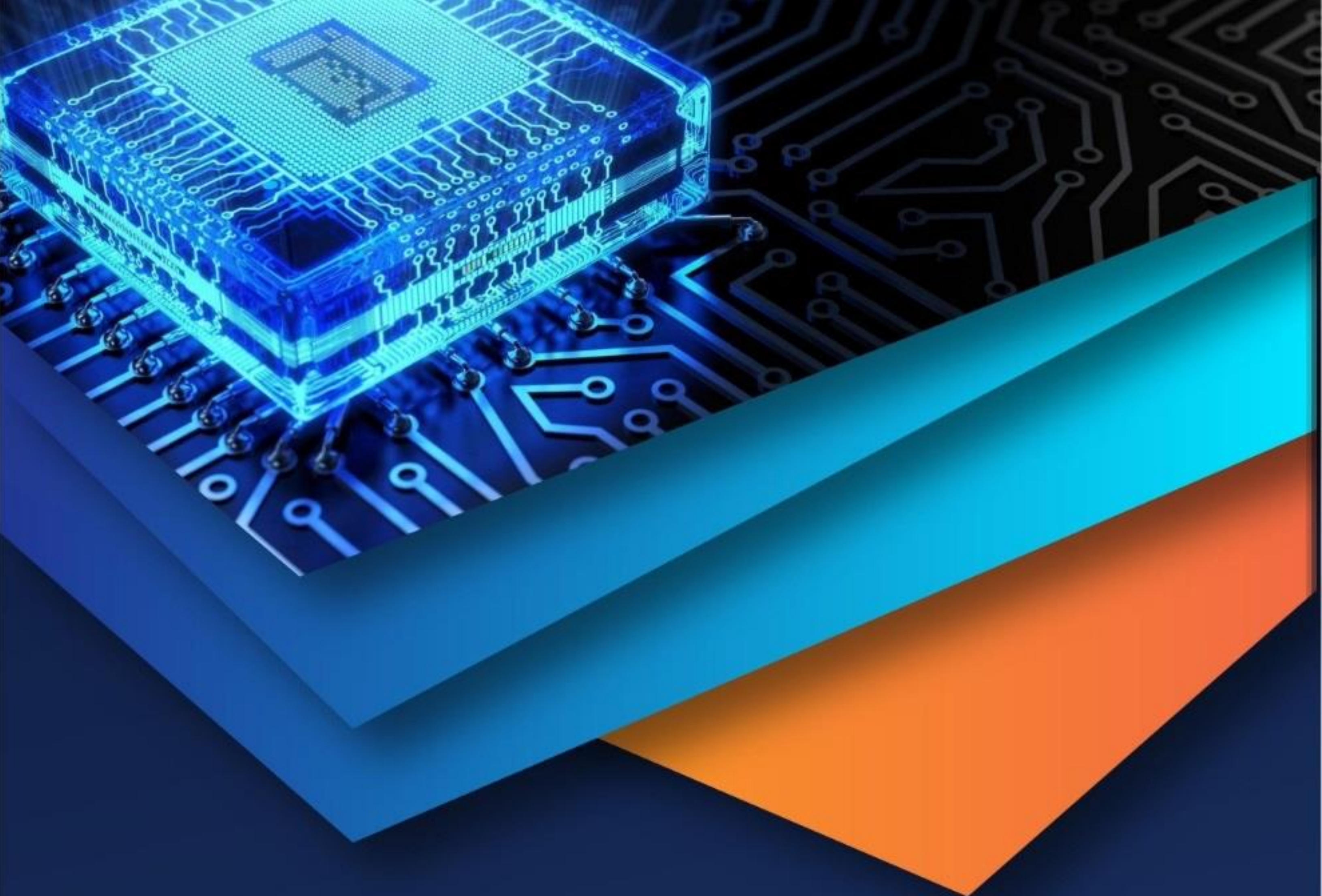
This paper presents a data-driven methodology for predicting electrical grid stability, utilizing both traditional machine learning techniques and more sophisticated deep learning models. A performance comparison based on accuracy, precision, recall, and F1-score—was conducted to evaluate how these models perform in the context of assessing and forecasting grid stability in complex power systems. The goal was to highlight the effectiveness of AI in enabling real-time grid condition prediction and guiding necessary actions during disturbances to maintain stability. The analysis identified the Random Forest model as the most reliable, achieving a high accuracy of 98.95%, followed by Support Vector Machines (SVM) with an average accuracy of 96.75%. While the Convolutional Neural Network (CNN) model reached an accuracy of 81%, the Long Short-Term Memory (LSTM) model underperformed, suggesting that it may require additional training data or further tuning of hyperparameters. Based on the evaluation metrics, Random Forest is recommended for immediate deployment or further optimization, while SVM could serve as a viable alternative or a backup. In summary, traditional machine learning models demonstrate strong performance, particularly when there is a need to strike a balance between model complexity and predictive accuracy.

VII. FUTURE WORK

Implementing these models in actual grid control systems using real-time data pipelines will be a major step toward operational integration including more diverse or larger datasets, especially with real-world events or anomalies would strengthen the model's generalization capability. Incorporating interpretability frameworks will help stakeholders understand model decisions, which is critical for deployment in critical infrastructure.

REFERENCES

[1] M. Hinz and A. Drossel, “Decentralized Smart Grid Stability Modeling with Machine Learning,” Energies, vol. 16, no. 22, p. 7562, Nov. 2023.
[2] J. Liu et al., “Deep Neural Network-Based Smart Grid Stability Analysis,” Energies, vol. 17, no. 11, p. 2642, 2024.
[3] A. Kumar et al., “Renewable Energy Management in Smart Grids Using Big Data Analytics,” International Journal of Energy Research, vol. 47, no. 2, pp. 1498–1515, 2023.
[4] R. Sharma et al., “A Machine Learning-Based Model for Stability Prediction of Decentralized Smart Grid Control Systems,” Journal of Electrical and Computer Engineering, vol. 2022, Article ID 2697303, 2022.
[5] M. A. Rana et al., “Predicting Smart Grid Stability with Optimized Deep Models,” IEEE Access, vol. 11, pp. 54322–54332, 2023.
[6] S. Roy et al., “Metaheuristic Optimization with Deep Learning Enabled Smart Grid Stability Prediction,” Journal of Information Security and Applications, vol. 77, p. 103743, 2023.
[7] H. Zhang and Y. Wang, “A Novel Approach to Predicting the Stability of the Smart Grid,” Applied Energy, vol. 314, p. 118973, 2022.
[8] P. Singh et al., “Review on Smart Grid Load Forecasting for Smart Energy Management Using Machine Learning,” IEEE Systems Journal, vol. 17, no. 3, pp. 3567–3578, 2023.
[9] A. Khan et al., “Optimizing Smart Power Grid Stability Based on the Prediction of a Deep Learning Model,” JOIV: International Journal on Informatics Visualization, vol. 7, no. 1, pp. 1–9, 2023.
[10] L. Zhang et al., “Leveraging the Power of Machine Learning and Data Balancing Techniques for Smart Grid Stability Prediction,” Electric Power Systems Research, vol. 214, p. 108998, 2023.
[11] K. Patel et al., “Classification of Smart Grid Stability Prediction Using Cascade Machine Learning Methods,” Neural Computing and Applications, vol. 35, pp. 14567–14584, 2023.
[12] T. Preusse et al., “Predicting Stability of a Decentralized Power Grid Linking Electricity Price to Frequency,” Journal of Reliable Intelligent Environments, vol. 5, pp. 167–180, 2019.
[13] R. Joshi and M. S. Islam, “A Short Report on Deep Learning Synergy for Decentralized Smart Grid Cybersecurity,” Frontiers in Artificial Intelligence, vol. 6, 2025.
[14] A. S. Rajput, “Smart Grid Stability Prediction Using Machine Learning,” GitHub Repository, 2024.
[15] P. Breviglieri, “Data Science for Smart Grid Stability,” GitHub Repository, 2024.
[16] T. Preusse, C. Möhrke, and A. Drossel, “Predicting Stability of a Decentralized Power Grid Linking Electricity Price to Frequency,” Journal of Reliable Intelligent Environments, 2019.
[17] Y. Li et al., “FedDiSC: A Computation-Efficient Federated Learning Framework for Power Systems Disturbance and Cyber Attack Discrimination,” IEEE Transactions on Smart Grid, vol. 14, no. 1, pp. 515–526, Jan. 2023.
[18] M. Chen et al., “Learning Regionally Decentralized AC Optimal Power Flows with ADMM,” IEEE Transactions on Power Systems, vol. 39, no. 2, pp. 1728–1741, 2024.
[19] R. Rahman and D. Zhao, “Electrical Load Forecasting Using Edge Computing and Federated Learning,” Applied Energy, vol. 330, p. 120317, 2023.
[20] X. Li and J. Wu, “A Review of Federated Learning in Energy Systems,” Energy Reports, vol. 9, pp. 2789–2805, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY