International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Forecasting Stability of Smart Grids using Highway Deep Pyramid Convolutional Neural Network (HPDCNN) Approach

Mr. Sankarapandian Sivarajan[1*], Dr. S. D. Sundarsingh Jebaseelan[2], Dr. Alagappan Pandian[3], Dr. Easwaramoorthy Nandakumar[4]

Submitted: 29/08/2023 **Revised**: 24/10/2023 **Accepted**: 02/11/2023

Abstract: This paper proposes the Highway Deep Pyramid Convolutional Neural Network (HPDCNN) technique for smart grid stability forecasting. The objective is to enhance the accuracy and reliability of stability predictions in smart grids. The technique aims to reduce errors and uncertainties associated with stability predictions, thereby improving the overall performance of smart grids. The HPDCNN technique controls the strengths of highway networks and pyramid networks to reduce the impact of noise and irrelevant features in the data. This reduction in noise improves the robustness and accuracy of stability predictions, enhancing the reliability of the technique. The HPDCNN approach effectively captures both temporal and spatial dependencies in smart grid data by combining the strengths of highway networks and pyramid networks. The highway network enables the model to learn long-term dependencies, while the pyramid network facilitates multi-scale feature extraction. Using the HDCNN algorithm, the paper presents a methodology for predicting the stability of smart grids based on various input parameters such as power generation, consumption patterns, weather conditions, and grid infrastructure. The algorithm is trained using labeled data, where each data point is classified as either stable or unstable based on the actual stability status of the corresponding smart grid. Once trained, the HDCNN algorithm can classify new, unseen data points as stable or unstable, providing insights into the current and future stability of the smart grid. By identifying unstable grids, operators and energy management systems can take appropriate actions to prevent potential disruptions or outages, ensuring the reliable and efficient operation of the smart grid system. Experimental results demonstrate that the HPDCNN technique achieves an accuracy rate of over 95%. The proposed schema is evaluated using testing and training values, along with a confusion matrix, to validate its performance. Overall, the proposed HPDCNN technique has the potential to improve the accuracy and reliability of stability predictions, leading to more efficient and sustainable smart grid systems. The Proposed FSSG-HPDCNN approach accuracy value is 99.88% which is higher than other existing methods like FSSG-WHO, FSSG-PSO, and FSSG-HBO methods

Keywords: *Smart grid, Electricity, Forecasting Stability, Stable, MinMaxScaler, Highway Deep Pyramid Convolutional Neural Network (HPDCNN), Voltage.*

1. Introduction

Networks transport electricity from locations where they are created and consumed by households and business establishments [1]. They are sophisticated networks of substations, wiring and transmission/distribution lines, transformers, and other components. To allow for the consumption of power by these various customers, the voltages in electrical networks are gradually lowered distribution voltage level from transmission voltage levels for servicing customers [2]. Transmissions and distributions where various wire and cabling systems are involved are often distinguished from one another [3]. Electricity must always be available whenever and wherever it is required, without interruption [4]. This raises a number of concerns. Grid management is extremely complicated; necessitating specialist sectors for experts to consider choices for energy laws and government sustainability efforts [5].These obstacles can include the effects of extreme weather conditions, wildlife damages, human sabotages, and other internal/external problems with equipment failure and crucial assets [6]. SGs are paradigm changes from conventional electric grids and have several functions [7].

Self-healing capabilities in SGs make it possible to quickly recover from disruptions and detect and address grid issues automatically [8]. Despite the fact that many nations began implementing smart metering programs more than a decade ago, most of them are still in early stages of developments [9]. Since the electrical markets, power consumption rates, rules, stakeholder expectations, and even the process of producing electricity are all changing, SGs are being introduced all over the world, but

[1*]*Research Scholar, Department of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India.*
[2]*Professor, Department of Electrical and Electronics Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India.*
[3]*Professor, Department of Electrical and Electronics Engineering Koneru Lakshmaiah Education Foundation, Guntur District, Andhra Pradesh, India*
[4]*Professor, Department of Electrical and Electronics Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India*
[*]*Corresponding Author Email: ssivarajan78@gmail.com*

occasionally with varying methodologies and objectives [10]. Electricity supply and demand must be balanced for electrical networks to remain stable. To preserve this equilibrium, traditional technologies produce power based on demand response, on the other hand, is a realistic alternative for Advanced Grids with a high source of Eco-Friendly Energy [11]. This suggests that changes in electricity use occur as a result of price fluctuations. Local power auctions and establishing realistic demand and supply estimates are two approaches for calculating and conveying the price to customers [12].SGs are more than advanced metering infrastructure and smart meters. Smart meters, however, are only one component of SGs; others include distribution lines and substations, as well as ways to minimize power outages and ensure power quality [13]. Energy integrations from diverse sources have a higher emphasis on Eco-Friendly Energy, power production, sensors along transmission lines, and power system automations. Energy integration with greener energies has been made possible by extensive use of smart sensors, intelligent devices, and automated controls that monitor and analyses network health and conditions in order to detect irregularities and problems [14].

Recently suggested systems of CSGs (Control for SGs) have received much attention. This is due to the fact that it doesn't necessitate gathering and processing a lot of data [15].It binds the cost of power to the grids frequency so that all participants, or all energy producers and consumers, may access it. Assumptions are used to build current CSG models. Simulations of its stability are made easier by certain hypotheses, i.e., to determine if participant response to pricing adjustments leads to grid instability [16]. To achieve this, differential equations are used to describe the system. Decentralized energy generations imply increasing energy productions in a variety of methods. If users of energy in general produce their own energy, fewer costs are involved at various levels of electrical networks. The integration of RESs (renewable energy resources) in SGs requires real-time monitoring of added energy sources. Computing stability of SGs are crucial challenges as the processes are time-dependent. In the present method of studying stability, certain inputs are given fixed values throughout all equations and simulation runs, whereas other inputs are given values taken from fixed distributions in experiments [17]. The outcome is a collection of one-dimensional intervals that illustrate how stability depends on the tabulated input values. The conclusions were drawn from analysis where certain inputs were not considered. It is possible to see where present approaches fall short. First of all, when only one input value is changed at a time, as in the example, only a small number of additional input values are examined. This makes it impossible to estimate how different inputs interact [18]. The term "fixed inputs issue might be used to describe this situation. Second, it is unrealistic to assume that all input values are equal, especially when some of them are intrinsic to system participants, such as private families, and cannot be controlled from outside the system. These inputs include things like energy users' price elasticity their proclivity to shift usage in in reaction to pricing fluctuations or the passage of time it takes for them to respond to a price change. Renewable energy is gaining appeal as a potential alternative to conventional, limited, and climate-damaging fossil fuels. Its implementation, on the other hand, introduces a cascade of new paradigms, two of which demand special consideration. Prior to the arrival of renewable energy sources, the traditional operational ecosystem was made up of a small number of industrial firms (sources) that transferred energy to consumers through unidirectional flows. [19]. With the arrival of renewable energy, end customers can not only consume but also generate energy, and are referred to be "presumes," and energy flow across distribution networks is bidirectional in SGs. Despite increased flexibility brought by introduction of renewable sources and emergence of presumed, managing supplies and demands have become complex. SGs have broader definitions and applications because of the numerous opportunities made possible by this significant paradigm shifts and increasing use of technologies in the context of deployments of SGs. To put it in another way, the goal is to develop Meta models. The use of AIs (Artificial Intelligences) on input parameters of SGs can bring new insights like stability of forecasting systems in the presence of new energy consuming devices. This study proposes HPDCNN to assess the stability of SGs [20]. Following this introductory section, subsequent section details on studies related to stability of grids followed by the proposed methodology in section three. Section four details on results with discussions while this paper is concluded in section five.

The key contributions of the proposed technique are abridged below:

- In the paper "Forecasting Stability of Smart Grids using Highway Deep Pyramid Convolutional Neural Network (HPDCNN) Algorithm," the proposed technique aims to classify the stability of smart grids into two categories: stable and unstable.
- Initially, Min-Max Normalizations are used to transform data values to values between 0 and 100, where new values are added.
- VGG-16 model is used to feature-extract more advanced image analysis and recognition by extracting meaningful information from raw visual data.
- Using the HDCNN algorithm, the paper presents a approach for estimating smart grid stability based on various input parameters such as power generation,

consumption patterns, weather conditions, and grid infrastructure.

- The algorithm is trained using labeled data, where each data point is classified as either stable or unstable based on the actual stability status of the corresponding smart grid.
- Overall, the HDCNN technique contributes to the advancement of smart grid stability forecasting by introducing a novel approach that leverages deep learning techniques and considers the complex dynamics of smart grid systems. Its potential benefits include improved prediction accuracy, proactive decision-making, and enhanced grid reliability.

2. Recent Research Work

Li, [21] have developed the rapid shift from fossil fuels to renewable energy sources in the energy system has brought cleaner energy to urban smart grid systems. However, the volatile and erratic nature of renewable energy (RE) has made accurate short-term load forecasting a challenging task. Recent attempts have focused on machine learning (ML) for short-term load projections, however many of these models overlook crucial features such as data characterization, parameter fine-tuning, and predictive stability. To address these concerns, a unique short-term load forecasting model was created. To reduce the unpredictability in load data, it uses deep data mining and multi-step rolling forecasting, as well as a de-noising technique. The model dynamically calculates artificial neural network (ANN) parameters using the phase space reconstruction (PSR) technique and then optimizes them using a multi-objective grasshopper optimization algorithm (MOGOA). This technique was proven through case studies in metropolitan smart grid networks in Victoria and New South Wales, Australia., offering a promising solution for more reliable and accurate short-term load forecasts in the era of renewable energy integration.

Alquthami et al., [22] have developed a variety of cutting-edge ML approaches, including logistic regression (LR), support vector machines (SVM), naive Bayes (NB), decision tree classifier (DTC), K-nearest neighbor (KNN), and neural networks (NNs), has created to assess their performance. This study's main purpose was to assess the prediction error and accuracy of machine learning (ML) algorithms for short-term load forecasting (STLF).

Hafeez et al., [23] have developed A forecasting framework known as "FE-SVR-mFFO" combines the Feature Engineering (FE) and Modified Fire-Fly Optimization (mFFO) methodologies with Support Vector Regression (SVR).FE reduces superfluous and unnecessary functionality to achieve maximum computing efficiency. The relevant parameters of the SVR model are collected and fine-tuned using the mFFO approach to successfully prevent trapping into local optimums and offer reliable prediction results. Furthermore, the majority of literature research focuses on improving forecast accuracy. However, the forecasting model's stability and convergence rate are equally important in determining its usefulness and productivity.

Ali and Ramaswamy [24] have developed a control mechanism based on complicated fuzzy logic. Advanced fuzzy control combines the finest renewable energy resource (RER) power distribution to limit interruptions caused by overloading and variations in the demand profile. The model's usefulness and validity were demonstrated using the transmission network's 9 Bus test system. Power network overloading is caused by nonlinearity in the load profile and variations in demand induced by forecasting mistakes for short-term loads. The electricity grid's condition deteriorated after a network breakdown caused overload. Identifying how these disturbances affected the power system network was a critical component of this project.

Liu et al., [25] have suggested A smart grid edge sensing data generates a multitude of important information that encourages the development of new smart power applications in IoT-focused smart cities and society. The accuracy of load prediction in the smart grid is limited since the potential links between time series of power demand data and characteristics of temperature, weather, and date have not been well investigated. A power load prediction technique based on edge sensing data-imaging conversion (DIC) was developed to increase forecasting accuracy in smart cities and society by finding generalized patterns and hidden correlations in data connected to power demand.

Ali et al., [26] have suggested a control method that emphasizes a sophisticated fuzzy logic approach. Overloading of the power network was caused by nonlinearity in the load profile and changes in demand caused by forecasting errors for short-term loads. When a breakdown in the power system network causes overloading, the power system's condition deteriorates. A key aspect of this research was analyzing how these disruptions affected the power system network. Advanced fuzzy control incorporates the best power distribution from renewable energy resources (RERs) to minimize disruptions caused by overloading and changes in the demand profile.

Zulfiqar et al. [27] have developed a hybrid load forecasting technique that is both quick and exact. The proposed model includes two modules and a forecaster based on locally weighted support vector regression (LWSVR). These modules are optimizers based on feature engineering (FE) and adaptive grasshopper optimization (AGO). The hybrid feature selector (HFS) was created in the FE module by combining wrapper and filter approaches to choose an appropriate subset of features. Furthermore, the curse of feature dimensionality was reduced by detecting and removing erroneous features using the instance-based Relief- F (REF) and information theoretic-based Mutual Information (MI) filters. To address the over fitting issue, the HFS module was additionally optimized utilizing the recursive feature elimination (RFE) wrapper feature selection technique.

2.1. Background of the Research work

Recent research indicates that the increasing integration of renewable energy sources, distributed energy resources, and advanced control systems in modern power grids has created a critical need for predicting the stability of smart grids. These grids, designed to enhance energy distribution efficiency and reliability, encounter new challenges due to the complexities introduced by these technologies. The variability in renewable energy generation, along with dynamic demand response and the intricate interplay of distributed resources, can lead to unforeseen stability issues. Additionally, smart grids rely on communication networks and digital control systems, making them vulnerable to cyber security threats. Therefore, it is essential to proactively identify potential stability risks. Accurate stability forecasts enable grid operators to anticipate voltage and frequency imbalances, optimize resource allocation, and implement preventive measures against cascading failures. This proactive approach is vital for ensuring the resilience and effectiveness of smart grids in the face of evolving energy landscapes and potential disruptions. These above mentioned drawbacks are inspired to do this research work.

3. Proposed Methodology

In this section, Forecasting Stability of Smart Grids is discussed. The proposed method is called as FSSG-HPDCNN to help us predict the stability of SGs. The selections of the input values for the CSG simulations are justified. The dataset was generated through the simulation of a four-node star electrical grid featuring centralized power production. The model incorporates input features such as total power balance, nominal power generated or consumed at each grid node, the time it takes for participants to alter their consumption or production in response to price changes (referred to as reaction time), and energy price elasticity. This dataset is primarily centered around star topologies comprising four nodes, featuring a sole centralized power source that supplies energy to three consumption nodes. It has been designed with a particular emphasis on the analysis of reliable Smart Grids (SGs).Block diagram of proposed HDPCNN method is shown in Figure 1.

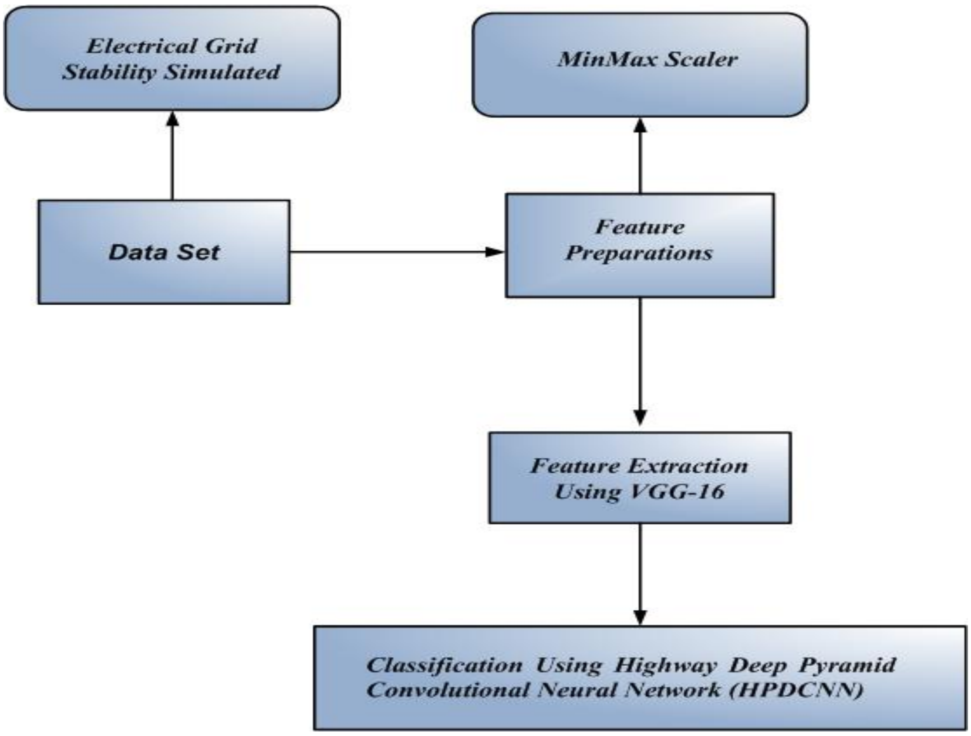


Fig 1: Block diagram of proposed HDPCNN method

The suggested HPDCNN was evaluated using the Smart Grid Dataset and have a five-stage evaluation process. Functions were created to help with mapping correlations and Image analysis of particular characteristics or observations from the dataset. The first step is to use correlations between variables to identify data relationships. The second stage is featuring preparation, which involves standardizing dataset recognizing values between 0 and 1relevant features for classifications. Division of Features, the following stage, involves dividing samples into Training and Testing sets. The classifications of stable and unstable grid sources are made using NNs, which are subsequently assessed using a Confusion matrix, metrics for training and testing accuracy.

3.1. Dataset

Electrical Grid Stability Simulated Dataset was produced at Karlsruher Institute für Technologies, Karlsruhe, Germany and donated to the University of California (UCI) Machine Learning Repository. This study makes use of an improved version of the original dataset. The analysis is carried out using a process similar to that described in [25] for various sets of input data. The original dataset has 10,000 observations. Because the reference grid is symmetric, a permutation of three customers occupying three consumer nodes can be represented by extending the dataset by a factor of 3!(3 factorial) which equals 6 times. In the upgraded edition, the dataset now contains a total of 60,000 observations. Additionally, it includes two dependent variables and 12 essential predictive characteristics. There were no missing values because the content of the dataset is drawn from simulated exercises. Additionally, there is no need to code any features because they are all initially numerical. Such dataset characteristics enable a seamless transition to machine modeling without the requirement for feature engineering or data preparation.

3.1.1. Assessments of Relationships between Data Using HPDCNN

Two or more variables are said to be statistically connected if their values fluctuate with correlations. For Example, there is a relationship between the two variables "hours ran" and outputs created, if an increase in hours run, outputs automatically increase. The statistical measure of correlations which are numbers, describe sizes and directions of relationships between variables. HPDCNN uses assessments of correlations between variables.

3.2. Feature preparations using MinMaxScaler

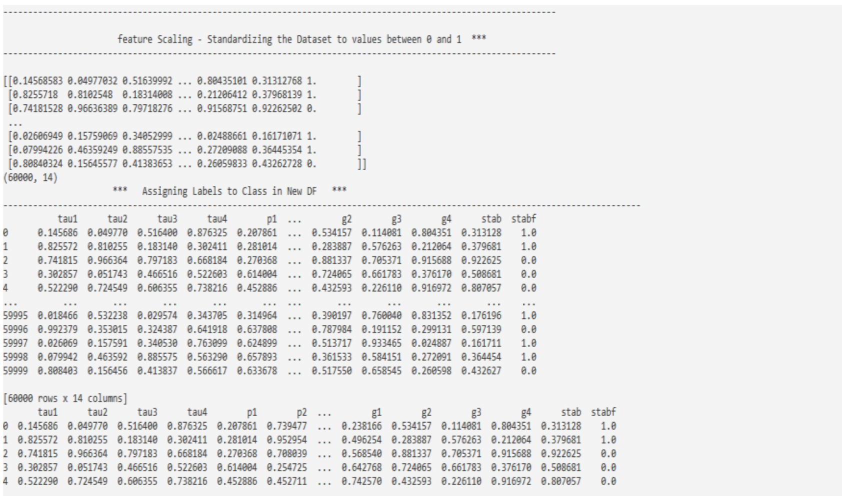
HPDCNN the MinMaxScaler is used to standardize dataset values between 0 and 1, and the normalizations are performed using Equation (1).

$$Z_i = (x_i - \min(x) - \min(x)) \quad (1)$$

where Z_i stands for ith normalized dataset values, x_i represents ith values of datasets, $\min(x)$ implies minimum values in datasets and $\min(x)$ represents maximum values in datasets. The goal of Min-Max Normalizations is to transform data values to numbers between 0 and 100, with new values derived using Equation (2).

$$New\ value = (value - \min)/(\max - \min) * 100 \quad (2)$$

The dataset comprises 12 attributes, and their distribution patterns in relation to the dependent variable "stab" have been examined. Most of these attributes exhibit distributions that closely resemble a normal distribution, as anticipated. However, it's worth noting that the combined absolute sum of "p2," "p3," and "p4" demonstrates a distribution that is nearly normal, with only a minor skew factor of -0.013.distributions, are typically consistent across the board because this data is derived from simulations with defined ranges for all features. Figure 2 shows Feature preparations of NNSPT.



```
--------------------------------------------------------------------------------------------
                  feature Scaling - Standardizing the Dataset to values between 0 and 1  ***
--------------------------------------------------------------------------------------------
[[0.14568583 0.04977032 0.51639992 ... 0.80435101 0.31312768 1.        ]
 [0.8255718  0.8102548  0.18314008 ... 0.21206412 0.37968139 1.        ]
 [0.74181528 0.96636389 0.79718276 ... 0.91568751 0.92262502 0.        ]
 ...
 [0.02606949 0.15759069 0.34052999 ... 0.02488661 0.16171071 1.        ]
 [0.07994226 0.46359249 0.88557535 ... 0.27209088 0.36445354 1.        ]
 [0.80840324 0.15645577 0.41383653 ... 0.26059833 0.43262728 0.        ]]
(60000, 14)
                   ***   Assigning Labels to Class in New DF   ***
--------------------------------------------------------------------------------------------------------
           tau1      tau2      tau3      tau4        p1  ...        g2        g3        g4      stab  stabf
0      0.145686  0.049770  0.516400  0.876325  0.207861  ...  0.534157  0.114081  0.804351  0.313128    1.0
1      0.825572  0.810255  0.183140  0.302411  0.281014  ...  0.283887  0.576263  0.212064  0.379681    1.0
2      0.741815  0.966364  0.797183  0.668184  0.270368  ...  0.881337  0.705371  0.915688  0.922625    0.0
3      0.302857  0.051743  0.466516  0.522603  0.614004  ...  0.724065  0.661783  0.376170  0.508681    0.0
4      0.522290  0.724549  0.606355  0.738216  0.452886  ...  0.432593  0.226110  0.916972  0.807057    0.0
...         ...       ...       ...       ...       ...  ...       ...       ...       ...       ...    ...
59995  0.018466  0.532238  0.029574  0.343705  0.314964  ...  0.390197  0.760040  0.831352  0.176196    1.0
59996  0.992379  0.353015  0.324387  0.641918  0.637808  ...  0.787984  0.191152  0.299131  0.597139    0.0
59997  0.026069  0.157591  0.340530  0.763099  0.624899  ...  0.513717  0.933465  0.024887  0.161711    1.0
59998  0.079942  0.463592  0.885575  0.563290  0.657893  ...  0.361533  0.584151  0.272091  0.364454    1.0
59999  0.808403  0.156456  0.413837  0.566617  0.633678  ...  0.517550  0.658545  0.260598  0.432627    0.0

[60000 rows x 14 columns]
       tau1      tau2      tau3      tau4        p1        p2  ...        g1        g2        g3        g4      stab  stabf
0  0.145686  0.049770  0.516400  0.876325  0.207861  0.739477  ...  0.238166  0.534157  0.114081  0.804351  0.313128    1.0
1  0.825572  0.810255  0.183140  0.302411  0.281014  0.952954  ...  0.496254  0.283887  0.576263  0.212064  0.379681    1.0
2  0.741815  0.966364  0.797183  0.668184  0.270368  0.708039  ...  0.568540  0.881337  0.705371  0.915688  0.922625    0.0
3  0.302857  0.051743  0.466516  0.522603  0.614004  0.254725  ...  0.642768  0.724065  0.661783  0.376170  0.508681    0.0
4  0.522290  0.724549  0.606355  0.738216  0.452886  0.452711  ...  0.742570  0.432593  0.226110  0.916972  0.807057    0.0
```

Fig 2: Feature preparations of NNSPT

Segregating train and test sets in HPDCNN: The label dataset only includes "stabf," however The features dataset included all 12 original predictive characteristics. Since, samples were shuffled; training sets included first 54,000 observations, while testing sets included 6,000 observations. Despite the fact that the dataset was large, To confirm that the original dataset distributions were retained following the split, the proportion of "stable" and "unstable" observations were estimated for both training and testing sets. In HPDCNN, sample splits between training and testing are shown in Figure 3.

```
(60000,)
                 ***  spliting the dataset into a training set (80%) and a test set (20%)  ***
------------------------------------------------------------------------------------------------------


       -----------------Training Set (X)------------------------
           tau2      tau3      tau4        p1        p2  ...        g2        g3        g4      stab  stabf
48572  0.114570  0.923984  0.389082  0.764838  0.447130  ...  0.656625  0.535970  0.956476  0.294565    1.0
38696  0.425058  0.903545  0.445517  0.685579  0.668732  ...  0.231953  0.407246  0.402395  0.394137    1.0
13611  0.446244  0.692132  0.534992  0.596379  0.846416  ...  0.946238  0.883800  0.550332  0.883276    0.0
35213  0.799964  0.816891  0.999183  0.433731  0.799129  ...  0.324475  0.380013  0.354765  0.534562    0.0
31766  0.361012  0.467041  0.692469  0.724725  0.256305  ...  0.921944  0.656107  0.485551  0.710936    0.0
...         ...       ...       ...       ...       ...  ...       ...       ...       ...       ...    ...
54343  0.002516  0.335316  0.010269  0.529081  0.222823  ...  0.483449  0.749452  0.172539  0.199357    1.0
38158  0.468321  0.440863  0.560319  0.266789  0.997731  ...  0.516116  0.822157  0.991631  0.863324    0.0
860    0.210435  0.640401  0.200481  0.666490  0.855771  ...  0.108930  0.780259  0.844330  0.589036    0.0
15795  0.921902  0.614180  0.148009  0.541279  0.399543  ...  0.882795  0.279434  0.620977  0.690967    0.0
56422  0.332598  0.286933  0.519761  0.707058  0.108240  ...  0.026524  0.678247  0.612471  0.551016    0.0

[48000 rows x 13 columns]


       ----------------- Training Set (y)------------------------
48572    1.0
38696    1.0
13611    0.0
35213    0.0
31766    0.0
        ...
54343    1.0
38158    0.0
860      0.0
15795    0.0
56422    0.0
Name: stabf, Length: 48000, dtype: float64
```

Fig 3: Train/test Splits in NNSPT

3.3. Feature Bifurcations using VGG-16

VGG-16 is a deep CNN composed of multiple convolutional and pooling layers. These layers perform hierarchical feature extraction, starting from low-level features like edges and textures to higher-level features like object parts and shapes. In the context of malicious sample classification, the network would learn to extract relevant patterns or features from the data, which could include binary code, assembly instructions, or other representations of the malicious code. Transfer Learning: VGG-16 is often used as a pre-trained model. This means it's trained on a large dataset like Image Net, which contains images of everyday objects to learn general image feature representations. For your task, you might fine-tune or transfer the pre-trained VGG-16 model to your specific problem domain (malicious code classification. This is called transfer learning, where you retrain the last few layers of the network with your specific dataset. Once the model has been fine-tuned on your dataset of malicious and non-malicious samples, it learns to associate certain features with the concept of being malicious. During the training process, the model learns to adjust its internal parameters to minimize the classification error. This results in a model that can accurately assign labels to new, unseen samples as either malicious or non-malicious based on the features it has learned. When you have a new, previously unseen sample that you want to classify, you pass it through the pre-processed VGG-16 model. The model will extract relevant features from the sample and use its learned knowledge to predict whether the sample is malicious or not Equation (3).

$$C_i = \sigma\left(_{j=1}^{F} W_{ij} * C_{i-1j} + b_i\right) \tag{3}$$

where C_i is the output feature map, W_{ij} is the convolutional kernel for the i-th layer and j-th filter, C_{i-1j} is the previous layer's *j-th* feature map, b_i is the bias term and σ is the activation function. The hierarchical feature extraction process, starting from low-level features to higher-level features due to crucial enabling computer vision systems to understand and interpret visual information in a manner similar to how humans perceive the world. This hierarchical approach allows for the extraction of increasingly abstract and meaningful information from raw visual data, ultimately contributing to more advanced and accurate image analysis, recognition, and understanding.

3.4. Classification Using Highway Deep Pyramid Convolutional Neural Network (HPDCNN)

HPDCNN uses neural network for classifications. HPDCNN has a significant benefit in establishing early ending callbacks (iterations), since it ends NN learning after 5 consecutive epochs of no progress in validation losses, where epochs mean training with all of the training data for one cycle.

The objective is to develop a robust classification model which distinguishes between stable and unstable grid sources using a Highway Deep Pyramid Convolutional Neural Network (HDP-CNN). This advanced architecture combines elements of highway networks and pyramidal structures to effectively capture complex patterns in time-series data related to power grid parameters. The process begins with the collection and preprocessing of a diverse dataset encompassing both stable and unstable instances from various grid sources. This dataset is labeled to indicate the stability status of each instance. The data is then prepared for input into the HDP-CNN by transforming the time-series data into a 2D image-like format, where the temporal dimension corresponds to the x-axis and various parameters are represented along the y-axis

HPDCNN transformation Equation are given as Equation (4),

$$H_{transformed} = \mathrm{ReLU}(W_H * X + b_H) \tag{4}$$

where W_H is learnable convolutional kernels for the transformation and gate respectively, * represents the convolution operation, b_H is corresponding bias terms.

The unique characteristics of grid source data. It consists of multiple convolutional layers with varying filter sizes to capture different scales of patterns, As such; the model may learn both short and long-term dependencies. Additionally, highway connections are integrated into the network to facilitate the flow of information across layers, thereby addressing challenges associated with vanishing gradients and promoting deeper network exploration. Training the HDP-CNN involves optimizing its parameters using the labeled dataset. This includes selecting an appropriate loss function (such as binary cross-entropy) and optimization algorithm (like Adam) to iteratively improve the model's performance. Hyper parameter tuning is crucial for achieving optimal results and might involve adjusting learning rates, batch sizes, and the number of layers or filters in the architecture. To evaluate the model's performance, it's tested on a separate dataset that it hasn't encountered during training. Metrics such as accuracy, precision, recall, F1-score, and stability curves could provide insights into its ability to generalize to new, unseen data. Once the model demonstrates satisfactory performance, it can be deployed to classify the stability of grid sources in real-time scenarios. Continual monitoring and periodic retraining of the model with updated data are crucial to account for potential changes in grid behavior over time. This iterative process ensures the model remains accurate and adaptable to evolving grid conditions. In essence, the HDP-CNN approach offers a sophisticated solution for classifying stable and unstable grid sources, contributing to the overall reliability and efficiency of power systems.

HPDCNN output formation is given as Equation (5),

$$H_{out} = T.H_{transformed} + C.X \tag{5}$$

where $H_{transformed}$ is the transformed information, H_{out} is the final output of the highway layer, In conclusion, developing a robust classification model to distinguish between stable and unstable grid sources using a Highway Deep Pyramid Convolutional Neural Network (HDP-CNN) is a promising approach for addressing grid stability and reliability issues. By leveraging the capabilities of deep learning and the hierarchical features extracted by the HDP-CNN architecture, this model has the potential to provide accurate and timely predictions, helping to prevent grid failures and improve overall energy infrastructure management. However, it is essential to emphasize that the success of this model depends on high-quality data, proper preprocessing, hyper parameter tuning, and rigorous validation to ensure its reliability and generalizability in real-world grid monitoring and control scenarios.

3.4.1. Evaluations of Classifications Using HPDCNN

The proposed method is assessed using two key concepts there are training and validation losses. These metrics serve as indicators of the model's performance. They quantify the errors generated by the models when processing inputs and producing outputs. High loss values suggest that the models are making significant errors, while low loss values indicate fewer errors. Losses are determined through cost functions that are tailored to the specific problems being solved and the data provided to the neural networks. Errors can be quantified in various ways, but commonly, cross-entropy is used for binary classification tasks. Training losses are used to evaluate how well the neural network models fit the training data. They measure the computational errors of the models on the training dataset. Training losses are computed by summing the errors for each example in the training set. These losses are typically calculated for each batch and can be visualized by plotting curves of training losses. Validation losses, on the other hand, assess the performance of the neural network models on validation datasets. Validation datasets are portions of the data that are set aside for the purpose of evaluating model performance. These losses are similar in nature to training losses and are calculated by summing the errors of samples in the validation set. Validation losses are typically computed after each epoch, helping to determine whether the models require further fine-tuning or adjustments. These losses can be visualized as learning curves. In practice, training and validation losses are often displayed together on graphs to diagnose the models' performance and identify areas that require adjustment. When both training and validation losses are high, and validation losses surpass training losses, it suggests under fitting. This means that the models are accurately capturing the training data but struggle with large errors. Conversely, when validation losses exceed training losses, it indicates over fitting. In such cases, the models perform well on the training data but struggle to generalize to new data. The proposed HPDCNN method is designed to mitigate both under fitting and over fitting issues through its optimizations of neural networks, ensuring more robust and reliable model performance.

3.5. Confusion Matrix:

To evaluate the performance of the classifier, the confusion matric is best. The basic idea is that instances of class a counted as instances of class B. Look in the fifth row and third column of the confusion matrix to discover the number of times the classifier confused pictures of 5s with 3s. A confusion matrix's rows indicate real classes, while the columns represent predicted classes. The confusion matrix provides information about preferring a simpler statistic at times. Table 1 lists the confusion matrix of classifications using HPDCNN

Table 1: Confusion matrix of HPD-CNN Total data set =10000 Train: 6000(60%) Test: 4000(40%)

	Predicted Unstable	**Predicted Stable**
Actual Unstable	1998	3
Actual Stable	2	1997

4. Results and Discussion

The proposed HPDCNN was developed in Python 3 on Windows 10 with an AMD processor. This section contains stage-by-stage outcomes of the proposed system in the form of figures or tables, as appropriate. The augmented smart grid dataset was used in the study.

To scale the proposed Performance metrics like accuracy, F-Score, precision, computation time, specificity, stability and error rate are investigated in order to assess performance. It is decided to use confusion matrix to scale the performance parameters.

4.1. Accuracy

The ratio of accurate prediction to total count of proceedings is the accuracy Equation (6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

here TP refers True Positive, TN refers True Negative, FP refers False Positive, FN and refers False Negative.

4.2 F1-score

The weighted average of precision and accuracy is the F1-Score.. This is determined by Equation (7),

$$F1 - Scorevalue = 2 \times \frac{recall \times precision}{recall + precision} \quad (7)$$

The performance Equation is provided in and the evaluation parameter of F-score for detecting emotion from input signals, are analyzed

4.3. Specificity

This is called as True Negative rate. This is scaled by Equation (8)

$$specificity = \frac{(TN)}{(TN + FP)} \quad (8)$$

4.4. Assessments of relationships between data using HPDCNN:

Analyzing correlations among numerical features and their relationships with the dependent variable is essential as it helps identify potential issues such as collinearity. The provided heat map illustrates the association between the dependent variable 'stabf' and the 12 numerical attributes. It's important to note that we've also included the alternative dependent variable 'stab' to provide additional context regarding its relationship with 'stabfIt is reasonable to expect such a link to be significant (-0.83), which supports the Section 3 prediction that it should be discarded. Furthermore, while the connection between "p1" and its subcomponents "p2," "p3," and "p4" is above average, it is insufficient to support any removal Figure 4 shows Correlations in HPD-CNN.

Fig 4: Correlations in HPD-CNN

Training and Validation Accuracy of HPDCNN on SDN traffic dataset is shown in Figure 5. The validation accuracy began at 0.65% during the first epoch, experienced a gradual increase, and reached 1% around 1.8 epochs. Subsequently, it plateaued, maintaining a constant accuracy level between 1.8 to 8 epochs. Training and Validation Loss of HPDCNN on SDN traffic dataset is shown in Figure 6. The validation loss value begins at 0.35% during the first epoch, subsequently decreasing to 0.00% between the first and third epochs. Following this, the loss value stabilizes and remains constant from the third to the eighth epoch. Comparison of accuracy value with Proposed FSSG-HPDCNN approach and existing method is shown in Figure 7. The Proposed FSSG-HPDCNN approach accuracy value is 99.88%, FSSG-WHO method accuracy value is 84%, FSSG-PSO method accuracy value is 62% and the HBO method accuracy value is 72%. The Proposed FSSG-HPDCNN approach is higher accuracy value compare than other existing methods for stable. The Proposed FSSG-HPDCNN approach accuracy value is 99.88%, FSSG-WHO method accuracy value is 72%, FSSG-PSO method accuracy value is 76% and the HBO method accuracy value is 62%. The Proposed FSSG-HPDCNN approach is higher accuracy value compare than other existing methods for unstable. Comparison of computation time with proposed and existing methods is shown in Figure 8. The Proposed FSSG-HPDCNN approach computation time is 100 sec, FSSG-WHO method computation time is 200sec, FSSG-PSO method computation time is 250 sec and the FSSG-HBO method computation time is 280 sec. The Proposed FSSG-HPDCNN approach is lower time value compare than other existing methods. Comparison of error value with proposed and existing method is shown in Figure 9.The Proposed FSSG-HPDCNN approach error value is 2%, FSSG-WHO method error value is 18%, FSSG-PSO method error value is 35% and the FSSG-HBO method error value is 25%. The Proposed FSSG-HPDCNN approach error value is lower than other existing methods for stable. The Proposed FSSG-HPDCNN approach error value is 2%, FSSG-WHO method error value is 32%, FSSG-PSO method error value is 25% and the FSSG-HBO method error value is 35%. The Proposed FSSG-HPDCNN approach error value is lower than other existing methods for unstable.

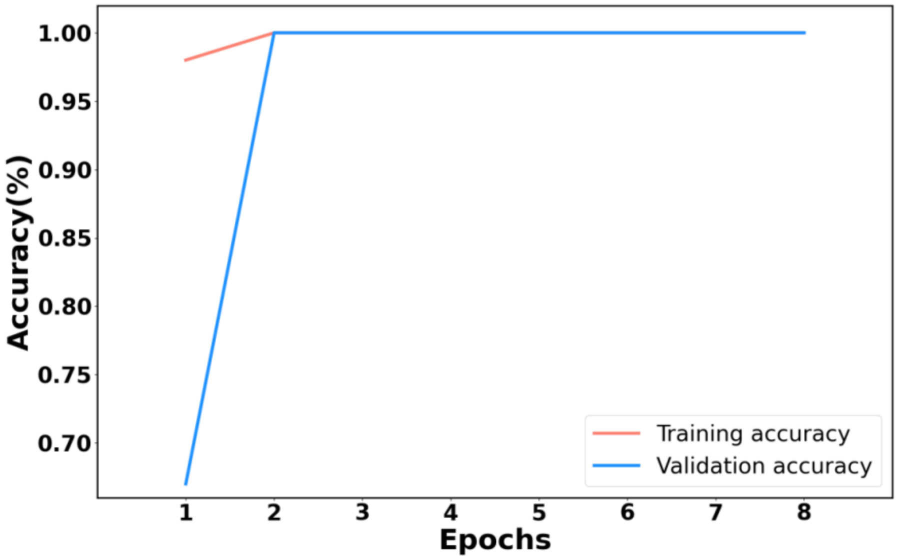


Fig 5: Training and Validation Accuracy of HPDCNN on SDN traffic dataset

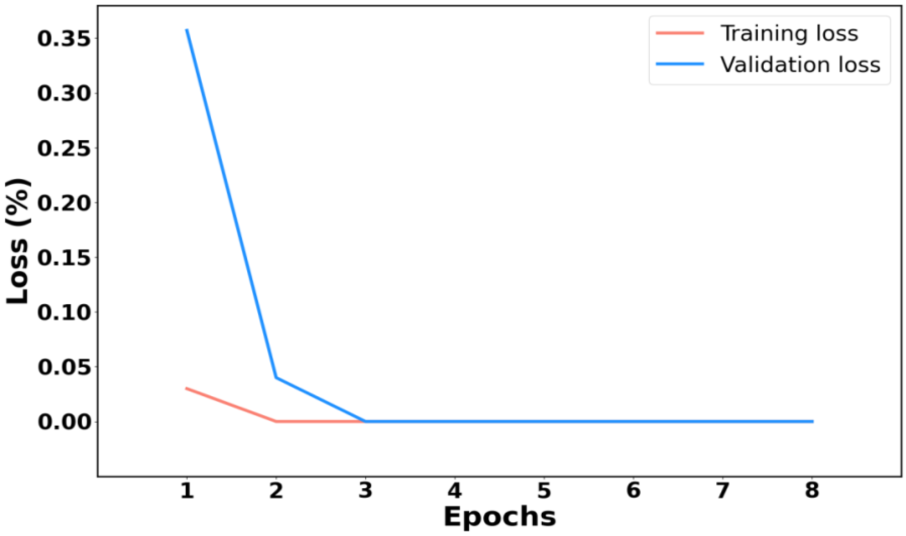


Fig 6: Training and Validation Loss of HPDCNN on SDN traffic dataset

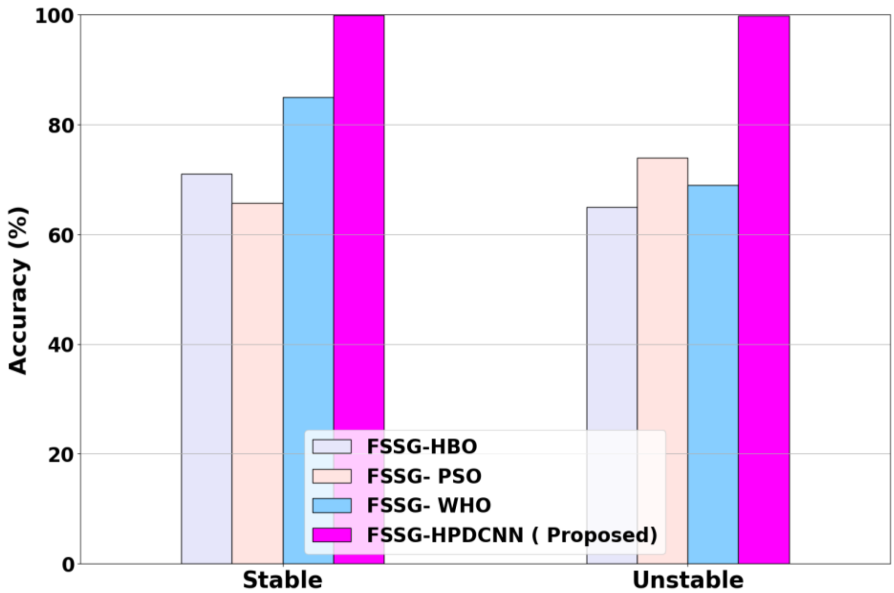


Fig 7: Comparison of accuracy value with proposed and existing method

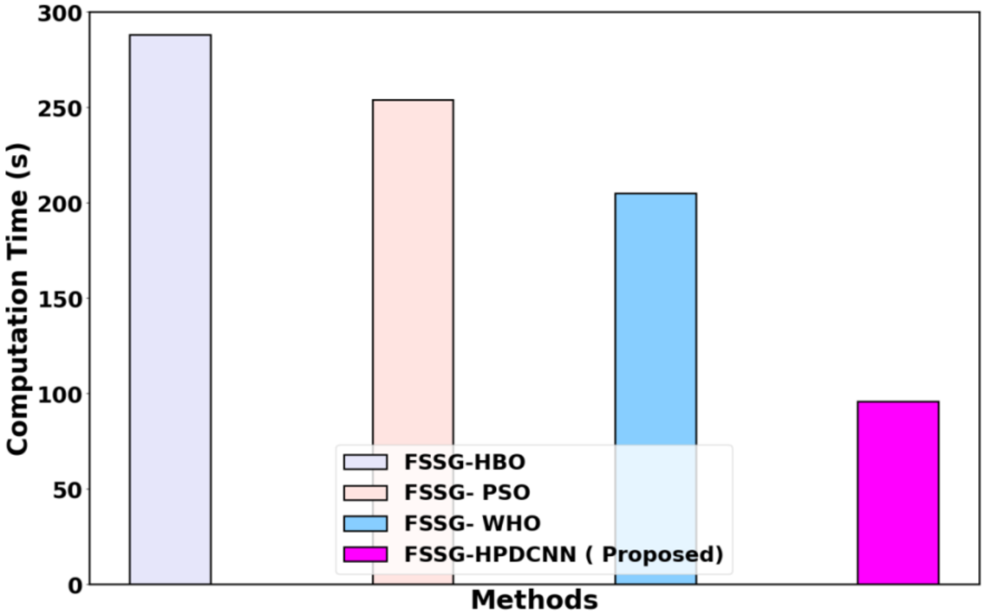


Fig 8: Comparison of computation time with proposed and existing methods

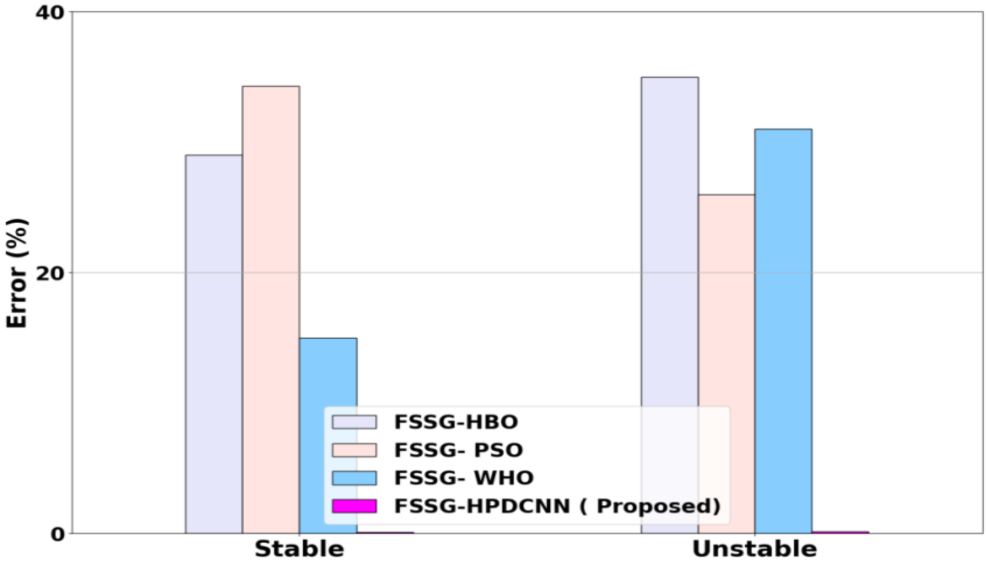


Fig 9: Comparison of error value with proposed and existing method

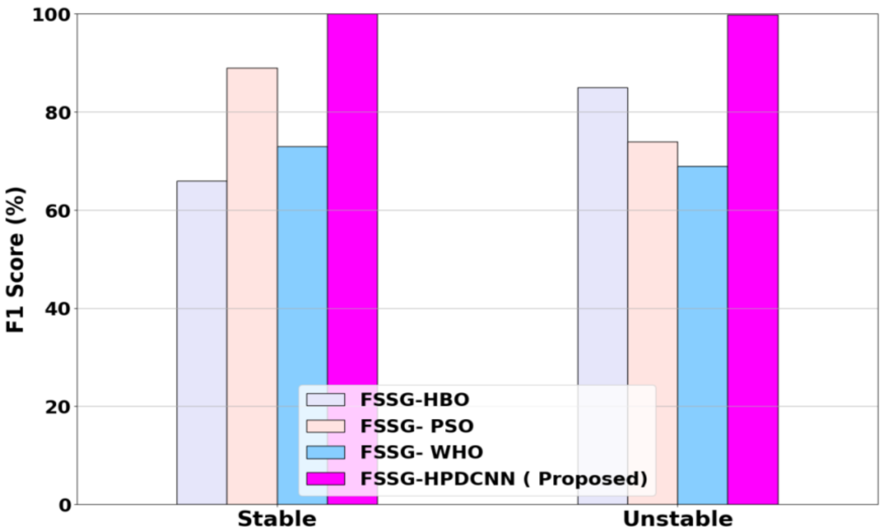


Fig 10: Comparison of F1 Score value with proposed and existing method

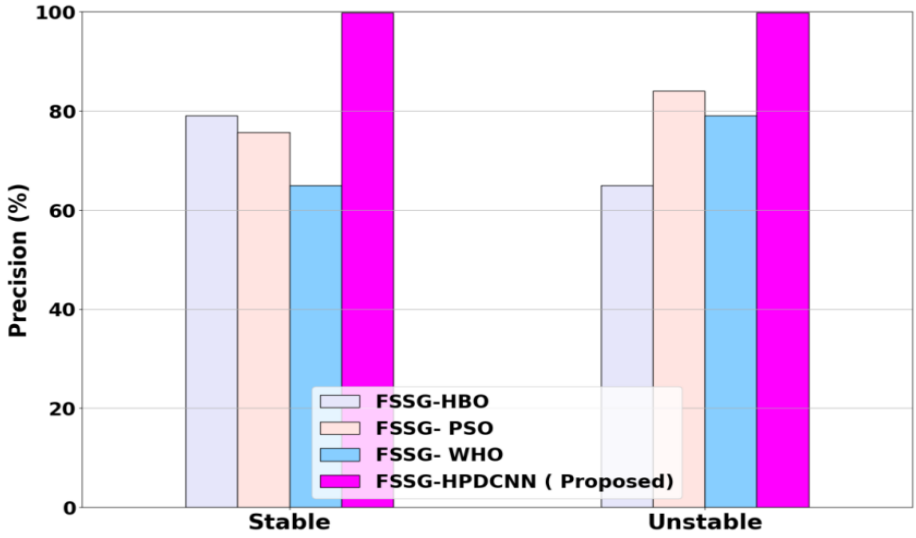


Fig 11: Comparison of precision value with proposed and existing methods

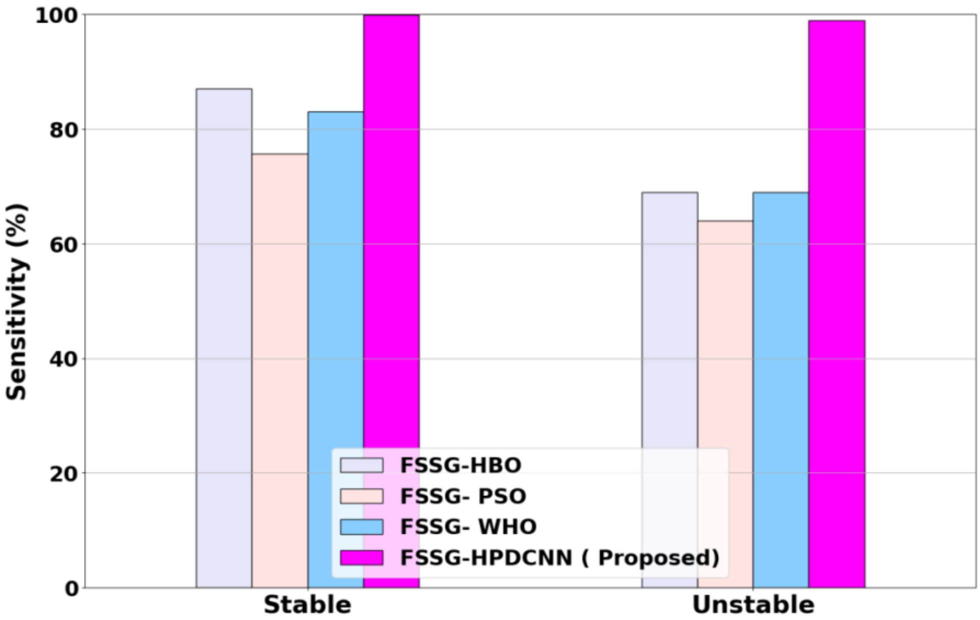


Fig 12: Comparison of sensitivity with proposed and existing methods

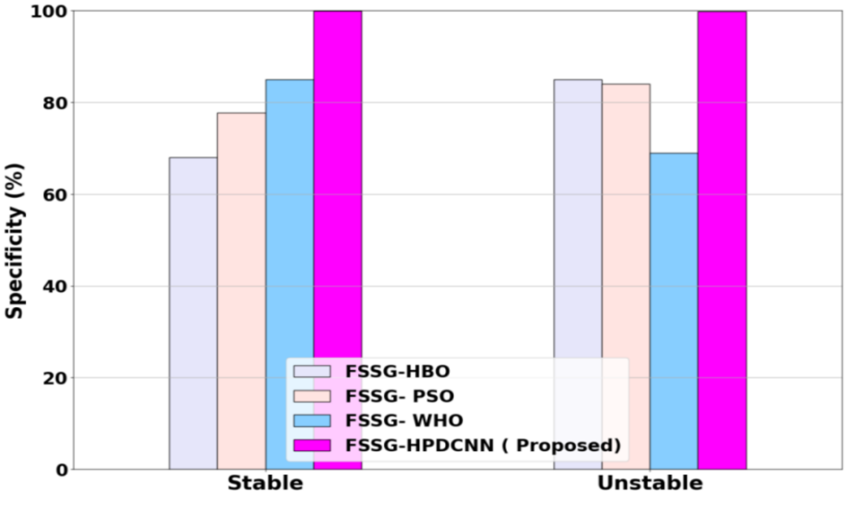


Fig 13: Comparison of specificity with proposed and existing methods

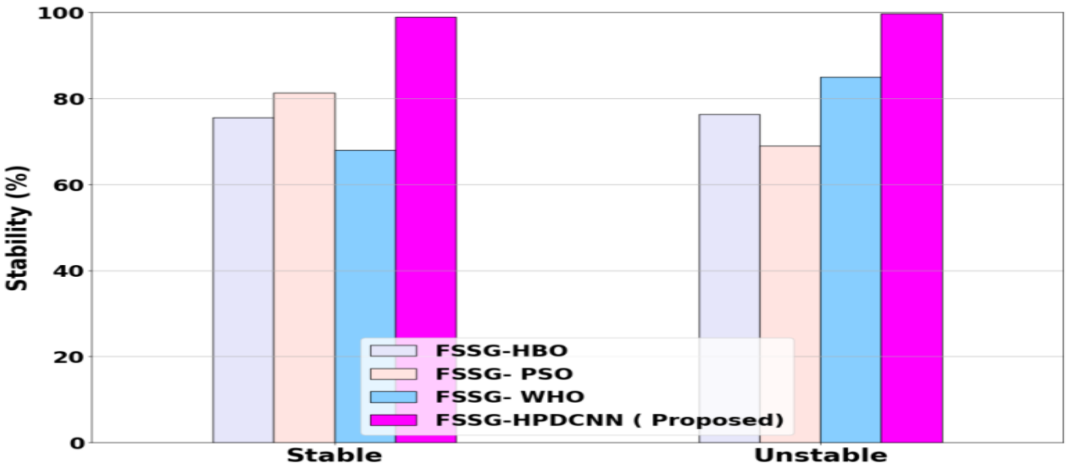


Fig 14: Comparison of stability with proposed and existing methods

Comparison of F1 Score value with proposed and existing method is shown in Figure 10. Comparison of F1 score value with proposed and existing method is shown in Figure 10.The Proposed FSSG-HPDCNN approach F1 score value is 99.2%, FSSG-WHO method F1 score value is 76%, FSSG-PSO method F1 score value is 85% and the FSSG-HBO method F1 score value is 70%. The Proposed FSSG-HPDCNN approach F1 score value is higher than other existing methods for stable. The Proposed FSSG-HPDCNN approach F1 score value is 99.7%, FSSG-WHO method F1 score value is 72%, FSSG-PSO method error value is 78% and the FSSG-HBO method error value is 85%. The Proposed FSSG-HPDCNN approach error value is higher than other existing methods for unstable. Comparison of precision value with proposed and existing methods is shown in Figure 11. The Proposed FSSG-HPDCNN approach precision value is 99.4%, FSSG-WHO method precision value is 68%, FSSG-PSO method precision value is 74% and the FSSG-HBO method precision value is 78%. The Proposed FSSG-HPDCNN approach precision value is higher than other existing methods for stable. The Proposed FSSG-HPDCNN approach precision value is 99.4%, FSSG-WHO method precision value is 78%, FSSG-PSO method precision value is 84% and the FSSG-HBO method precision value is 65%. The Proposed FSSG-HPDCNN approach precision value is higher than other existing methods for unstable. Comparison of sensitivity with proposed and existing methods is shown in Figure 12. The Proposed FSSG-HPDCNN approach sensitivity value is 99.4%, FSSG-WHO method sensitivity value is 68%, FSSG-PSO method sensitivity value is 84% and the FSSG-HBO method sensitivity value is 78%. The Proposed FSSG-HPDCNN approach sensitivity value is higher than other existing methods for stable. The Proposed FSSG-HPDCNN approach sensitivity value is 99.4%, FSSG-WHO method sensitivity value is 64% FSSG-PSO method sensitivity value is 62% and the FSSG-HBO method sensitivity value is 70%. The Proposed FSSG-HPDCNN approach sensitivity value is higher than other existing methods for unstable. Comparison of specificity with proposed and existing methods is shown in Figure 13. The Proposed FSSG-HPDCNN approach specificity value is 99.4%, FSSG-WHO method specificity value is 84%, FSSG-PSO method specificity value is 78% and the FSSG-HBO method specificity value is 70%. The Proposed FSSG-HPDCNN approach specificity value is higher than other existing methods for stable. The Proposed FSSG-HPDCNN approach specificity value is 99.4%, FSSG-WHO method specificity value is 70% FSSG-PSO method specificity value is 82% and the FSSG-HBO method specificity value is 84%. The Proposed FSSG-HPDCNN approach specificity value is higher than other existing methods for unstable. Comparison of stability with proposed and existing methods is shown in Figure 14. The Proposed FSSG-HPDCNN approach stability value is 98.4%, FSSG-WHO method stability value is 70%, FSSG-PSO method stability value is 78% and the FSSG-HBO method stability value is 74%. The Proposed FSSG-HPDCNN approach stability value is higher than other existing methods for stable. The Proposed FSSG-HPDCNN approach stability value is 98.4%, FSSG-WHO method stability value is 84% FSSG-PSO method stability value is 72% and the FSSG-HBO method stability value is 78%. The Proposed FSSG-HPDCNN approach stability value is higher than other existing methods for unstable.

Discussions

Decision trees are one of the machine learning models that can provide outputs in the desired format, as discussed in Section I. It's worth noting that some argue for the joint selection of experimental design and statistical models [26]. Additionally, there are various types of system stability that we can briefly touch upon, along with their respective advantages and disadvantages. Stability against Single Perturbations refers to a system's ability to return to an equilibrium state after experiencing a single disturbance, often related to power requirements during short periods. Analyzing this type of stability introduces several simulation complexities, such as determining which grid nodes to perturb, defining the disturbance characteristics (e.g., magnitude, duration), and establishing the observation period required to assess stability. Basin stability is a concept that entails defining a set of potential disturbances and then running simulations of the system with randomly selected disturbances from within this predefined set. The term "basin volume" is used to quantify this by calculating the proportion of initial conditions that lead to a stable system operation compared to all possible initial conditions. This methodology is more comprehensive than simply assessing stability against specific disturbances, as it inherits and considers all relevant constraints. On the other hand, local stability analysis, also known as linear stability analysis, delves into the examination of dynamic stability in the vicinity of a steady-state operating point within a grid or system. While our current simulation method is fairly effective, there is room for generalization. We acknowledge areas of study that remain unexplored but hold promise as research directions. In our trials, we achieved an accuracy of approximately 80%, which suffices for a general understanding of the system. However, for simulations aimed at identifying pure stability areas, this level of accuracy may be insufficient. One intriguing research avenue is the exploration of cost-sensitive classifiers that prioritize stable design points. Additionally, we employed our knowledge of system symmetry to extract characteristics from initial input data, but questions persist about the optimality of these features and whether they can be automatically discovered. Furthermore, extending this analysis to large grids with more than ten individuals presents significant challenges [28].

5. Conclusion

The decentralization of energy generation and the inclusion of distributed energy storage are reshaping the landscape of the electricity grid. This shift presents both opportunities and challenges and represents a significant transformation in the electricity market, alongside trends like electrification and digitalization. When we refer to "closeness," it doesn't necessarily pertain solely to physical proximity. For instance, a corporation with on-site power generation facilities may not always be physically close to the end user. However, situations can arise where a power plant is indeed physically nearby. What truly matters is the ability to aggregate multiple energy resources. Decentralized energy generation essentially means producing energy closer to where it will be consumed, as opposed to relying on large centralized power plants that distribute electricity across a wide area. By the year 2023, it is projected that 65% of energy companies will have invested in digital technologies and platforms to enhance flexibility services, which could potentially activate up to 35% of the total installed capacity. Smart grids (SGs) are electrical networks that enable a bidirectional flow of electricity and data. They utilize digital communication technology to detect, respond to, and address changes in electricity demand and various challenges. Smart grids also possess self-healing capabilities and empower electricity consumers to actively participate in the grid. The stability of smart grids holds paramount importance in evaluating the effectiveness of smart grid architecture. Therefore, it is crucial to conduct testing and predictions of stability under diverse conditions. This is essential to prevent undesirable instabilities in future smart grid architectures. The integration of renewable energy sources into SGs necessitates intelligent techniques for stability prediction. Several factors come into play, including production, decentralization (distributed energy), regulatory changes, the emergence of presumptive behaviors, the proliferation of micro-generation and micro grids, mandates for renewable energy, and the growing demand for electric vehicle charging infrastructure. In this context, AI plays a vital role in forecasting smart grid stability. Utilizing DL models, we examine complex customer-centric smart grid systems across a wide range of input values, eliminating limitations associated with input value assumptions. Our proposed approach achieves accuracy rates exceeding 95%, offering valuable insights through a simulated system. Notably, fast adaptation emerges as a factor that can enhance overall system stability. This encompasses various aspects such as energy market liberalization, metering evolution, and shifts in electricity demand. Through this method, previously unknown information about smart grids comes to light. For instance, it reveals that systems can maintain stability even if certain individuals adjust their energy consumption with significant delays, and in some scenarios, rapid adaptation is preferable for ensuring stability. The Proposed FSSG-HPDCNN approach accuracy value is 99.88% which is higher than other existing methods like FSSG-WHO, FSSG-PSO, and FSSG-HBO methods.

Reference

[1] B. Schäfer, M. Matthiae, M. Timme, and D. Witthaut, 2015. Decentral smart grid control. *New journal of physics*, *17*(1), p.015002.

[2] B. Schäfer, C. Grabow, S. Auer, J. Kurths, D. Witthaut, and M. Timme, 2016. Taming instabilities in power grid networks by decentralized control. *The European Physical Journal Special Topics*, *225*, pp.569-582.

[3] J.P. Kleijnen, 2018. *Design and analysis of simulation experiments* (pp. 3-22). Springer International Publishing.

[4] A. Ulbig, T.S. Borsche, and G. Andersson, 2014. Impact of low rotational inertia on power system stability and operation. *IFAC Proceedings Volumes*, *47*(3), pp.7290-7297.

[5] D. Manik, D. Witthaut, B. Schäfer, M. Matthiae, A. Sorge, M. Rohden, E. Katifori, and M. Timme, 2014. Supply networks: Instabilities without overload. *The European Physical Journal Special Topics*, *223*, pp.2527-2547.

[6] P. Kundur, 1994. „Power System Stability and Control" McGraw.

[7] W.W. Price, C.W. Taylor, and G.J. Rogers, 1995. Standard load models for power flow and dynamic performance simulation. *IEEE Transactions on power systems*, *10*(CONF-940702-).

[8] T. Nishikawa, and A.E. Motter, 2015. Comparative analysis of existing models for power-grid synchronization. *New Journal of Physics*, *17*(1), p.015012.

[9] K. Schmietendorf, J. Peinke, R. Friedrich, and O. Kamps, 2014. Self-organized synchronization and voltage stability in networks of synchronous machines. *The European Physical Journal Special Topics*, *223*, pp.2577-2592.

[10] O. Kamps, 2014. Characterizing the fluctuations of wind power production by multi-time statistics. In *Wind Energy-Impact of Turbulence* (pp. 67-72). Berlin, Heidelberg: Springer Berlin Heidelberg.

[11] K.C. Budka, J.G. Deshpande, and M. Thottan, 2016. *Communication networks for smart grids*. Springer London Limited.

[12] S.K. Routray, D. Gopal, A. Pallekonda, A. Javali, and S. Kokkirigadda, 2021, January. Measurement, control and monitoring in smart grids using nbiot. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (pp. 229-234). IEEE.

[13] A.A. Freitas, 2014. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, *15*(1), pp.1-10.

[14] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens, 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, *51*(1), pp.141-154.

[15] M.R. Berthold, and K.P. Huber, 1999. Constructing fuzzy graphs from examples. *Intelligent Data Analysis*, *3*(1), pp.37-53.

[16] H. Pierreval, 1996. A metamodeling approach based on neural networks. *Int. Journal in Computer Simulation*, *6*(3), p.365.

[17] T. Yoshida, and S. Nakasuka, 1989, December. A dynamic scheduling for flexible manufacturing systems: hierarchical control and dispatching by heuristics. In *Proceedings of the 28th IEEE Conference on Decision and Control*, (pp. 846-852). IEEE.

[18] H. Pierreval, and H. Ralambondrainy, 1992. A simulation and learning technique for generating knowledge about manufacturing systems behavior. *Artificial Intelligence in Operational Research*, pp.239-252.

[19] J.D. McCalley, S. Wang, Q.L. Zhao, G.Z. Zhou, R.T. Treinen, and A.D. Papalexopoulos, 1997. Security boundary visualization for systems operation. *IEEE Transactions on Power Systems*, *12*(2), pp.940-947.

[20] B. Jayasekara, and U.D. Annakkage, 2006. Derivation of an accurate polynomial representation of the transient stability boundary. *IEEE transactions on Power Systems*, *21*(4), pp.1856-1863.

[21] C. Li, 2020. Designing a short-term load forecasting model in the urban smart grid system. *Applied Energy*, *266*, p.114850.

[22] T. Alquthami, M. Zulfiqar, M. Kamran, A.H. Milyani, and M.B. Rasheed, 2022. A performance comparison of machine learning algorithms for load forecasting in smart grid. *IEEE Access*, *10*, pp.48419-48433.

[23] G. Hafeez, I. Khan, S. Jan, I.A. Shah, F.A. Khan, and A. Derhab, 2021. A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid. *Applied Energy*, *299*, p.117178.

[24] S.F. Ali, and A. Ramaswamy, 2009. Optimal fuzzy logic control for MDOF structural systems using evolutionary algorithms. *Engineering Applications of Artificial Intelligence*, *22*(3), pp.407-419.

[25] X. Liu, Z. Xiao, R. Zhu, J. Wang, L. Liu, and M. Ma, 2020. Edge sensing data-imaging conversion scheme of load forecasting in smart grid. *Sustainable Cities and Society*, *62*, p.102363.

[26] M. Ali, M. Adnan, and M.Tariq, 2019. Optimum control strategies for short term load forecasting in smart grids. *International journal of electrical power & Energy systems*, *113*, pp.792-806.

[27] M. Zulfiqar, M. Kamran, M.B. Rasheed, T. Alquthami, and A.H. Milyani, 2023. A hybrid framework for short term load forecasting with a navel feature engineering and adaptive grasshopper optimization in smart grid. *Applied Energy*, *338*, p.120829.

[28] F. Zheng, Q. Yan, V.C. Leung, F.R. Yu, and Z. Ming, 2022. HDP-CNN: Highway deep pyramid convolution neural network combining word-level and character-level representations for phishing website detection. *Computers & Security*, *114*, p.102584.