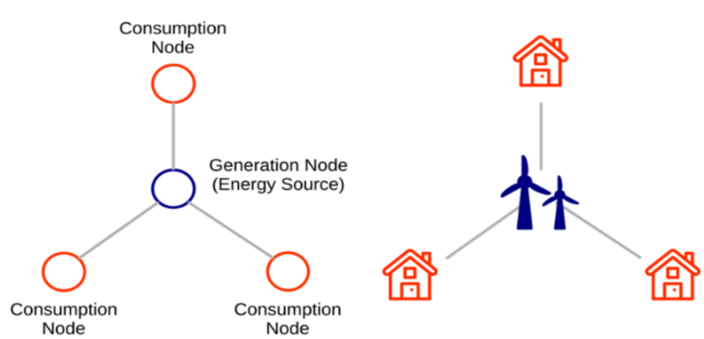
PREDICTING SMART GRID STABILITY USING ARTIFICIAL NEURAL NETWORKS

- **INTRODUCTION:**

Renewable sources of energy are on the rise as climate change rapidly affects various aspects of our planet and the livelihood of its inhabitants. A huge accelerant to climate change has been our grand usage of non-renewable energy sources such as fossil fuels. To combat this issue, we have to focus on mobilizing the usage of renewable energy. To replace the traditional use of conventional energy sources, the governments have pushed for renewable sources, which give the power of producing energy from the consumer end. Thus, **smart grids,** which facilitate the bidirectional transmission of energy, have become popular.

Our data is centered around **Decentral Smart Grid Control (DSGC) systems**. This methodology mainly monitors the grid's frequency. It consists of four primary operations concerning electricity: **generation**, **transmission**, **distribution,** and **control**. This grid system is considered "smart" due to communication between the utility and its customers and sensing along the transmission lines.

This is done when the smart grids gather consumer demand data and compare it to the supply conditions at the time. They then send the proposed price to the consumers, so consumers can make knowledgeable decisions on how to use their power. The structure of the DSGC system is based on a **4-node star** architecture which includes one power source and three consumption nodes. The inputs/features are the **total power balance,** the **reaction time** of the consumers to the price changes, and the **energy price elasticity**.



We predict the stability of smart grids based on the **Artificial Neural Network (ANN**) models and compute accuracies based on different structures and hyperparameters.

- **<u>DATA DESCRIPTION:</u>**

In this dataset, there are a total of **10,000 observations**. The data is augmented by using the DSGC system for **60,000 observations**. The augmented data helps in providing better predictions. There are **two dependent variables** and **12 primary predictive features**.

- Feature 1-4:
    - **tau1, tau2, tau3, tau4**
        - These represent the reaction time of each network participant
        - tau1 corresponds to the supplier node
        - tau2, tau3, and tau4 correspond to the consumer nodes
        - Values are within the range of 0.5 to 10
- Feature 5-8:
    - **p1, p2, p3, p4**
        - These represent the nominal power produced (positive) or consumed (negative) by each network participant
        - p1 (supplier node) = –(p2 + p3 + p4)
        - Values are within the range of –2.0 to –0.5 for consumer nodes
- Feature 9-12:
    - **g1, g2, g3, g4**
        - These represent the price elasticity coefficient for each network participant
        - g1 corresponds to the supplier node
        - g2, g3, and g4 correspond to the consumer nodes
        - Values are within the range of 0.05 to 1.00

We also have two dependent variables within this dataset:

- **stab** – the maximum real part of the characteristic differential equation root
    - Positivity indicates that the system is linearly unstable
    - Negativity suggests that the system is linearly stable
- **stabf** – a categorical binary label of "stable" or "unstable."

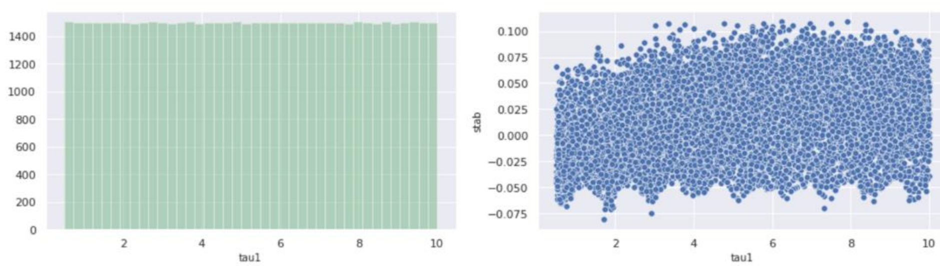
There happens to be a direct relationship between the 'stab' and 'stabf' variable, which is that if stab is less than or equal to 0, 'stabf' will be set to "stable" and vice versa. We drop the variable 'stab' altogether, which leaves 'stabf' as the sole dependent variable.

- **<u>DATA PREPROCESSING</u>:**

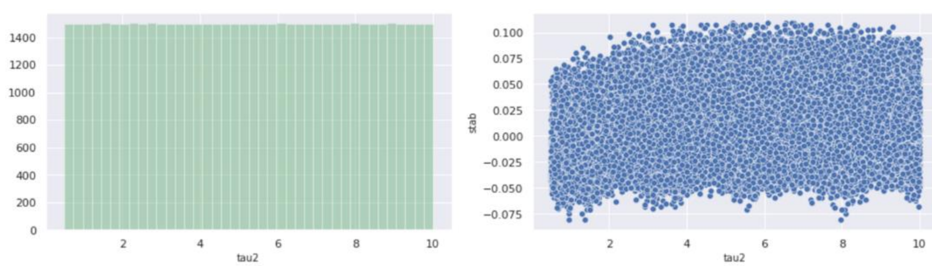
In this dataset, all features used in the model are numeric, including tau1 to tau4, p1 to p4, and g1 to g4. We also scaled the data values using the **Standard Scaler**, so the effect of gradient descent will be applied the same to all features. We split the data into **90%** as the **train,** which includes **54,000** observations, and **10%** as **test consisting** of **6,000** observations.

- **<u>DATA VISUALIZATIONS</u>:**

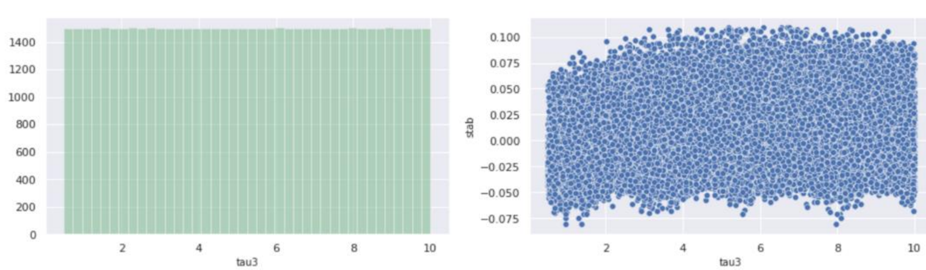
Distribution patterns and the relationship with 'stab' for feature tau1



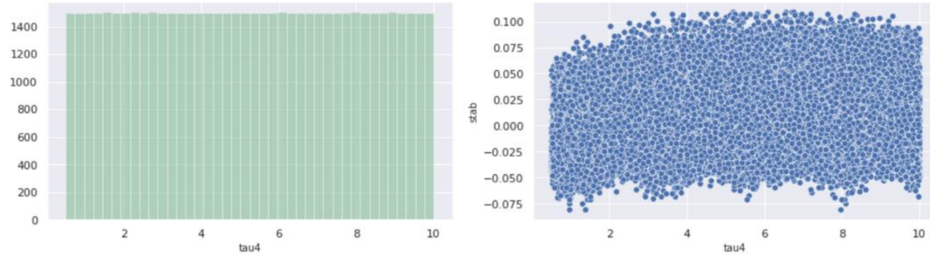
Distribution patterns and the relationship with 'stab' for feature tau2



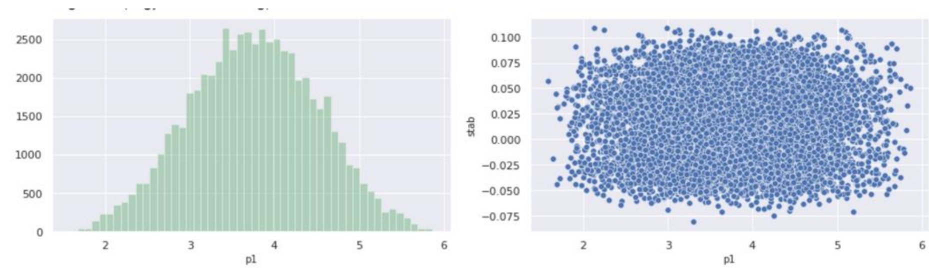
Distribution patterns and the relationship with 'stab' for feature tau3



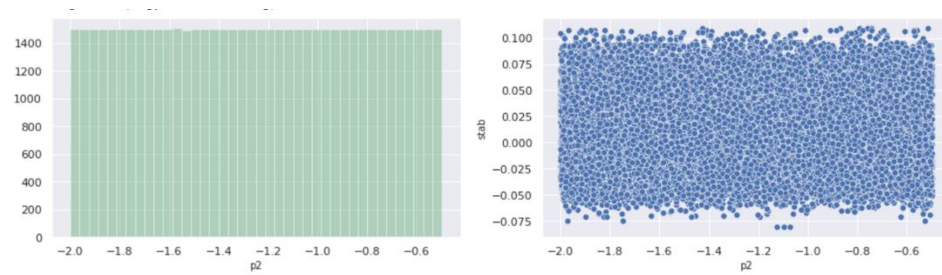
Distribution patterns and the relationship with 'stab' for feature tau4



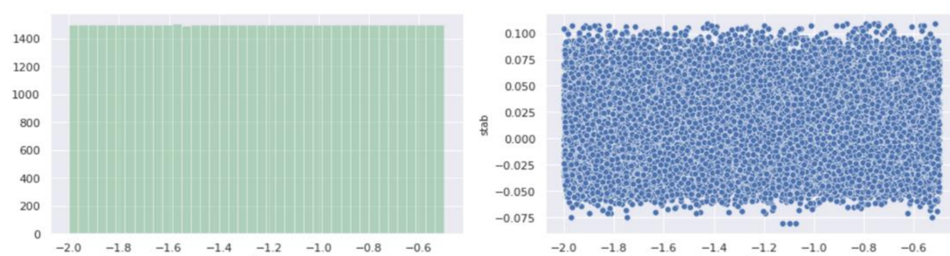
Distribution patterns and the relationship with 'stab' for feature p1



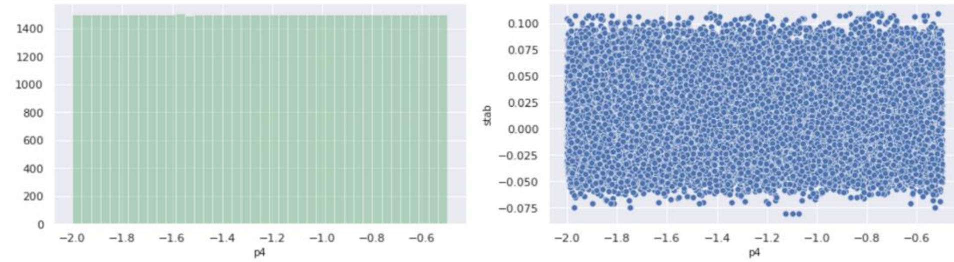
Distribution patterns and the relationship with 'stab' for feature p2



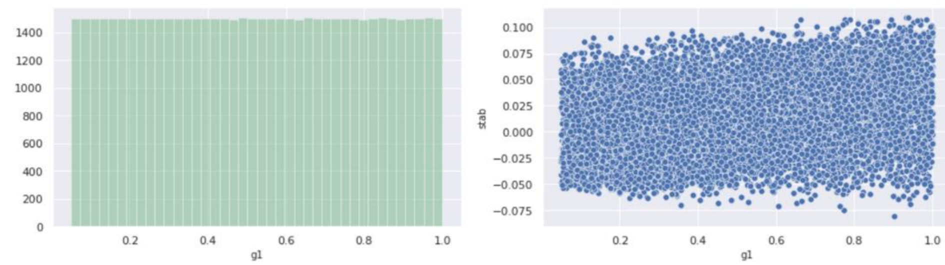
Distribution patterns and the relationship with 'stab' for feature p3



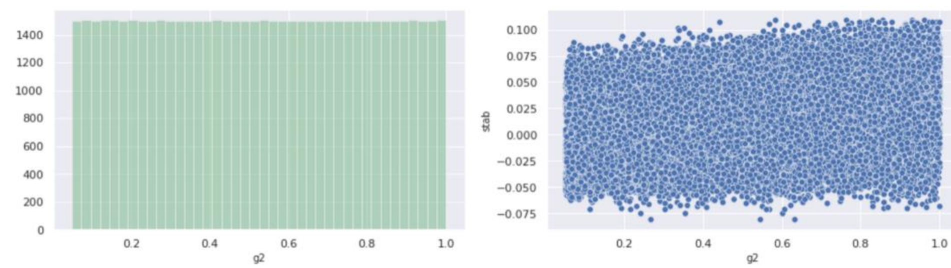
Distribution patterns and the relationship with 'stab' for feature p4



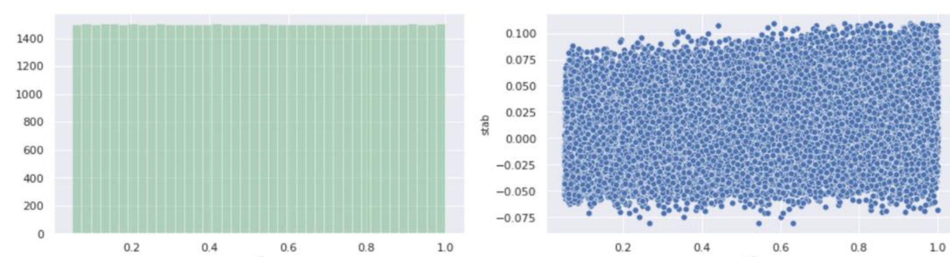
Distribution patterns and the relationship with 'stab' for feature g1



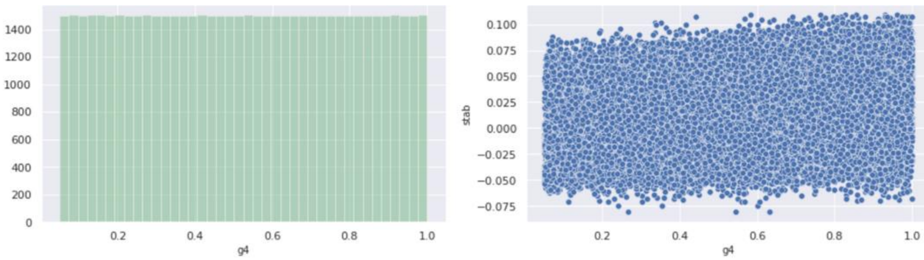
Distribution patterns and the relationship with 'stab' for feature g2



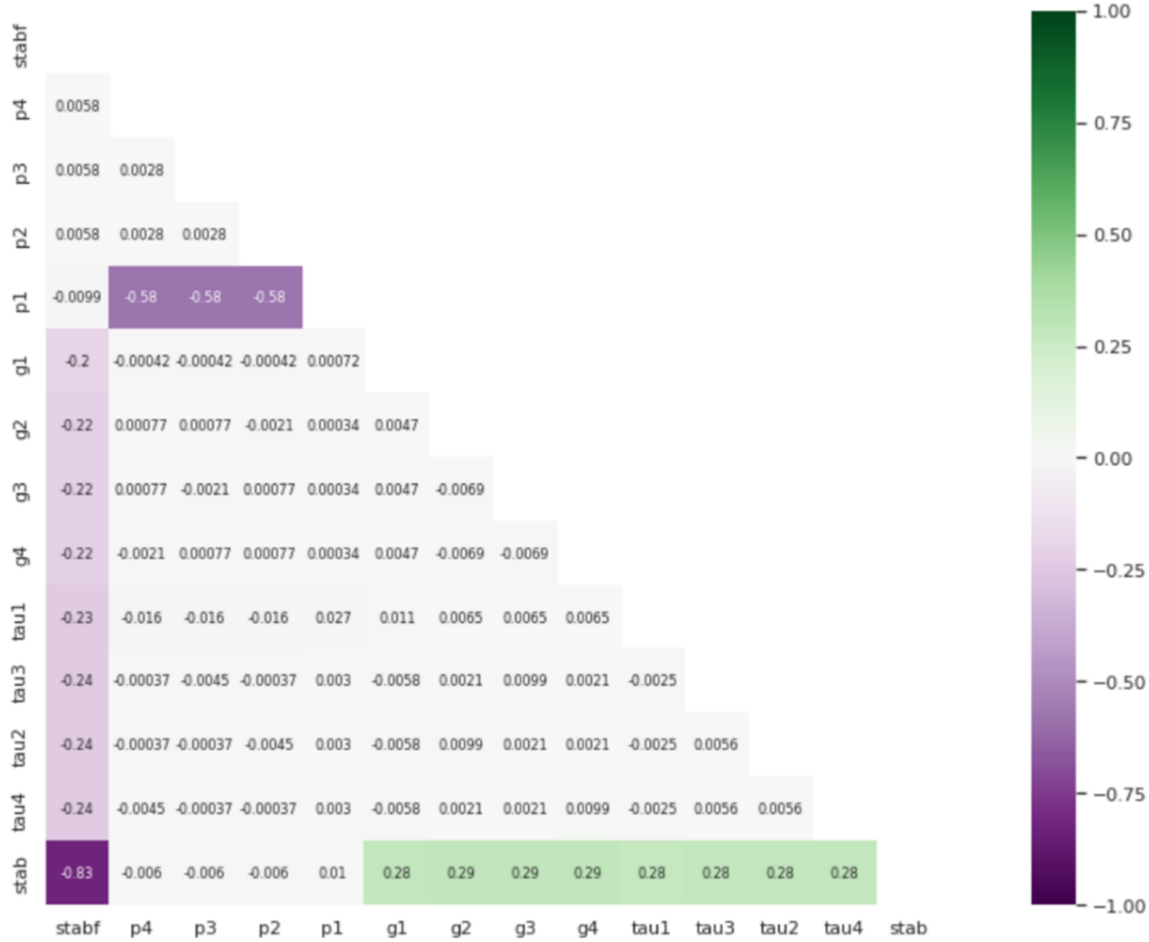
Distribution patterns and the relationship with 'stab' for feature g3



Distribution patterns and the relationship with 'stab' for feature g4



- **<u>CORRELATION MAP</u>:**

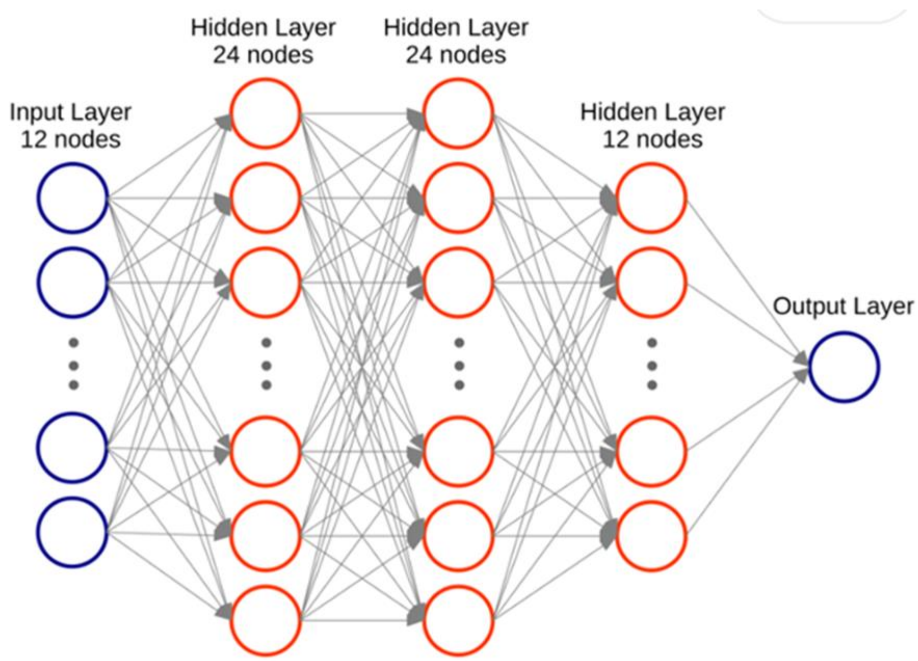


The correlation map shows us that the dependent variables' stab' and 'stabf' are highly correlated, making sense as the' stabf' values are based on the values of 'stab' itself. Therefore, we would be removing the variable stab to make the analysis and modeling relevant to the problem.

We also see a positive correlation between the stability factor and the reaction time and price elasticity variables which is intuitive because of the nature of those variables.

➢ <u>**DATA MODELING**</u>:

Our model consists of three hidden layers. The **input layer** has **12 nodes**, and the **output layer** is a **single node**. The **first two hidden layers** include **24 nodes** each, while the **last hidden layer** only has **12 nodes**. For the hidden layers, we use the **"ReLU"** function and the "**Sigmoid"** function for the output layer.

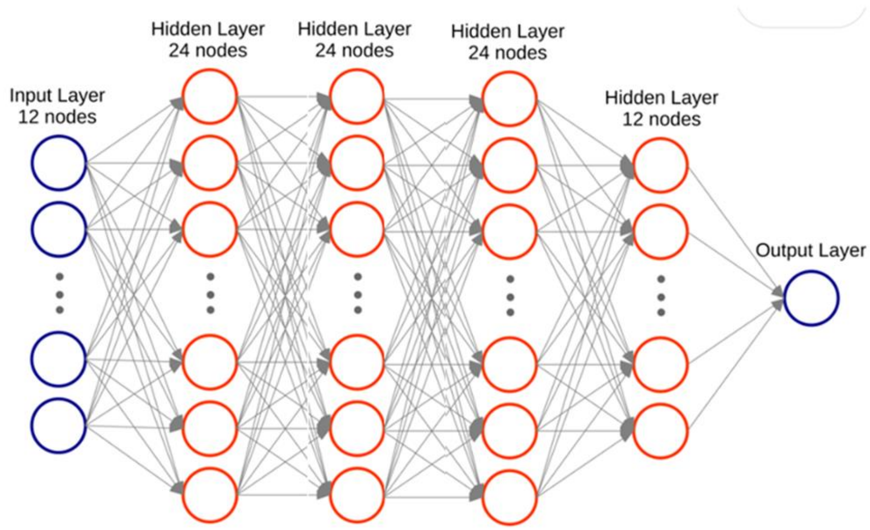


We use the "**Adam"** optimizer within the model and **the "Binary Cross-entropy"** loss function to assess the functionality of our model. Additionally, we use the **10 Fold Cross Validation** to double-check our results. We increase the **epochs** from **10-20-50**. We use the **confusion matrix** as our evaluation metric below, which gives us an accuracy of **97.73%**.

	Predicted Unstable	Predicted Stable
Actual Unstable	3800	34
Actual Stable	102	2064

- **MODIFICATION:**

We added another **24 nodes** as its **dense layer** and ran it on 10 – 20 – 50 epochs to see if our accuracy would improve, increasing our accuracy to **97.87%**. The new structure and confusion matrix is added below.



	Predicted Unstable	Predicted Stable
Actual Unstable	3802	32
Actual Stable	96	2070

- **MODEL COMPARISON:**

Below is a table that shows the parameters of the models we have used and their effectiveness.

Augmented Data Set (60,000)					
Architecture	Folds	Epochs	Confusion Matrix		Accuracy
			3769	65	
24-24-12-1	10	10	100	2066	97.25%
			3772	62	
24-24-12-1	10	20	88	2078	97.50%
			3800	34	
24-24-12-1	10	50	102	2064	97.73%

Augmented Data Set (60,000)					
Architecture	Folds	Epochs	Confusion Matrix		Accuracy
			3747	87	
24-24-24-12-1	10	10	80	2086	97.22%
			3776	58	
24-24-24-12-1	10	20	91	2075	97.52%
			3802	32	
24-24-24-12-1	10	50	96	2070	97.87%

- **INFERENCE:**

From the model comparison table above, we can infer that the results seem to improve as we increase the number of hidden layers within a neural network and raise the number of epochs. Additionally, using augmented data helps produce better accuracy when predicting the stability of a smart grid. If this model can be scaled up and improved upon, then we would be able to predict more accurately and dynamically whether different 4-star node systems are stable or not.

➢ **REFERENCES:**

- *"Taming instabilities in power grid networks by decentralized control"* (B. Schäfer, et al, The European Physical Journal, Special Topics, 2016, 225.3: 569-582)

- *"Towards Concise Models of Grid Stability"* (V. Arzamasov, K. Böhm and P. Jochem)