EEE AND UID

AI BASED REAL TIME BLACKOUT PREVENTION USING ANN

TEAM MEMBERS:
CH.VAISHNAVI KRISHNA-CB.SC.U4AIE24108
NETHRA.R-CB.SC.U4AIE24147
SAHAANA SHRI.SK-CB.SC.U4AIE24149
MOUNIKA.CH-CB.SC.U4AIE24169

INTRODUCTION:

➤*Power blackouts can cause major problems, including disruptions in daily life and financial losses. Many power systems struggle to predict failures in advance, leading to unexpected outages.*

➤*Our project uses Artificial Neural Networks (ANNs) to analyze power data and predict blackouts before they happen. By doing this, we can prevent power failures, improve grid stability, and make electricity supply more reliable.*

➤*This smart system will help mini-grids work better by making sure energy is used efficiently and reducing sudden power cuts.*

OBJECTIVES:

1.Prevent Blackouts:

➤Use AI technology to predict and stop power failures before they happen.

2.Improve Power Stability:

➤Keep the electricity flow steady by fixing power changes and overloads.

3.Monitor in Real-Time:

➤Continuously track and analyze power usage to make quick decisions.

PROBLEM STATEMENT:

➤ *Mini grids face frequent blackouts due to demand fluctuations and unexpected failures.*

➤ *Traditional monitoring methods are not efficient in preventing failures before they occur.*

Methodology:

1. ***Dataset Loading:***

- *The dataset is loaded from Google Drive using Pandas. Column names are cleaned to remove special characters and spaces to ensure consistency.*

2. ***Handling Missing Values:***

- *Columns with more than 30% missing values are dropped to enhance data quality.*
- *Median imputation is used for numerical missing values.*

3. ***Feature Engineering:***

- *Datetime Processing: Outage start and restoration timestamps are converted to numerical format by computing the number of days from the earliest date in the dataset.*
- *Categorical Encoding: Categorical data is converted into numerical form using Label Encoding and One-Hot Encoding where applicable.*
- *Feature Scaling: StandardScaler is used to normalize numerical features, ensuring consistent feature magnitudes.*

4.Handling Class Imbalance:

- *SMOTE (Synthetic Minority Over-sampling Technique) was applied to generate synthetic samples, ensuring a balanced dataset for training.*

5.Splitting Data:

- *The dataset is split into 80% training and 20% testing to ensure a robust evaluation.*
- *The training set is used to develop the model, while the test set evaluates generalization performance.*

2. ANN Model Development:

Design a 3-layer feedforward neural network (ANN) with:

- *Input Layer: Takes power grid parameters.*
- *Hidden Layers: Extract patterns using ReLU activation.*
- *Output Layer: Uses Sigmoid activation for binary classification.*

ANN MODEL& ARCHITECHURE:

1. Input Layer:

- *The input layer is the first layer of the neural network.*
- *It takes raw data as input (features) and passes it to the next layer.*
- *Number of neurons = Number of features (columns) in your dataset.*

2.Hidden Layer:

a) Dense Layer (Fully Connected Layer)

- *Each neuron in one layer connects to every neuron in the next layer.*
- *Performs the weighted sum of inputs and applies an activation function.*

b) Batch Normalization Layer

- *Normalizes the outputs of the previous layer for better and faster learning.*

ANN contd

c) Activation Functions:

- Activation functions introduce non-linearity in the network, helping the model learn complex relationships.
- ReLU (Rectified Linear Unit) is commonly used:
  - ReLU(x) = max(0, x) (keeps positive values and removes negative ones)
  - Example: layers.Dense(256, activation='relu')

3. Output Layer

- The final layer of the ANN that produces predictions.
- Uses an activation function to determine the output type:
  - Sigmoid (for binary classification like blackout prediction):
    - Outputs values between 0 and 1 (probability of blackout occurring).
    - Example: layers.Dense(1, activation='sigmoid')

Functions used

- *Sequential() – Initializes a linear stack of layers for building the neural network.*
- *Dense(units, activation) – Fully connected layer where each neuron connects to all neurons in the previous layer.*
- *Dropout(rate) – Randomly drops neurons during training to prevent overfitting.*
- *fit(X_train, y_train, epochs, batch_size, validation_data, callbacks) – Trains the model using the training dataset over multiple epochs.*
- *evaluate(X_test, y_test) – Measures the model's performance on unseen test data.*

Training Process

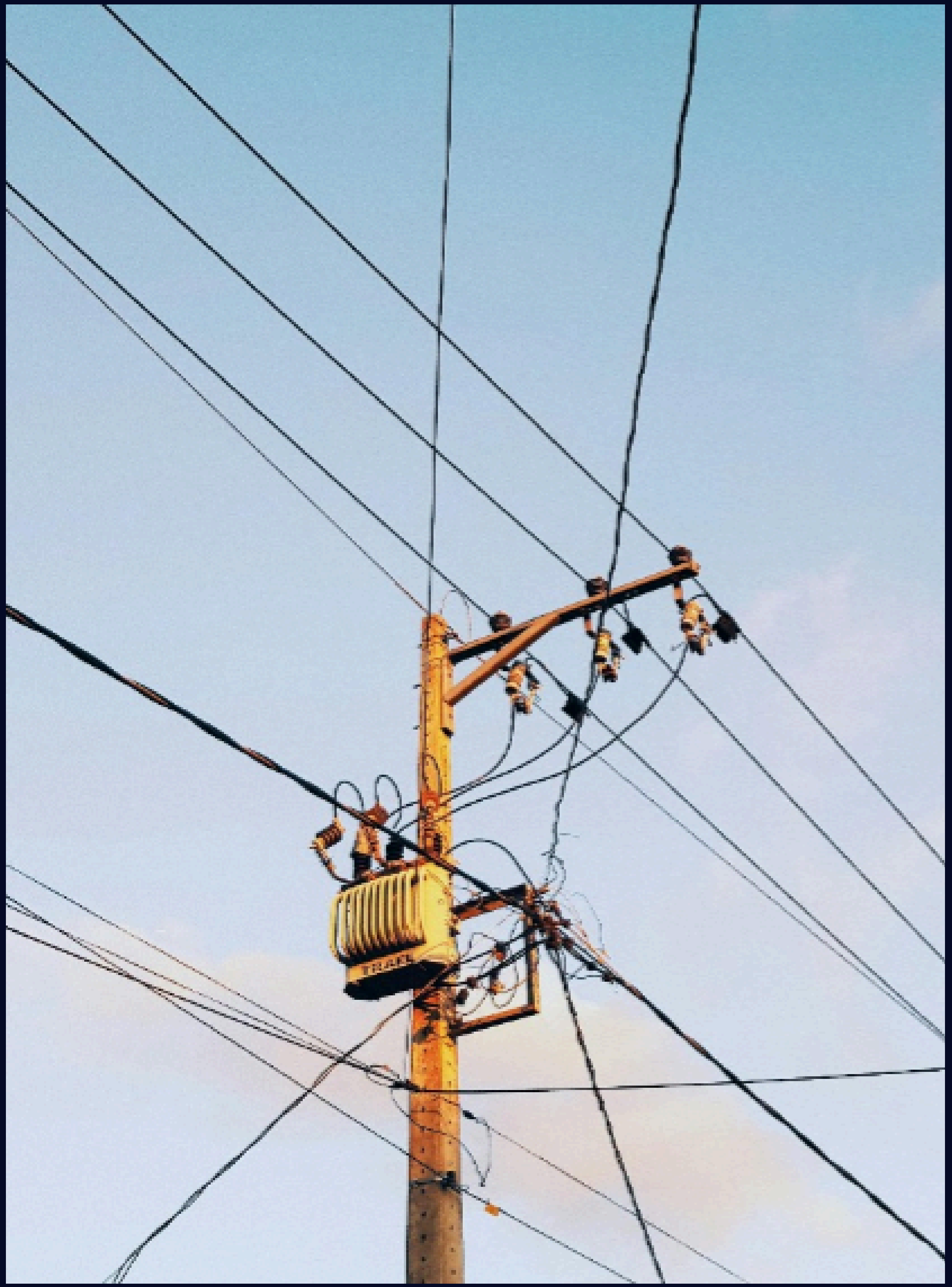
Model Training & Optimization:

a. Training Process

- *The model is trained using historical blackout data.*
  Back propagation algorithm updates weights to reduce prediction errors.
- *Batch Size: 65 (for stable training) .*
- *Epochs: 100 (to ensure sufficient learning).*

b. Optimization Techniques

- *Optimizer: Adam Optimizer (adjusts learning rate dynamically).*
- *Dropout Layers: Helps prevent overfitting by randomly deactivating neurons.*



Prediction & Deployment:

1. Model Evaluation:

➤*Model is tested on unseen validation data for accuracy measurement. Validation accuracy represents how well the model generalizes to real scenarios.*

2. Real-Time Blackout Prediction:

➤*Trained ANN model predicts blackouts in test data*

Results & Analysis:

Testing and Evaluation:

Predictions are made using the trained ANN, with probability outputs converted to binary classifications (threshold = 0.5).

- *Performance Metrics:*
- *Test Accuracy: 92.47%*
- *Precision: 92.58% (indicates how many predicted blackouts were actual blackouts)*
- *Recall: 92.22% (measures the model's ability to detect actual blackouts)*
- *F1 Score: 92.40% (harmonic mean of precision and recall, ensuring balanced performance)*

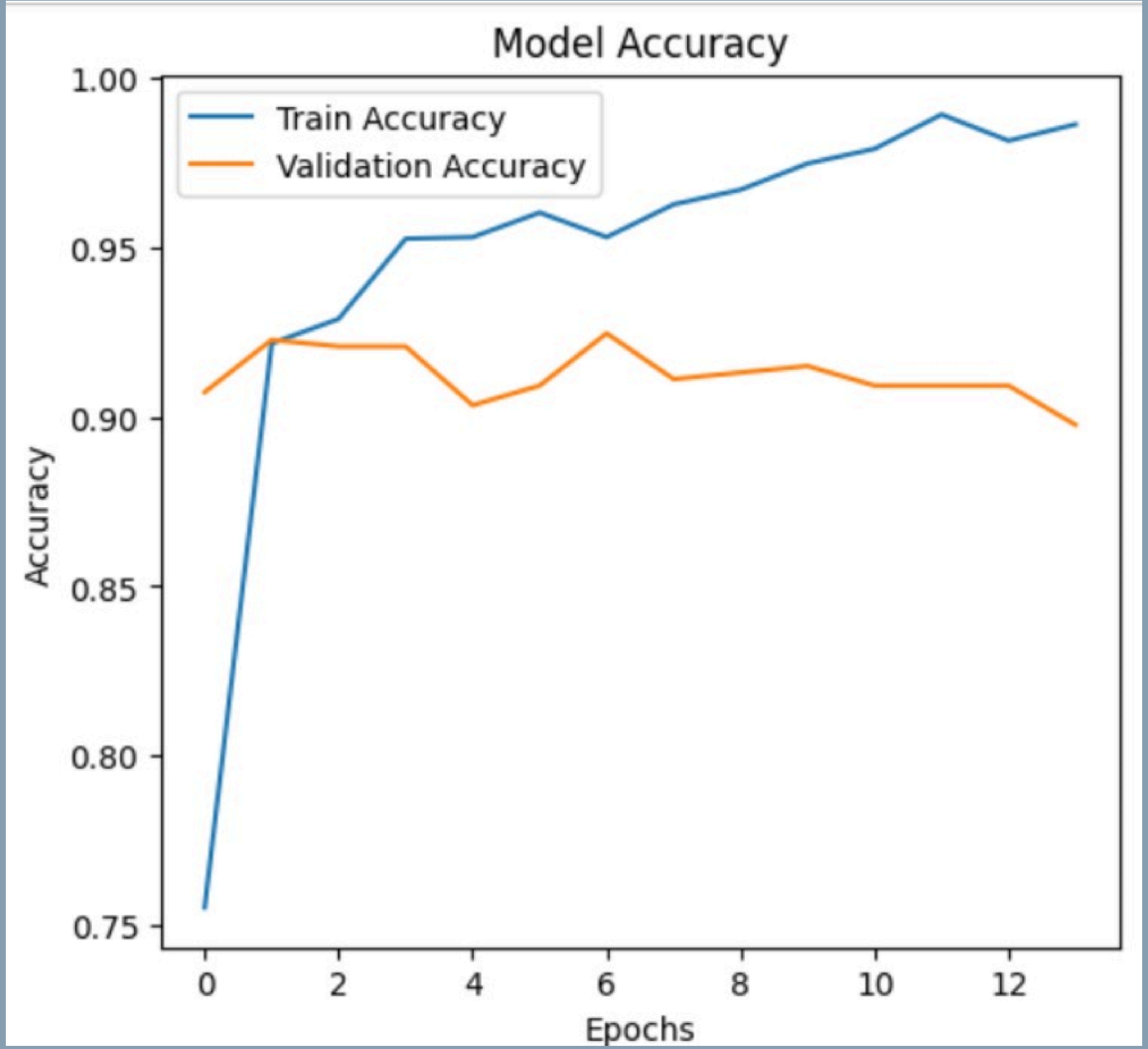
Model Training Performance:

➤ The ANN model was trained for 100 epochs, and the training and validation accuracy showed a stable increase.

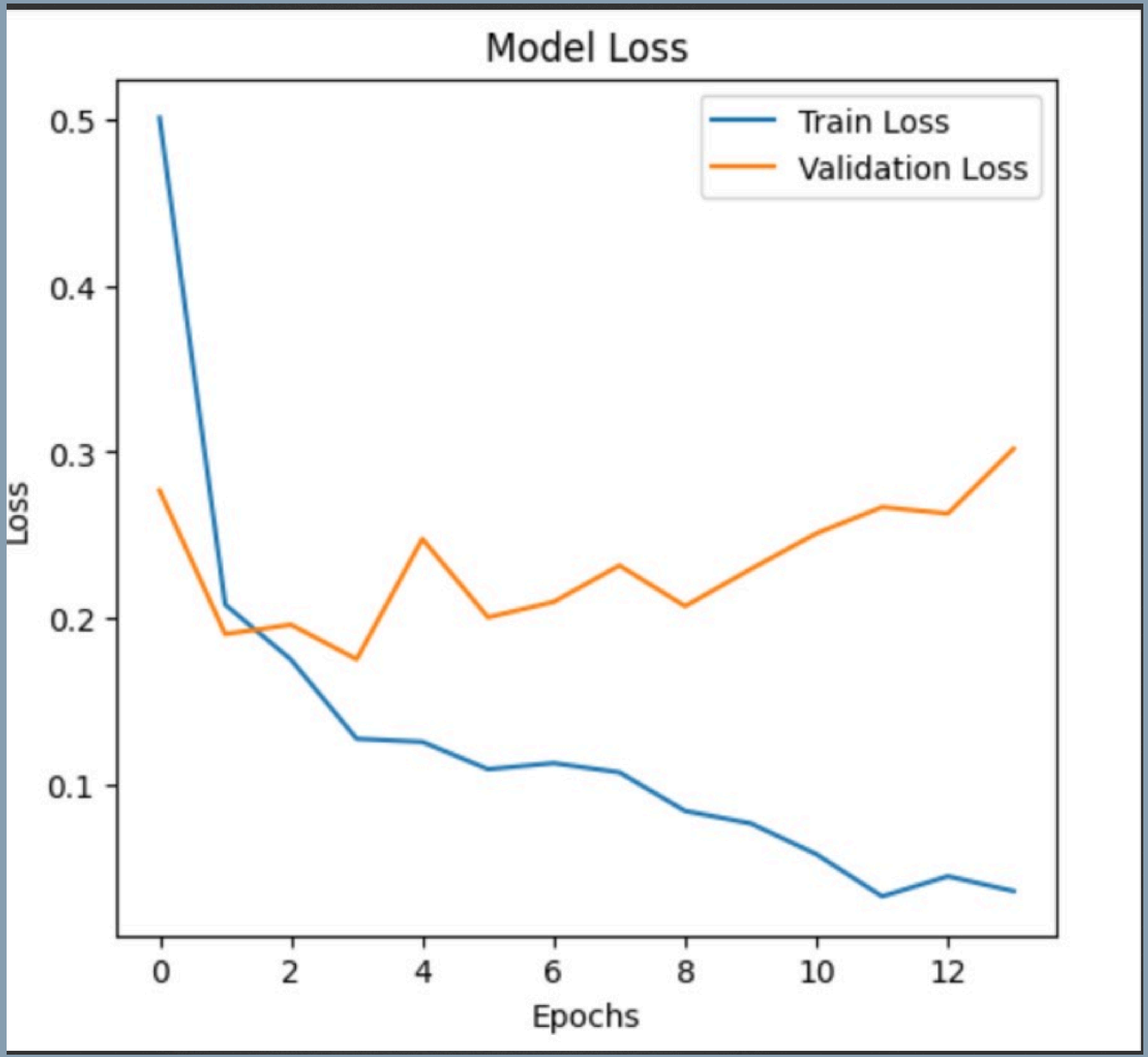
Graphical Analysis:

➤ *Graphs such as Accuracy vs. Epochs were plotted to analyze model performance.*

RESULTS:



Model Accuracy
Train Accuracy
Validation Accuracy
Accuracy
Epochs
1.00
0.95
0.90
0.85
0.80
0.75
0
2
4
6
8
10
12



Model Loss
Train Loss
Validation Loss
Loss
Epochs
0.5
0.4
0.3
0.2
0.1
0
2
4
6
8
10
12

Limitations:

➤*Although the model perform ed well, certain real-time factors such as sudden voltage spikes or grid malfunctions were challenging to predict.*

➤*Further optimization and real-time data integration could enhance performance.*

FUTURE SCOPE:

➤Advanced AI & Deep Learning Models – Enhance accuracy using Recurrent Neural Networks (RNNs) & Transformers.

➤Automated Self-Healing Grids – Develop AI-driven systems to auto-correct power fluctuations and prevent failures.

CONCLUSION:

➤This project successfully implemented an AI-based real-time blackout prevention system using Artificial Neural Networks (ANNs) in a mini-grid environment. The model was able to predict potential blackouts with high accuracy, reducing power losses and enhancing grid stability.

➤By leveraging historical and real-time data, our system effectively analyzed critical parameters such as voltage fluctuations, load demand, and grid frequency variations. The results demonstrated that ANN-based prediction models can play a crucial role in preventing grid failures and improving energy reliability.


AI
THANK YOU