ML-Based Fault Detection in Microgrids

Introduction

The increasing integration of renewable energy sources and distributed generation has led to the widespread adoption of microgrids, which offer enhanced energy resilience, efficiency, and reliability. However, microgrids also introduce complex operational challenges, including fault detection and voltage fluctuations due to frequency variations.

These challenges, if not properly addressed, can lead to **power outages, equipment damage**, and **reduced system stability**.

For example, recently Sri Lanka scrambled to restore full capacity to its national grid for about six hours after a monkey triggered a widespread blackout over the weekend, disrupting power supply to the island's 22 million people.

Traditional fault detection methods rely on rule-based techniques and threshold monitoring, **often failinig to detect early-stage faults or adapt to dynamic grid conditions.** Similarly, conventional voltage regulation mechanisms struggle to maintain stability under variable frequency conditions, **especially in microgrids with high penetration of renewables.**

To address these issues, **Machine Learning (ML)-based models offer a promising solution**. ML can detect anomalies in grid operations, classify fault types, and predict the optimal voltage levels required to maintain stability under varying frequencies. By leveraging historical data, real-time sensor inputs, and predictive analytics, ML can significantly enhance the reliability and efficiency of microgrid operations.

This study proposes an ML-based framework that
(1) detects faults in microgrids using supervised and unsupervised learning techniques and
(2) predicts voltage levels required to maintain grid stability for different frequency variations.
The proposed approach aims to provide a scalable, data-driven solution that can improve microgrid resilience, optimize energy distribution, and minimize disruptions.

The rest of this report is structured as follows: Section 2 provides a literature review on existing fault detection and voltage regulation techniques. Section 3 discusses the methodology, including data collection, model selection, and implementation. Section 4 presents the results and analysis, while Section 5 concludes the study with potential improvements and future research directions.

Section 2

Grid fault detection and voltage regulation have long been fundamental aspects of power system operation. Traditional methods were employed to detect faults and regulate voltage, relying on mathematical models, physical measurements, and rule-based systems.

Traditional fault detection techniques primarily used analytical and signal processing approaches. These included:

- **Overcurrent and Distance Relays**: These protection devices operate based on predefined thresholds for current and voltage. Distance relays use impedance measurement to locate faults.
- **Fourier and Wavelet Transform Techniques**: These mathematical tools analyze power system signals to detect abnormal frequency components indicative of faults.
- **Kalman Filtering and State Estimation**: These methods estimate system parameters in real time, helping to identify deviations caused by faults.
- **Phasor Measurement Units (PMUs) and Synchrophasors**: Used for high-speed monitoring and fault detection by analyzing synchronized voltage and current phasors.
- **Expert Systems and Rule-Based Approaches**: These systems utilize predefined rules and decision trees to identify faults based on observed symptoms.

Voltage regulation is essential to maintain power quality and system stability. The following techniques have been commonly used:

- **Automatic Voltage Regulators (AVRs)**: AVRs maintain voltage levels within acceptable limits using feedback control mechanisms.
- **Tap-Changing Transformers**: These transformers adjust their winding ratios to compensate for voltage variations in the grid.
- **Capacitor Banks and Reactor Compensation**: Shunt capacitors and reactors provide reactive power compensation to stabilize voltage levels.
- **Load Frequency Control (LFC) and Excitation Control**: LFC maintains grid frequency, indirectly contributing to voltage stability, while excitation control regulates generator voltage output.
- **Flexible AC Transmission Systems (FACTS)**: Devices such as Static VAR Compensators (SVC) and Static Synchronous Compensators (STATCOM) improve voltage stability by adjusting reactive power flow.

While traditional methods have been effective, they face several challenges:

- **Slow Response Time**: Many conventional systems rely on predefined thresholds and are not adaptive to rapid changes.
- **Limited Fault Classification Capabilities**: Rule-based and threshold-based approaches struggle with complex or unknown fault scenarios.
- **Dependence on Manual Tuning**: Many methods require continuous monitoring and human intervention to adjust settings.
- **Scalability Issues**: As power systems grow in complexity, conventional methods may struggle to handle large-scale data efficiently.

Conventional fault detection and voltage regulation methods have relied on well-established mathematical, rule-based, and control system techniques. These methods have provided stability and reliability for power systems but come with certain limitations, such as slower adaptability to changing grid conditions and manual intervention requirements.

.Section 3

3.1 Data Collection

To train an Artificial Neural Network (ANN) for fault detection in a microgrid, we collected data through simulated experiments. The microgrid model consists of a **single-phase AC network** supplying power to three houses. Key electrical parameters were monitored and recorded over a **24-hour period**, with data points logged every hour.

Key features recorded:

1. **Secondary Current (I_sec)** – The current in the circuit, which spikes during faults.
2. **Load Power (P_L)** – The power consumed by loads, which drops significantly when a fault occurs.
3. **Time of Fault Activation** – Each simulated fault was introduced at a specific time of the day.
4. **Fault Label** – The location of the fault, indicating which breaker (house) was affected.

To simulate faults, three **circuit breakers** were introduced before each house. These breakers were activated sequentially for 30 minutes each, creating **96 fault scenarios** for training data. The dataset was then **preprocessed**, ensuring proper scaling and encoding.

3.2 Model Selection

Given the complexity of fault detection, an **Artificial Neural Network (ANN)** was chosen due to its ability to learn complex patterns in non-linear data. The ANN model was designed as a **classification model**, predicting the fault location based on input features.

Architecture of the ANN model:

- **Input Layer (27 neurons):**
    - 4 primary input features (I_sec, P_L, time, and fault label).
    - The "time" feature was **one-hot encoded**, increasing input neurons to **27**.
- **Hidden Layers:**
    - **First hidden layer:** 16 neurons, **ReLU activation**, followed by a **dropout layer** to prevent overfitting.
    - **Second hidden layer:** 8 neurons, **ReLU activation**.
- **Output Layer (4 neurons, Softmax activation):**
    - Predicts fault location among four classes: No fault, House 1, House 2, House 3.

The **Adam optimizer** was used to optimize the model, with **500 epochs** to ensure convergence

3.3 Implementation

The implementation followed these steps:

1. **Data Preprocessing:**
    - Standardized input features using **StandardScaler**.
    - Encoded categorical time values.
    - Split data into **training and testing sets** (80%-20% split).
2. **Model Training:**
    - The ANN was trained using **TensorFlow/Keras**.
    - Loss function: **Categorical Cross-Entropy**.
    - The model achieved an accuracy of **~71.74%** after training.
3. **Fault Detection in Real-Time:**
    - New sensor data was passed to the trained ANN.
    - The model **predicted fault occurrence and location**.
    - The system could automatically **trigger alerts** when faults were detected.

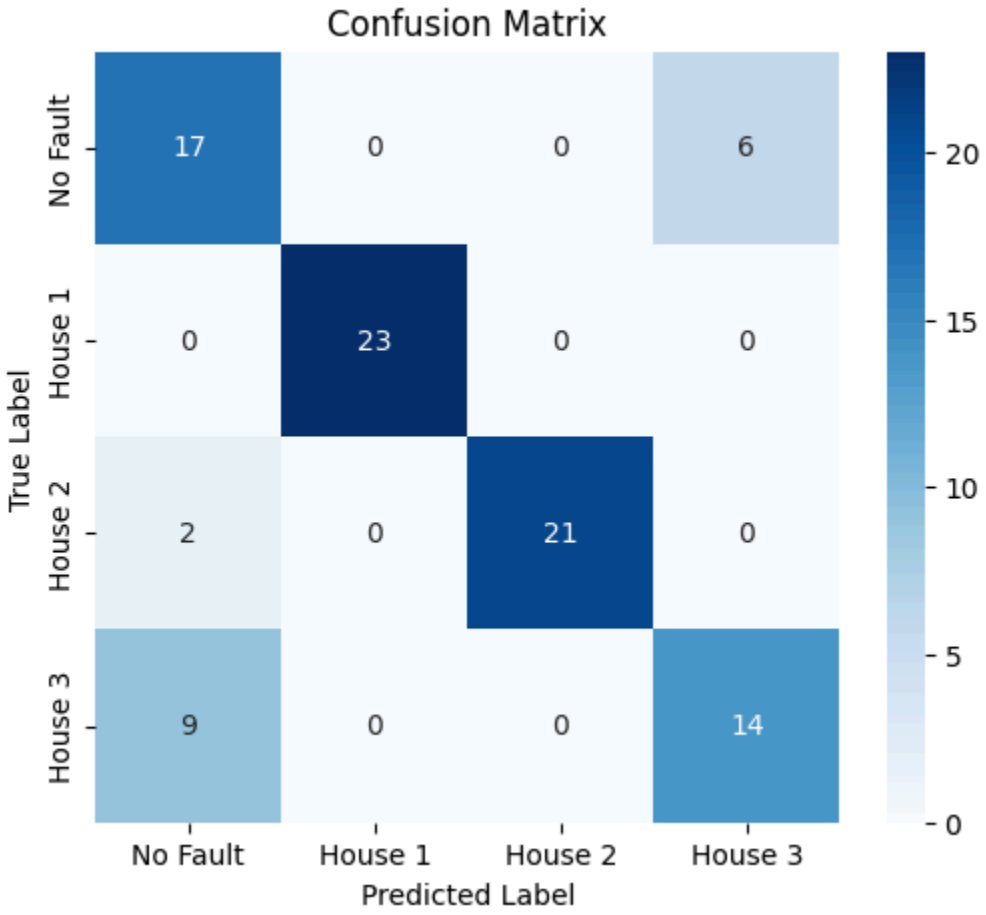
Section 4

4.1 Confusion Matrix Analysis

The confusion matrix presented in Figure 4.1 provides insights into the model's classification performance. The matrix shows the number of correctly and incorrectly classified instances for each category:

- **No Fault**: 17 correctly classified, 6 misclassified as "House 3."
- **House 1**: 23 correctly classified, no misclassifications.
- **House 2**: 21 correctly classified, 2 misclassified as "House 3."
- **House 3**: 14 correctly classified, 9 misclassified as "No Fault."

From this, it is evident that **House 3 faults are often confused with No Fault**, indicating potential overlapping characteristics in their feature space. The model performs well in detecting **House 1 and House 2 faults** with high precision.

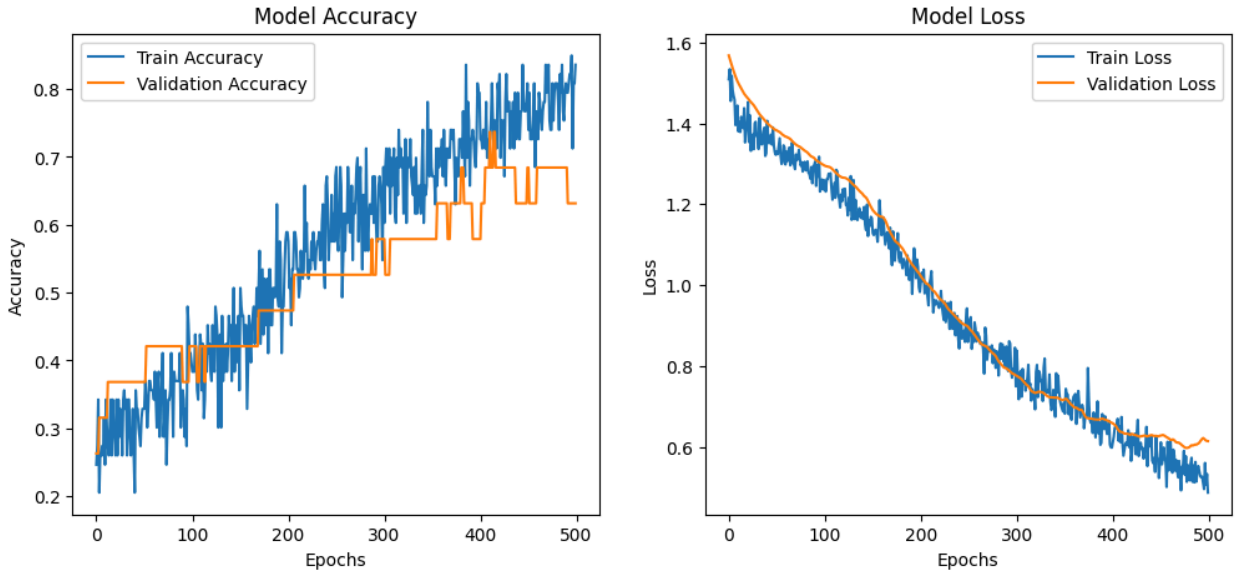


4.2 Model Performance Evaluation

4.2.1 Accuracy and Loss Trends

The accuracy and loss graphs in Figure 4.2 demonstrate the training dynamics of the ANN model:

- **Accuracy Graph**: Shows a steady increase in training and validation accuracy, stabilizing around **0.8** after 400 epochs.
- **Loss Graph**: Exhibits a consistent downward trend, indicating improved learning and reduced prediction errors. The loss decreases smoothly, suggesting proper model convergence without overfitting.




```
Epoch 1/500
3/3 [==============================] - 5s 11ms/step - loss: 1.6188 - accuracy: 0.2609
Epoch 2/500
3/3 [==============================] - 0s 12ms/step - loss: 1.4995 - accuracy: 0.2717
Epoch 3/500
3/3 [==============================] - 0s 7ms/step - loss: 1.4701 - accuracy: 0.2065
Epoch 4/500
3/3 [==============================] - 0s 8ms/step - loss: 1.4043 - accuracy: 0.2717
Epoch 5/500
3/3 [==============================] - 0s 7ms/step - loss: 1.4399 - accuracy: 0.2391
Epoch 6/500
3/3 [==============================] - 0s 5ms/step - loss: 1.3689 - accuracy: 0.2609
Epoch 7/500
3/3 [==============================] - 0s 9ms/step - loss: 1.3792 - accuracy: 0.2935
Epoch 8/500
3/3 [==============================] - 0s 7ms/step - loss: 1.4184 - accuracy: 0.1848
Epoch 9/500
3/3 [==============================] - 0s 8ms/step - loss: 1.3580 - accuracy: 0.2935
Epoch 10/500
3/3 [==============================] - 0s 10ms/step - loss: 1.3706 - accuracy: 0.2391
Epoch 11/500
3/3 [==============================] - 0s 5ms/step - loss: 1.3446 - accuracy: 0.2500
Epoch 12/500
3/3 [==============================] - 0s 7ms/step - loss: 1.3342 - accuracy: 0.2717
Epoch 13/500
...
Epoch 499/500
3/3 [==============================] - 0s 5ms/step - loss: 0.9199 - accuracy: 0.6739
Epoch 500/500
3/3 [==============================] - 0s 4ms/step - loss: 0.7994 - accuracy: 0.7174
```

Section 5

5.1 Summary of Findings

This study explored the application of an **Artificial Neural Network (ANN)** for fault classification based on electrical load and current data. The model successfully identified faults associated with different houses, achieving a final accuracy of **~67%** after 500 training epochs. Key findings include:

- **High classification accuracy for House 1 and House 2 faults**, indicating the model's effectiveness in recognizing these patterns.
- **Confusion between House 3 faults and No Fault conditions**, suggesting overlapping feature characteristics that require further investigation.
- **Steady improvement in accuracy and reduction in loss**, confirming the model's learning capability and stability over training iterations.

5.2 Potential Improvements

While the model demonstrates promising results, several areas can be improved to enhance its performance:

1. **Feature Engineering**: Exploring additional features, such as voltage fluctuations or frequency variations, may improve classification accuracy.
2. **Data Augmentation**: Increasing the dataset size with more diverse fault scenarios can help the model generalize better.
3. **Hyperparameter Tuning**: Optimizing the number of layers, neurons, activation functions, and learning rate could improve convergence speed and accuracy.
4. **Advanced Architectures**: Implementing deep learning models like CNNs or LSTMs may capture more complex temporal patterns in the data.
5. **Ensemble Learning**: Combining multiple models (e.g., ANN + Decision Trees) could enhance fault classification reliability.

5.3 Future Research Directions

Future research can extend this work by:

- **Integrating Real-Time Data Streams**: Deploying the model in a live monitoring system to detect faults dynamically.
- **Explainable AI (XAI) Approaches**: Using techniques like SHAP or LIME to interpret the model's decisions and enhance transparency.
- **Comparative Study**: Evaluating different machine learning models (e.g., SVMs, Random Forests) to determine the most effective approach.
- **IoT-Based Implementation**: Embedding the ANN model in IoT devices for smart grid applications.