Smart Grid Renewables and AI Integration

Jonathan Dofka
Lane Department of Computer Science and Electrical Engineering
West Viginia University
Morgantown, USA
jtd00007@mix.wvu.edu

Andrew Gautier
Lane Department of Computer Science and Electrical Engineering
West Viginia University
Morgantown, USA
aeg00011@mix.wvu.edu

Peter Sorrentino
Lane Department of Computer Science and Electrical Engineering
West Viginia University
Morgantown, USA
pcs00008@mix.wvu.edu

***Abstract*— This project presents the integration and simulation of a microgrid system consisting of wind and solar power stations, a smart grid, and an energy management station. The system is analyzed under various fault conditions, including single-phase-to-ground, line-to-line, double-line-to-ground, and three-phase-to-ground faults, with varying fault-to-ground impedance. Data collected from these fault scenarios is classified using a Convolutional Neural Network (CNN) developed in PyTorch. The trained model aims to enhance fault detection and system stability in microgrids. Results demonstrate the potential of AI-driven solutions in improving the reliability and efficiency of renewable energy-based microgrid systems**

Keywords—Smart Grid, AI, Fault Detection, LabSoft, Lucas-Nuelle, Deep Learning, CNN

I. Introduction

Overview of the Microgrid System

The microgrid system is a small-scale simulation of an active power grid created by Lucas-Nuelle, including multiple types of power generation and simulated distribution. It contains five total modules: wind power plant, smart grid, energy management, professional photovoltaics, and battery storage. Each module is meant to represent a real component of a modern energy grid. The wind, solar, and smart grid modules each can generate power, with the smart grid acting as a balance point. The battery, energy management, and smart grid are all capable of using this power, and the smart grid routes all of it through the distribution network. A majority of these components are able to be managed through a software program called SCADA. These programs are able to capture live data from the grid and control certain aspects of each module. **Insert importance of microgrids in energy management systems.** Microgrids provide a more modular approach to energy production and distribution, reducing strain on the larger grid. With the ability to operate cooperatively or independently with the larger network, they can help

Project Objective

The primary objective of the project was to use the provided lab equipment from Lucas-Nuelle to simulate a power grid being fed by both renewable and classic energy generation sources. This simulated grid was then used to generate faults using the provided fault module. The data collected from these faults could then be used to develop an AI able to indicate when one of these faults occurred.

Team Roles

The Electrical Engineering students used their knowledge and skills to first learn how to operate and maximize the usefulness of the Lucas-Nuelle stations. This was primarily capturing data from the stations and analyzing the results. This also included finding the limitations of the stations and working with the Computer Science student to determine which aspects of the data would be most useful.

The Computer Science student used experience with deep learning architecture and python programming to utilize the data provided by the EE team. Data preprocessing was the responsibility of this student as well as choosing the approach to implementing a neural network which would extract optimal feature information from the dataset available, and in turn tuning the parameters of this network to produce satisfactory results.

II. Microgrid System Setup

A. Component Descriptions

The microgrid system was operated using three major components: wind power plant, smart grid, and professional photovoltaics. The wind power plant consisted of a three-phase, multifunction machine, servo machine test system, induction generator control unit, and dynamic fault simulator. This station was supplied power from the smart grid module and into the control unit and servo machine. This was used to synchronize the simulated power production from the machine to the balance point of the grid supplied by the smart grid station. The included fault generator was used for all the fault collection data and could simulate three-phase, two-phase, two-phase-earth, and one-phase faults.

The smart grid station was used primarily as an energy distribution system. It consists of a main power supply, a consumer, and several double bus-bars for routing power throughout the grid. These bus-bars were controlled using the SCADA software. There was also a module to simulate the losses incurred from long-distance power transmission.

The professional photovoltaic module included several transformers, a power supply for simulating a solar panel, and a smart inverter to convert the DC to AC and synchronize with the balance point. The transformers were used to create an active power regulation unit that held the voltage at the inverter at 400VL-L.

Importantly, all stations included at least one Siemens Sentron PAC4200 Power Quality Meter, or smart meter. These powerful meters were able to capture all aspects of the power going through them and save the data around 30 times a second. This was the primary method of data capture.

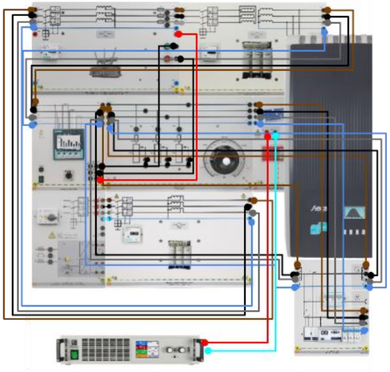

Figure 1: Solar, Voltage Regulator, and Battery

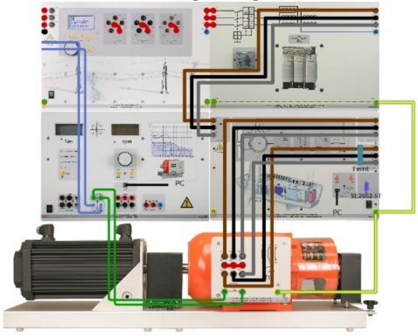

Figure 2: Wind Power Plant

Integration of Components

Each station comes equipped with its own power supply coming from an isolated wall outlet. These supplies are used to run experiments on each station separate from the grid. When operating the microgrid as one station, the power supply module is replaced with an output from a double busbar from the smart grid station. This way, the power was connected to each station through the SCADA software using MainGrid.pvc, a file with the ability to control and monitor all the smart meters and connection switches.

An iterative approach was used to slowly incorporate more components into the microgrid. This began with the wind plant and smart grid only, then the solar plant, battery, autotransformer, and the transmission lines. Once a stable condition could be reached with the last module added, the next module was added and stabilized. With each new module, this became more challenging. These challenges were caused by inherent boundary conditions of the system and growing instability of the grid with more modules. Through a few blown fuses and over-voltage protection faults (OVPs), it was determined that the maximum amount of active power in the microgrid was 1.5kW, maximum voltage of 500VL-L, and maximum current of 2A. If the combined output of the wind and solar station was too high, the excess generated power would flow back into the wall, tripping the breaker of the main power supply. To compensate, the wind generator was limited to 10m/s wind speeds and the solar to 25% irradiance. This resulted in about 400W being produced by the wind station, 300W from the solar station, and the rest from the wall supply, keeping the total power low enough to avoid faults. This was further helped by the addition of consumers draining power, one 560Ohm and one 750Ohm three-phase wye connected resistor. Having these in the grid would help limit the amount of current in each phase, allowing more production from each source.

The instability from each new module was observed to be heavily tied to the amount of internal inductive and/or capacitive loads of the module. This was most easily observed with the battery. It introduced a large capacitive load into the system as well as high-frequency noise produced by the DC-AC inverter. This caused fluctuations in the grid voltage and current, which would lead to OVPs in the solar inverter, solar supply, and cause a desynchronization of the wind plant's DFIG. If the grid voltage fell below or landed above ±.05 Vp.u. for an extended period of time, these problems would occur. To compensate, during initial grid startup, the main voltage as monitored on the smart meter manually and adjusted as needed to maintain 208VL-L.

III. Fault Analysis

A. *Types of Faults Investigated*

There were four types of faults investigated through this project: Three-phase, two-phase, two-phase-Earth, and one-phase. Three-phase faults are the worst-case scenario with maximum voltage loss in the system. This could either be all three-lines have shorted together and touching Earth, or just shorted to each other. Unique to this kind of fault, this will end up creating a balanced load since all three are shorted simultaneously. Two-phase and two-phase-Earth both involve two lines, but the Earth fault also includes a ground short. One-phase faults are a single line-to-ground short and typically results in the smallest loss of voltage. The fault impedance controls what percent of the power in the line is allowed to flow into the fault and was one of the parameters able to be varied (100%, 60%, 40%, 20%, and 0% for each line where the percentage is the amount of power allowed to flow through the fault). Lower fault impedance allowed more current to be dumped into the fault which would create more dramatic reactions from the grid. Every fault was executed for 500ms to capture the largest amount of data possible while a fault was occurring.

Data Collection

Data was gathered from the computer running the MainGrid.pvc software. Each smart meter can be monitored from this one station. For every fault configuration (percent loss and fault type), capture was started, allowed five seconds of normal operation (mains on, solar at 25% irradiance, and wind at 10m/s), then the fault was triggered and allowed one minute to return to normal operation. The smart meters would capture the reactions from the voltage, current, active power, reactive power, total harmonic distortion, phase imbalance, and frequency. For consistency, this data was collected from the same place in the grid (main power supply/balance point), but by using these smart meters, the response could be captured from any location in future experiments. The smart meters would capture a data point every 30ms and then was exported to an Excel spreadsheet with a label of the kind of fault. The faults were labeled based on the kind of fault that occurred (three-phase, two-phase, etc.). Two different versions of the fault data were created: One used a less sensitive, but more realistic rule of a fault being defined as a voltage $.95<V_{L\text{-}N,\ p.u.}<1.05$, and one more sensitive using a rule of $.98<V_{L\text{-}N,\ p.u.}<1.02$.

Results of Fault Analysis

For the main grid supply, while a fault occurs, the voltage seen in the balance point drops and the current increases proportionally to the severity of the fault. For example, the three-phase 100-100-100 fault reduced the voltage by 30VL-L and increased the current by 2.5A, while a one-phase 0-0-100 fault lowered the voltage by 7.3VL-L and raised current by .3A. If the fault was severe enough, there would also be a recovery period after the fault ranging from 500ms to non-existent where the voltage would return to a higher voltage and current than before the fault. The overvoltage level was 6.3VL-L and .6A for the largest fault. These effects were largely mirrored in the other two stations. The exact same per-unit drop in voltage was seen across the wind and solar, but the current increase was less in magnitude.

The solar plant only had an increase of .05A during the most extreme fault, demonstrating a system more resilient to changes in current than the rest. This is due to the voltage step-up and auto transformers employed to raise the voltage to and maintain it at 400VL-L and its inability to create any more power than what is provided by the sun while operating at the maximum power point. The active power regulator using the autotransformer would initially work to bring the voltage back to nominal levels, but the fault never lasted long enough for this to have an observable effect.

When given a severe enough fault, the wind station would automatically increase the pitch angle of its blades. This was done to decrease the amount of power produced. The wind stations smart meter would read a large increase in current in the 3P-100-100-100 as an example jumping from 1.1A to 3.4A for each of the 3 phases at the start of the fault, along with a large increase in reactive power.

When labeling the faults for AI training, it became clear that not all of the faults produced strong enough changes in the V_{L-N} to always be labeled as an event. This was much more of an issue for faults occurring in fewer lines, especially the one-phase series. The more sensitive data labels were implemented to counter this, but was not 100% effective for the high fault impedance faults.

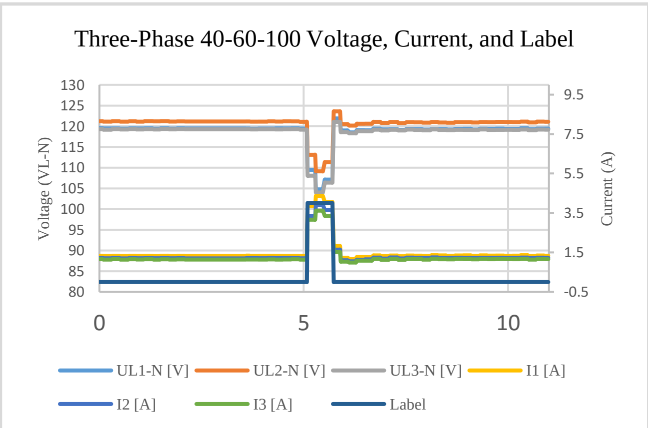

Figure x: Example of Fault Collection and Labeling

IV. AI Model Development for Fault Classification

A. *Objective of AI in the Project*

The objective of AI in this project was to evaluate the effectiveness of building deep learning models from data collected from a microgrid and using these signals to classify faults in distribution systems.

Tools Used

This project was carried out using Python for data preprocessing, visualization, and statistical analysis. The following libraries were used for the aforementioned tasks: Pandas, NumPy, Scikit-learn, Matplotlib. The PyTorch library was utilized for training and implementing the neural network.

Preprocessing

Because data obtained through the smart grid was very clean, there was minimal preprocessing required to correct for missing data and anomalies that would typically make up a large portion of preprocessing tasks. The only work that was required was implementing normalization by utilizing sci-kit learns minmax scaler. Additionally, labeled were appended to csv files by the EE team based on the threshold of voltage abnormality that constituted a fault. Some additional time was required to stack csv files to make a suitable training set. The final step was creating copies of faults by applying 1% gaussian noise to classes of faults that were underrepresented.

D. Model Architecture

As previously mentioned, the architecture chosen for this task was CNN. CNN's have shown robust ability to extract relevant features from electrical signal data that made it an obvious choice for this architecture. A summary of the layers of the neural networks can be seen in Figure 3.

$$X' = \mathrm{permute}(X, (0,2,1)) \quad \Rightarrow \quad X' \in \mathbb{R}^{B\times C\times T}$$

$$H_1 = \mathrm{Conv1D}(X'; W_1, b_1), \quad W_1 \in \mathbb{R}^{16\times C\times k},\ b_1 \in \mathbb{R}^{16}$$

$$A_1 = \mathrm{ReLU}(H_1)$$

$$P_1 = \mathrm{AdaptiveMaxPool1d}(A_1, \mathrm{output\_size} = T/2)$$

$$H_2 = \mathrm{Conv1D}(P_1; W_2, b_2), \quad W_2 \in \mathbb{R}^{32\times 16\times k},\ b_2 \in \mathbb{R}^{32}$$

$$A_2 = \mathrm{ReLU}(H_2)$$

$$P_2 = \mathrm{AdaptiveMaxPool1d}(A_2, \mathrm{output\_size} = T/2)$$

$$F = \mathrm{flatten}(P_2) \quad \Rightarrow \quad F \in \mathbb{R}^{B\times(32\cdot T/2)}$$

$$Z_1 = W_3 F + b_3, \quad W_3 \in \mathbb{R}^{64\times(32\cdot T/2)},\ b_3 \in \mathbb{R}^{64}$$

$$A_3 = \mathrm{ReLU}(Z_1)$$

$$Z_2 = W_4 A_3 + b_4, \quad W_4 \in \mathbb{R}^{N\times 64},\ b_4 \in \mathbb{R}^{N}$$

$$Y = \mathrm{softmax}(Z_2) \quad \Rightarrow \quad Y \in \mathbb{R}^{B\times N}$$

Figure 3: Layers of CNN Classifier

Experiments were conducted with additional layers of convolution, but validation loss worsened during experiments so that architecture was abandoned. One of the parameters that was experimented with the most was window size. After conducting a parameter grid search, a window size of 50 rows was decided upon, which represented roughly one second of data from our power quality meters. So, our input to our model would then be a tensor of 34 features, 50 rows, and a batch size of 32 which was also chosen over 64 in parameter tuning although batch size optimization could be further explored for these large datasets. These are then converted into tensors for training as PyTorch requires. Training loss curve can be seen in figure 4. The optimizer used in training was Adam.

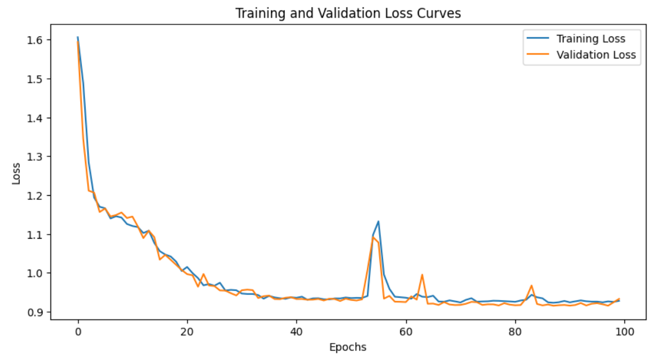

Figure 4: Training and Validation Loss

E. Results of AI Model

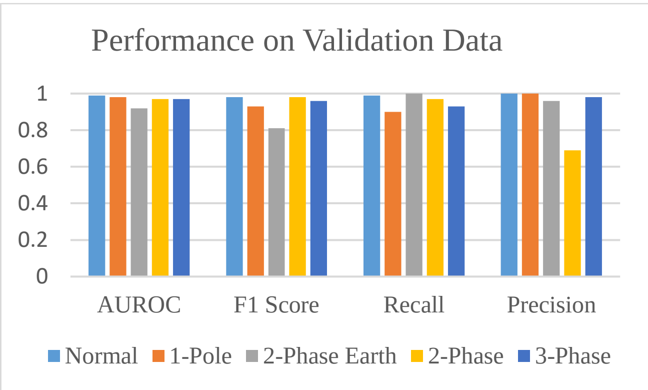

Figure 5: Classifier Performance

As seen in figure 5, the trained classifier performed exceptionally well when introduced to unseen data in the validation phase, with an overall accuracy rating of 99%. F1 score, recall, and precision were evaluated as well.

V. Discussion and Analysis

A. Microgrid Fault Behavior

Summary of how the microgrid system behaves under each fault type.
Insights gained from simulation results.

AI Performance and Results

The effectiveness of the AI model in classifying faults.
Challenges in training and areas for improvement in model performance.

Integration of AI and Fault Data

How the AI model can be used in real-world microgrid systems for fault detection and maintenance.

VI. Future Work

A. Improvement In AI Model

Several modifications can be explored to enhance the performance and robustness of the proposed CNN architecture. First, batch normalization layers can be introduced after each convolutional layer to stabilize training by normalizing feature maps and reducing internal covariate shift. To mitigate overfitting, dropout regularization with rates between 20-50% may be applied, particularly in the fully connected layers. Replacing max pooling with strided convolutions could improve downsampling efficiency while preserving spatial hierarchies. Additionally, the model's feature extraction capabilities may benefit from a deeper architecture or the inclusion of residual connections to facilitate gradient flow. Data normalization should be prioritized as a preprocessing step to ensure stable and efficient training dynamics. Finally, adaptive learning rate scheduling, such as ReduceLROnPlateau (provided by PyTorch), can be implemented to dynamically adjust the learning rate during training, potentially improving convergence and final model performance.

System Expansion

Future simulations with more complex fault types.
Scaling the microgrid system with more renewable energy sources and complex fault scenarios.

Real-World Application

How the results can be applied to real-world microgrids for better energy management and fault detection.

VII. Conclusion

Summary of findings.
The value of integrating AI with microgrid fault analysis.
Future research and the potential impact of this project on energy management systems.

Acknowledgements