# Event Classification of PMU Time-Series Data Using Recurrent Neural Networks

Kai Long 4778776

December 12, 2025

## 1 Introduction and Objective

Real-time anomaly detection is critical for maintaining the stability of modern power grids. Synchrophasor technology, specifically Phasor Measurement Units (PMUs), provides high-resolution data that captures the dynamic behavior of power systems. The primary problem addressed in this project is the automated classification of power system events based on streaming measurements.

The study use the "Bus39 competition Data" dataset derived from the IEEE 39-bus System. The dataset consists of 14 key features, including 3-phase voltage and current phasors, system frequency and the Rate of Change of Frequency (ROCOF).

The objective of this project is to design a Recurrent Neural Network (RNN) model to classify system states into six categories: normal operation, fault, line outage, generation change/outage, load change/drop and bad data. The goal is to achieve high classification accuracy on test data to demonstrate the viability of RNNs for real-time power system monitoring.

## 2 Data Preprocessing

The dataset was loaded directly from the provided `Bus39_Competition Data.xlsx` file using the Pandas library. An initial inspection for missing values was conducted across all 14 feature columns. The check confirmed that the dataset was complete with no missing entries. therefore, no imputation or sample removal was required.

The `TIMESTAMP` column was dropped as it does not contain physical system state information. The `EVENT` columns was dropped from feature matrix X and preprocessed as the target label. Since the labels don't have continuous original class labels, a Label Encoder was used to preprocessed the target label y.

According to the requirement, the dataset was split into a training set (80%) and a testing set (20%). A stratified split was employed to maintain the same class distribution, and the data was shuffled to mitigate any time dependent bias.

To facilitate model convergence, feature standardization was applied using `StandardScaler`. To prevent data leakage, the scaler was fit only on the training set, and transform the testing set.

## 3 Model Architecture

In compliance with the assignment, the proposal model is built upon a Recurrent Neural Network (RNN). The model used in this project is a double-layer RNN. The input to model is a sliding sequence of 10 measurements, each consisting of 14 features. The tensor shape is (batch size, sequence, 14). the hidden size of this model is 32 neurons. This layer extracts non-linear relationships between different states. This size prevents underfitting while minimizing the risk of overfitting compared to larger size given the dataset size. The standard Tanh activation function is used within the recurrent unit. The output size is set to be 6 due to the amount of classes defined in the dataset. To prevent overfitting and improve model generalization, the dropout layer of a probability of 0.4 is applied to the output of the RNN layer. Due to multi-class classification, cross-entropy function is applied to evaluate the loss of the model. Meanwhile, Adam is used as its optimizer to update the weight of the model.

# 4 Training Setup

The model was implemented using the PyTorch deep learning framework. All experiments were conducted on MPS environment to accelerate computation. The learning rate was set to 0.001, which was found to be effective for the Adam optimizer in minimizing the loss without causing oscillation. L2 normalization is set to 0.0001, preventing the overfitting. To achieve a balance between memory efficiency and gradient accuracy, the batch size was set to 64 samples. The model was trained for 20 epochs by observing the rapid converging process and stabilized epochs.

# 5 Results

The training and validation loss curves over 20 epochs are presented in Figure 1. The model demon-

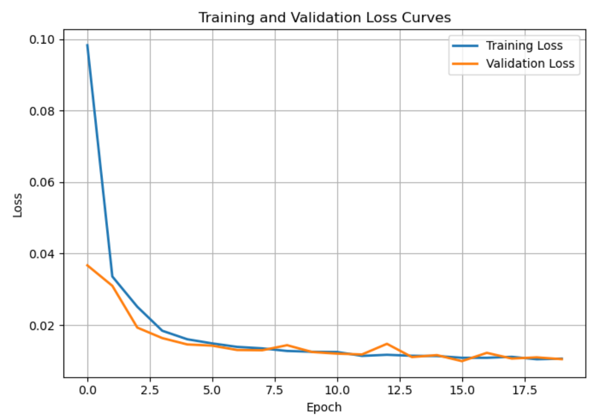


Figure 1: Training and Validation Loss over Epochs

strated rapid convergence, with both training and validation losses dropping significantly within the first 5 epochs. The validation loss tracked the training loss throughout the process and stabilize around 0.01. This behavior confirms that the implementation of the 2-layer architecture with a dropout rate of 0.4 successfully mitigated overfitting. The final model achieve an overall accuracy of 99.78% on the test set. As shown in the confusion matrix, the predictions are strongly concentrated on the vast majority of samples. However, a detailed inspection of the classification metrics reveal performance

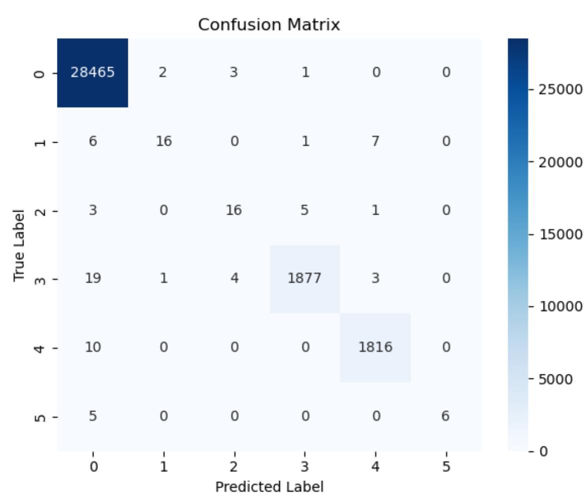


Figure 2: Confusion Matrix

disparities due to class imbalance. The results indicate that the RNN is highly effective at identifying dominant system states. Classes normal operation, Generation change/outage and load change/drop

| Event | Precision | Recall | F1-score | support |
|---|---|---|---|---|
| Normal operation | 0.9985 | 0.9998 | 0.9991 | 28471 |
| Fault | 0.8421 | 0.5333 | 0.6531 | 30 |
| Line outage | 0.6957 | 0.6400 | 0.6667 | 25 |
| Generation change/outage | 0.9963 | 0.9858 | 0.9910 | 1904 |
| Load change/drop | 0.9940 | 0.9945 | 0.9943 | 1826 |
| Bad data | 1.0000 | 0.5455 | 0.7059 | 11 |
| accuracy | | | 0.9978 | 32268 |
| macro avg | 0.9211 | 0.7832 | 0.8350 | 32267 |
| weighted avg | 0.9977 | 0.9978 | 0.9977 | 32267 |

Table 1: Classification Report

achieved the F1-score exceeding 0.99. Conversely, rare events such as Fault, Line outage,and Bad data, show lower recall score. This is attributed to the class imbalance. The model maintain relatively high precision for rare events. The loss function 'cross entropy' is biased toward the majority class, leading to missed detections for the minority class. The macro averaged performance remain robust for the real system.

## 6 Discussion and Conclusion

The primary objective of this project was to design a deep learning model for real-time anomaly detection using PMU data. The experimental results demonstrate that the proposed 2-layer Vanilla RNN architecture is effective. The overall testing accuracy is 99.78%.

The model shows great performance in identifying dominant system state, all of which achieve F1-score above 0.99. However, rare event categories have lower recall rate. This performance disparity is caused by the imbalance dataset. To deal with this situation, The loss function 'cross-entropy' can be added weight using the reversely percentage of each events' amount. As a result, the recall rates of rare events improved significantly, exceeding 0.90, However, this improvement comes at the cost of reduced precision (0.07) for minority classes, showing a trade-off between recall and precision.

Despite the limitation, the results shows that RNN model is capable of capturing patterns in PMU measurements and suitable for power system event classification tasks. Future work may focus on finding the best weight to get higher score for both minority events and majority events. More advanced model such as LSTM network, could be explored to further enhance model robustness.