Journal of
Engineering Sciences
Faculty of Engineering
Assiut University
JES
مجلة
العلوم الهندسية
كلية الهندسة
جامعة أسيوط
journal homepage: http://jesaun.journals.ekb.eg

A Real-Time Approach for Error Detection in μPMU Measurements

Received 21 August 2024; Revised 23 November 2024; Accepted23 November 2024

Rahma Hasan[1]
Eslam Alqabbani[2]
Mansour A. Mohamed[3]
M. A. Nayel[4]

Keywords
μPMU's errors, continuous wavelet transforms, convolutional neural network, feature extraction, error detection.

Abstract: The quality of phasor data from micro-Phasor Measurement Units (µPMUs) is critical for smart grid applications. It plays a key role in various aspects of power system management and is essential for the transition to a smarter and more sustainable grid. Recent studies imply that despite having a high level of monitoring features and accurate algorithms, µPMUs are vulnerable to errors in the measurements. Traditional methods for error detection in µPMUs typically rely on direct analysis of voltage signals. While effective to some extent, these methods can struggle with the complex and dynamic nature of power system measurements, especially under varying load conditions and in the presence of noise. To address these challenges, this paper presents a novel approach for error detection in µPMU voltage measurements using a combination of continuous wavelet transform (CWT) and a convolutional neural network (CNN). The proposed detection approach is applied on Assiut university distribution grid sub-feeder. A set of evaluation metrics such as accuracy, recall, precision, and F1 score were used to compare the error detection performance of the proposed CNN model with conventional machine learning (ML) algorithms. The results show that the proposed CNN model outperforms the conventional ML algorithms for detecting errors in µPMU voltage measurements under different load conditions.

1. Introduction

The modern structure of distribution grids has become more dynamic due to the rapid integration of distributed energy resources (DER) and electric vehicles [1]. These developments have introduced new challenges and complexities, transforming the traditionally uni-directional power flow into a bi-directional one. Consequently, the traditional unidirectional power flow control and protection systems in the distribution grid need to be enhanced with smart features, real-time monitoring, and two-way communication capabilities to ensure grid stability. The µPMU is the most advanced measurement device

[1] Dept. of Electrical Engineering, Assiut University, Assiut, Egypt. asia111@yahoo.com
[2] Dept. of Electrical Engineering, Assiut University, Assiut, Egypt. Islam@aun.edu.eg
[3] Dept. of Electrical Engineering, Assiut University, Assiut, Egypt. mamohamed2004@yahoo.com y
[4] Dept. of Electrical Engineering, Assiut University, Assiut, Egypt. mohamed.nayel@aun.edu.eg -

https://doi.org/10.21608/jesaun.2024.314227.1361

currently available for medium and low-voltage distribution grids. It can measure various parameters, including three-phase voltage and current phasors, active and reactive power, frequency, and the rate of change of frequency (ROCOF), with a reporting rate of up to 120 samples per second in a 60 Hz system. This high sampling rate provides an angular accuracy of ±0.01° and an amplitude accuracy of ±0.05%, resulting in a total vector error (TVE) of ±0.01% [2]. Additionally, the μPMU features precise timestamps synchronized with phase angles across multiple locations, enhancing their reliability and effectiveness in monitoring the power system. μPMUs have a wide range of applications in modern distribution grids. According to a recent survey [3], these applications include model validation, distribution system state estimation, topology detection, phase identification, and transient analysis. Also, most studies have focused on detecting the event and anomaly in distribution grids using μPMU data [4]–[6].

Despite the high accuracy of μPMUs, their measurements are not entirely free from errors, which can arise from various sources. Primarily, these errors can be classified into synchronization errors and instrumental errors. Synchronization errors occur due to inaccuracies in the timing synchronization between the μPMU and the reference time source, usually provided by Global Positioning System (GPS). Instrumental errors arise from the μPMU hardware itself [7], including components such as current transformers (CTs), voltage transformers (VTs), analog filter, Analog to Digital Converter (ADC) sampling, digitization process, and other components involved in the measurement process. These errors can be either systematic or random. Systematic errors remain constant across successive measurements taken over a short period of time, while random errors vary with each observation [8]. After applying the proposed approach for error estimation and compensation in μPMU in [7], there can still be residual errors resulting from the inaccuracies of CTs and VTs. CT and VT errors are external errors introduced to the primary measurement unit of the μPMU. In traditional PMUs, these errors are typically minimal and negligible compared to the accuracy and resolution required for their applications. However, in distribution systems, CT and VT errors are more significant due to higher noise levels [9]. These transducers are a significant source of error in the μPMU's synchrophasor output, and they can drown steady-state phasor differences between measurements taken at close locations on a distribution circuit. This causes difficulties for a number of μPMU applications at distribution level [9].

Error detection in PMU measurements has evolved significantly over time. Traditional methods primarily relied on statistical analysis and redundancy checks, where data from multiple PMUs were compared to identify discrepancies. Techniques such as state estimation, which use mathematical models to predict expected measurements and compare them with actual PMU data, were also commonly employed. In [10], it was presented a method for detecting and correcting errors in PMU measurements using the concept of calibration factors. This method's effectiveness is heavily dependent on the accuracy of the initial model parameters, which may not generalize well across different grid configurations or under varying operating conditions. In [11], a phasor-data-based state estimator was proposed to correct bias errors in phase angle measurements. However, this method neglects the possibility that bias errors could also be present in the magnitude component of synchrophasor measurements. In [12], a method was introduced for detecting systematic

errors in PMU measurements based on the power system state estimation, but it has several limitations. The effectiveness of this method is closely tied to the accuracy of the underlying state estimation model, which may be compromised by model assumptions and simplifications that do not fully capture the complexities of real-world power systems. Additionally, this method primarily addresses systematic errors and may not adequately cover random or intermittent errors that could also impact the reliability of PMU data. The reliance on state estimation techniques, which require extensive computation and accurate system models, might limit the method's practicality for real-time applications in large-scale or dynamically changing power grids. In [13], an innovative approach was presented for online bias error detection and calibration of PMU measurements using density-based spatial clustering. The effectiveness of the density-based spatial clustering method is sensitive to the selection of clustering parameters, which may require fine-tuning for different grid conditions and might not be universally applicable. Moreover, the reliance on clustering algorithms may lead to challenges in distinguishing between genuine measurement errors and anomalies caused by transient events or grid disturbances, potentially impacting the reliability of the calibration process. In [14], principal component analysis (PCA) was used for detecting bad data in PMU measurements by identifying anomalies that deviate from normal system behavior. The effectiveness of PCA depends heavily on the assumption that normal system behavior is well-captured in the training data, which may not always be the case, particularly in dynamically changing power systems.

Recently, signal processing methods, including Fourier and wavelet transforms, were introduced to detect anomalies by analyzing the frequency components of the measurements [15]. The wavelet technique is an efficient signal processing technique for PMU data that exhibit nonstationary characteristics [16]. In [17], an approach based on wavelet analysis was proposed to identify anomalies in PMU measurements, which may indicate potential cyber threats to the system. More recently, machine learning (ML) and artificial intelligence techniques have been developed to enhance error detection in PMU measurements. These modern methods, such as support vector machines (SVM), artificial neural networks (ANN), and convolutional neural networks (CNN), leverage large datasets to learn patterns associated with both normal and erroneous measurements, allowing for more accurate and real-time error detection. In [18], an approach was introduced for identifying outliers in PMU voltage measurements using ANN. This approach leverages PCA to extract relevant features from the PMU data, which are then used to train an ANN classifier. In [19], Pearson correlation was utilized in conjunction with a neural network (NN) classifier to detect errors in μPMU voltage measurements. This method provides insights into linear dependencies between data sets, aiding in the identification of discrepancies. In [20], a approach based on support vector regression (SVR) was presented, which can detect and correct the false data in PMU voltage measurements.

In this study, random errors in μPMU voltage measurements were detected using a hybrid approach that integrates signal processing and deep learning (DL) techniques. Initially, the distribution grid sub-feeder model is represented using MATLAB/Simulink to generate μPMU sample data and simulate error injection scenarios due to the VT of μPMU. CWT is then applied to the μPMU data to extract time-frequency coefficients, capturing the intricate

patterns associated with both healthy and erroneous measurements under different load conditions. These coefficients are fed into a CNN network which focused on extracting spatial features, allowing the model to classify and differentiate between healthy and error states. The model is trained on a dataset that included different error scenarios, allowing it to learn the distinct features associated with each error type. The main contributions of this paper can be summarized as follows:

1- In contrast to existing μPMU error detection methods, the proposed approach in this study uses CWT coefficients instead of raw voltage signals, which capture the essential features of the signals more effectively.
2- The combined use of CWT and CNN enables effective pattern recognition, allowing the model to distinguish normal operation from the presence of errors in μPMU measurements. Also, the combined approach aims to improve the generalization capabilities of the model, enabling it to perform well on unseen data. This is crucial for robust error detection in different scenarios and under different load conditions.
3- The performance of the proposed CNN model was compared with other ML models, such as SVM and NN, demonstrating superior performance. These methods only require μPMU measurements without needing information about the network model.
4- A time analysis was conducted to address real-time application concerns. This includes measuring the time required for data generation for testing, as well as the time taken by the model to detect errors. These analyses provide insight into the model's performance and potential applicability in real-world smart grid environments.

2. Literature review on the nature of errors in PMU measurements

Previous studies have noted that errors in PMU measurements can manifest as anomalies. Anomalous data encompasses missing data, outliers [14], [18], and event data. Among these, outliers and missing data are classified as bad data, typically resulting from erroneous measurements or poor data quality [21]. Fig. 1 shows the various types of PMU anomaly data.

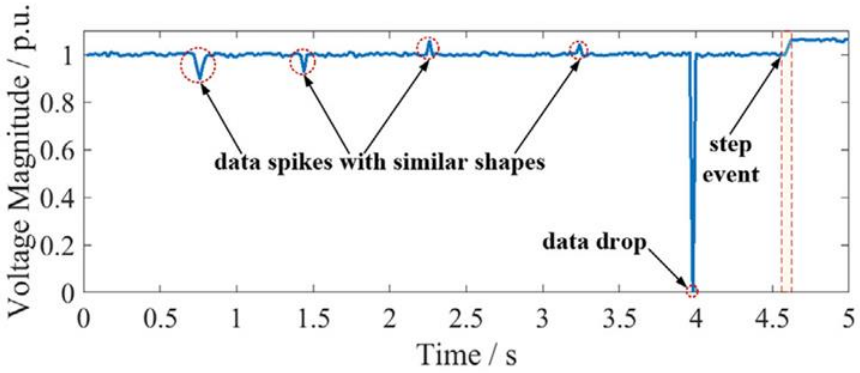


Fig. 1: Different types of PMU anomaly data [21].

Outliers in measured data can result from network disturbances influenced by the system's dynamic properties. In PMU data, outliers are data points that deviate significantly from expected measurements, often appearing as spikes when no system events occur. Event data, on the other hand, arises from system events such as switching operations or sudden load changes. Fig. 2 illustrates the variation in the measured voltage magnitude at Bus 1. Bad data, including outliers, can be intentionally or unintentionally introduced. For example, as shown in Fig. 2, bad data was injected into the voltage measurements at Bus 1 at t = 25.4s.

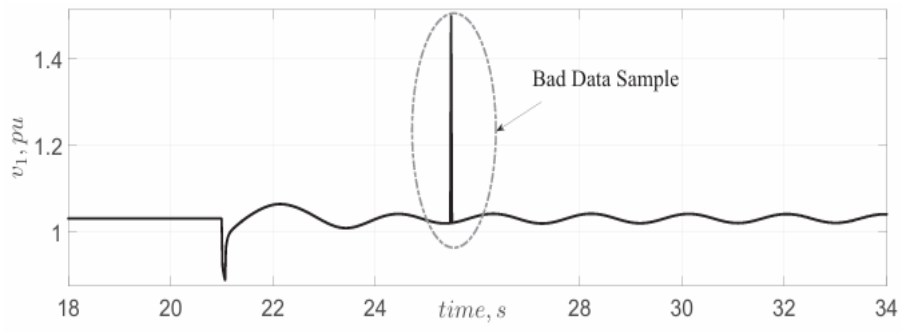


Fig. 2: Voltage magnitude of bus 1 in pu [18].

Bad data can occur randomly, manifesting as either isolated instances or contiguous sequences of erroneous measurements. In [22], a spectral clustering (SC) was proposed for detecting both single and contiguous bad data within PMU measurements. For single bad data points, the algorithm evaluates each measurement against its neighboring values, employing statistical tests to determine anomalies. Contiguous bad data, characterized by a series of erroneous measurements, is identified by analyzing temporal correlations between sequential data points. Fig. 3 illustrates examples of single and contiguous bad data. These types of errors are represented through the three error scenarios developed in this paper.

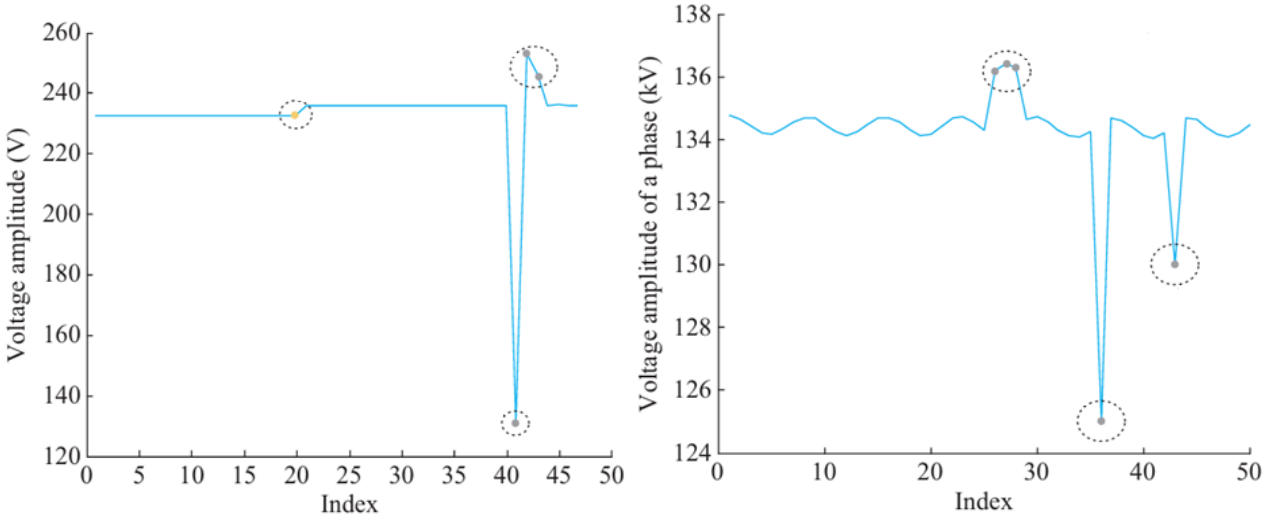


Fig. 3: Detection results of field data with three methods in a distribution network. (a) in case of single bad data, and (b) in case of contiguous bad data [22].

3. Theoretical background

3.1 Continuous wavelet transform

A practical application of the CWT is real-time monitoring of voltage and current signals for the rapid detection and elimination of transient events that may affect the quality of the electric service [23]. CWT allows for the representation of signals in different frequency components, providing a multiresolution analysis of the data. This is beneficial for capturing both high and low-frequency features in the μPMU measurements. CWT allows for time-frequency analysis, providing insights into the variations of voltage and current signals at different scales and time intervals. This is crucial for detecting frequency changes associated with errors. Also, CWT allows the localization of features in both time and frequency domains. This helps in pinpointing the exact time when errors occur and understanding their frequency characteristics. CWT of a function $x(t)$ is expressed as follows [24]:

$$CWT\ (\tau, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\ \psi^* \left(\frac{t-\tau}{a}\right) dt \qquad (1)$$

where $CWT\ (\tau, a)$ is the output, which describes the wavelet coefficients. $x(t)$ shows the one-dimensional signal to be transformed. τ is translation (or shifting) parameter, a is scale (or dilation) parameter of wavelet function. ψ^* is the complex conjugate of mother wavelet. This equation allows CWT to compute the array of coefficients for each frequency scale parameter, resulting in a detailed representation of the signal's frequency and temporal characteristics through the coefficients.

3.2 Convolutional neural networks

CNNs are a subset of DL models widely used for image processing, video analysis, and various other applications involving spatial data [25]. CNNs can be used for both classification (e.g., identifying healthy / error states) and regression (e.g., predicting the error values in measurements) tasks, providing flexibility in the analysis. CNNs are known for their ability to generalize well to new, unseen data.

This adaptability is crucial for handling different error scenarios and variations in μPMU measurements. CNNs are also capable of capturing non-linear relationships, making them suitable for identifying errors in μPMU measurements under different load conditions. Compared with ML algorithms, the most distinctive characteristic of CNN is its ability to automatically extract features via a training process. Hence, CNN can be used to automatically recognize and analyze the time-frequency coefficients of signals obtained by CWT, enhancing the error detection in μPMU voltage measurements. Generally, CNN consists of three different layers: a convolution layer, a pooling layer, and a fully connected layer. CNN architecture leverages convolutional operation, a core mechanism for extracting features from input data. Convolutional layers in CNNs contain learnable filters, or kernels, which perform convolution operations on the input data to generate feature maps. This process involves element-wise multiplication of the filter weights with localized regions of the input data, followed by a summation to produce the output [25]. The convolution operation can be mathematically expressed as follows:

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * K_{ij}^l + b_j^l\right) \quad (2)$$

where $f(.)$is activation function, K_{ij}^l is the convolutional kernel at position (i, j) of the l-th layer. b_j^l represents the bias, x_i^{l-1} the feature mapping of the previous layer, M_j the set of feature mappings and x_j^l represents the feature map of the current layer. These convolutional layers are followed by batch normalization, which normalizes the output of each convolutional layer to enhance network stability and convergence [26]. Activation functions are also applied in this stage to introduce non-linear characteristics, enabling the network to learn complex data relationships. Common activation functions include sigmoid, tanh, and Rectified Linear Unit (ReLU). ReLU is widely adopted as it significantly enhances CNN performance and accelerates training compared to other activation functions [27]. The ReLU function is defined as follows:

$$f(x)_{ReLU} = max(0, x) \quad (3)$$

where x is the output of the previous neuron. After the image features are obtained from the convolutional layer, a pooling layer is added in order to reduce the size of the feature parameters of the images and maintain important features while reducing computational complexity. Common pooling operations include max pooling and average pooling. The pooling layer is defined as the following form,

$$x_j^l = f\left(pooling\left(x_i^{l-1}\right) + b_j^l\right) \quad (4)$$

Here, pooling(.) denotes the pooling function. Finally, the classification occurs in the fully connected (FC) layer, which consolidates the results from the convolutional and pooling processes to predict and classify the images. Dropout layers are then employed to randomly deactivate a fraction of neurons during training, reducing overfitting and improving the network's generalization capabilities [28]. The final layer in the CNN is the softmax layer, which calculates the probability distribution over the classes based on the features extracted by the convolutional layers. The softmax function is defined as follows:

$$\text{softmax}(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_i}} \quad (5)$$

where y_i represents the i-th output in the fully connected layer, and softmax(y_i) is the corresponding probability of the i-th output converted by softmax function.

3. Proposed approach for error detection

The proposed approach for error detection in µPMU voltage measurements leverages a combination of advanced signal processing and deep learning techniques to achieve precise

and reliable results. The process begins with the generation of μPMU datasets using a Simulink model of a distribution grid sub-feeder. Next, CWT is employed to extract meaningful features from the μPMU voltage signals, capturing the essential characteristics needed for effective error detection. The core of the approach lies in the CNN architecture, which is designed to detect errors in μPMU voltage measurements with high accuracy. The CNN is responsible for effectively extracting relevant features from the voltage measurements, capturing the spatial patterns and signal characteristics. The model's performance was evaluated on a separate test dataset that was not used during training, ensuring robustness and generalization of new data. Finally, the effectiveness of the proposed approach is evaluated using a set of metrics that assess the overall performance of the model in detecting errors under different load conditions.

4.1 Sub-feeder modeling

To collect a dataset of μPMU measurements, the sub-feeder NBS_8 of Assiut university grid is used. A three-phase unbalanced distribution grid sub-feeder consists of 5 buses. Bus_1 is the slack bus of the system, while the other four remaining buses are pure load buses. Distribution grid cables are used to connect the buses, modelled by their equivalent π-Model. Two μPMUs are installed at buses 1 and 4, providing full observability of the feeder while minimizing the number of units and current channels required, taking operational costs into account [29]. The interring branch current, load current, and low voltage side measurements are obtained from μPMUs at the two buses. The data from the μPMU are received at 50-Hz sampling frequency. At each sampling time, three-phase voltage and three-phase current of two μPMUs have been obtained with a reporting rate of one measurement/cycle. Therefore, 1 sample has been extracted every 0.02s (50 samples in one second). The feeder is assumed to be unbalanced with random loads change within 10%. The load's transformers are connected in Dy_{11g} configuration and of 11/0.4 KV voltage transformation ratio. The sub-feeder is modelled using MATLAB/Simulink software with the μPMU Simulink model. Fig. 4 shows the sub-feeder model.

4.2 CWT-based feature extractor

In this study, a matrix x is constructed, consisting of six signals. The signals included are three-phase low voltage signals of two μPMUs. Then, the CWT is applied separately on each signal to extract coefficients. At each time, a window of length w = 64 is applied to six signals and the window is shifted by 16 samples in each step. For each signal, the CWT is computed using an analytic Morlet wavelet (amor) with a filter bank, obtaining the wavelet coefficients. A matrix x is represented as follows:

$$x = \left[V_{a1}^{\mu PMU1}, V_{b1}^{\mu PMU1}, V_{c1}^{\mu PMU1}, V_{a4}^{\mu PMU2}, V_{b4}^{\mu PMU2}, V_{c4}^{\mu PMU2}\right] \quad (6)$$

The coefficients obtained by CWT are size of 32x64x6. These coefficients are then input to the trained CNN that determines which class it belongs to. Fig. 5 shows examples of voltage signals under healthy conditions and three different error scenarios, along with their corresponding CWT spectrogram images.

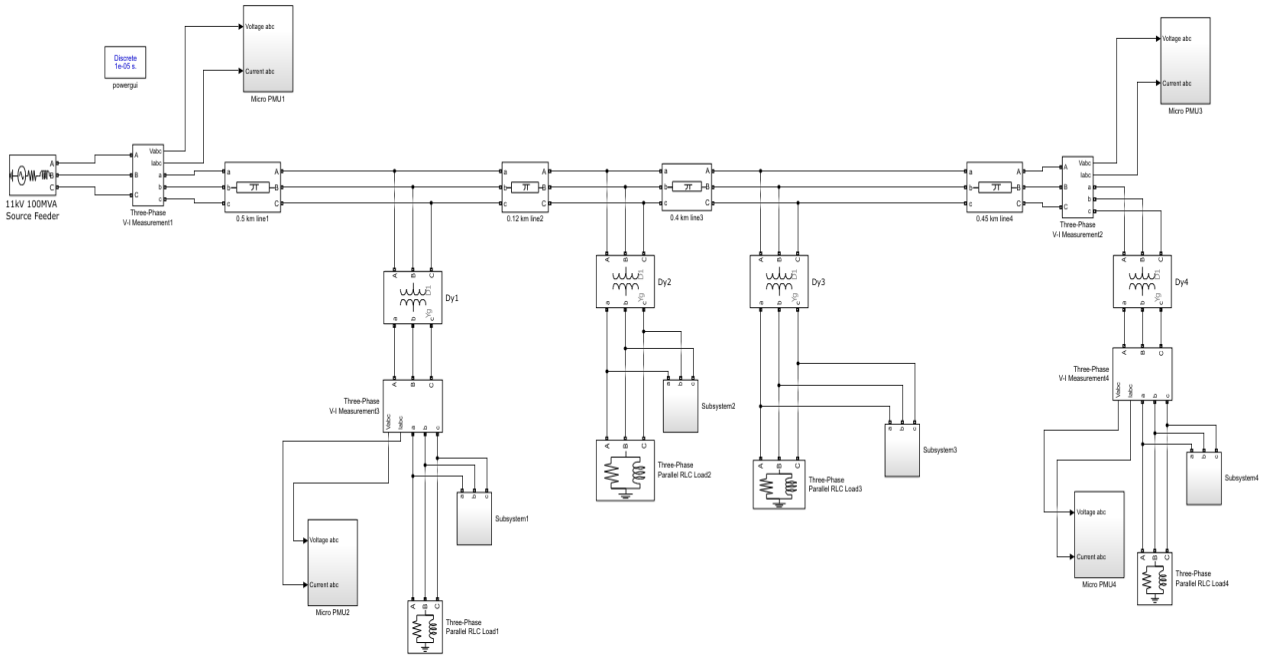


Fig. 4: The sub-feeder model in MATLAB Simulink

4.3 Classification-based CNN approach

Fig. 6 shows the classification-based CNN configuration developed for multiclass error detection tasks. For the classification task, the network learns to distinguish between seven classes: class 0, class 1, class 2, class 3, class 4, class 5, and class 6 corresponding to healthy, error in Va1, error in Vb1, error in Vc1, error in Va4, error in Vb4, and error in Vc4, respectively.

The proposed CNN architecture is designed for a CWT coefficient input of dimensions [32 64 6] without normalization. The network comprises multiple layers to perform classification tasks effectively. Table I shows detailed information about the proposed CNN architecture. As given in Table 1, the architecture includes three convolutional layers. The first convolutional layer (Conv1) applies 16 filters of size 5x5 with padding to preserve the input dimensions, followed by a batch normalization layer to stabilize the learning process and a ReLU activation layer to introduce non-linearity. The second convolutional layer (Conv2) uses 32 filters of size 3x3, with a stride of 2, reducing the spatial dimensions by half. It is also followed by batch normalization and a ReLU activation layer. The third convolutional layer (Conv3), with 64 filters of size 3x3, further increases the feature extraction capability, and similarly, it is followed by batch normalization and ReLU activation. A dropout layer with a 0.5 dropout rate is added to prevent overfitting by randomly deactivating neurons during training. Finally, a fully connected layer (FC) with 7 output neurons maps the extracted features to the 7 classes, followed by a softmax layer for probability distribution, and a classification layer to determine the final class label. This architecture is designed to effectively extract features from the input data while minimizing overfitting and maximizing classification accuracy.

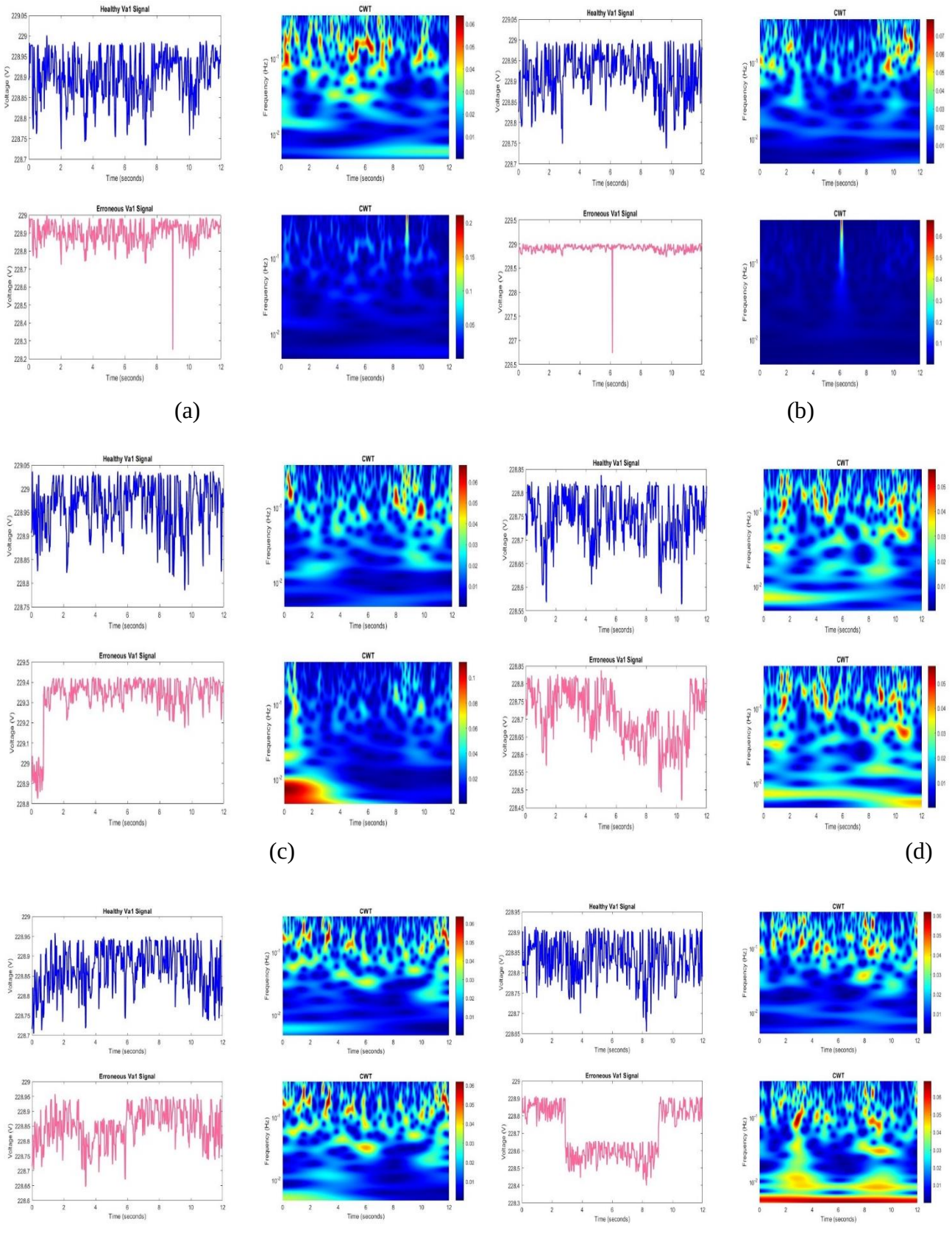


Fig. 5: Voltage signals of phase a under healthy conditions and the three error scenarios, along with their corresponding CWT spectrogram images. Subfigures (a) and (b) illustrate examples from the first error scenario. Subfigures (c) and (d) depict examples from the second error scenario. Subfigures (e) and (f) present examples from the third error scenario.

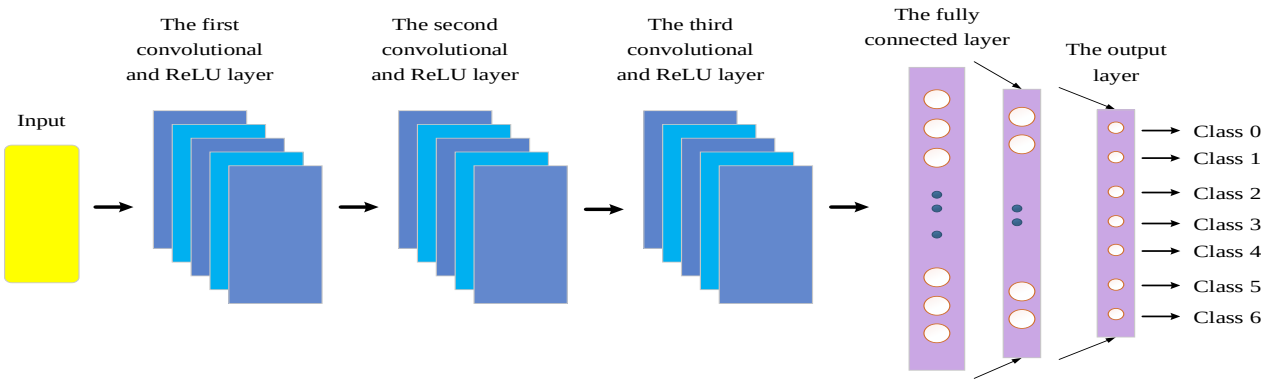


Fig. 6: Classification-based CNN configuration developed for multiclass error detection tasks.

Table 1: CNN configuration of each layer.

Layer Type	Filter Size / #Cells	Input Size	Output Size
2D Conv1 + Batch Norm + ReLU	5x5, 16 filters	32×64×6	32×64×16
2D Conv2 + Batch Norm + ReLU	3x3, 32 filters	32×64×16	32×64×32
2D Conv3 + Batch Norm + ReLU	3x3, 64 filters	32×64×32	32×64×64
Dropout	0.5	32×64×64	32×64×64
FC + Softmax	7 cells	131072x1	7x1

5. Experiments and Results

5.1 Dataset Preparation

To simulate various loading conditions, the loads at four buses were varied at increments of [20%, 40%, 60%, 80%, and 100%], creating diverse operating conditions. Given the five possible load levels at each of the four buses, this setup resulted in a total of 5^4=625 unique cases, covering a comprehensive range of load combinations across the feeder. To address overfitting concerns, 625 distinct runs were generated using MATLAB/Simulink, with each run lasting 3 seconds. During each run, load variations were introduced randomly over time to simulate realistic changes. From these runs, 80% (500 runs) were used for training, and 20% (125 runs) were reserved for testing. Each run is entirely independent, with separate readings, ensuring no data leakage occurred between the training and testing sets.

Error is injected into µPMU voltage measurements through three distinct scenarios. In the first scenario, an error is injected into a single measurement. In the second scenario, an error is introduced at a specific measurement and persists until the end of the signal. In the third scenario, an error is injected during a specified period within the signal. After injecting errors, the CWT was applied to the signal to extract time-frequency features. The resulting CWT coefficients will be used to effectively train and test the classification models.

5.2 Offline training and online error detection

The problem of error detection in µPMU measurements can recast as a classification problem based on the CWT coefficients which are obtained by the µPMUs measurements. In the

proposed methodology, the process of offline training and online error detection plays a crucial role in detecting errors in µPMU measurements. Initially, the offline training phase begins with data collection, where healthy measurements of voltage signals are gathered from the µPMU. To simulate various error scenarios, errors are systematically injected into these healthy measurements to create a comprehensive dataset representing both healthy and erroneous states. After the error injection, CWT is applied to the signals to extract time-frequency features, which serve as the input for the classifier models. The entire dataset comprises 4800 coefficients.

Specifically, of the 4000 coefficients, 3200 coefficients for each class have been used for training the classifier models, while 800 coefficients for each class were set aside for validation, and an additional 800 coefficients have been reserved exclusively for testing. This approach ensures that the testing data remains separate from the training and validation phases. During offline training, ML models like SVM, NN and DL model like CNN are trained using the extracted CWT coefficients. These coefficients are reshaped according to the model requirements. For CNN model, 3D input is utilized, while 2D feature vectors are used for traditional ML models. Stochastic Gradient Descent Method (SGDM) has been used as optimizer for training CNN and hyperparameters such as learning rate, batch size, and epochs are tuned to optimize performance. Table 2 shows the training parameters of CNN. The training data is randomly shuffled before each epoch.

Table 2: Training parameters of CNN.

Optimizer	Sgdm
Learning rate	0.0003
Epoch	10
Batch size	64
Loss function	Cross entropy

For the SVM model, it was trained using a linear kernel function with standardized input data. The fitcecoc function was used to handle the multi-class classification task. For the NN model, a network with three fully connected layers is employed. The first fully connected layer comprises 128 units, followed by a second fully connected layer with 64 units. ReLU activation functions are applied after each of these layers to introduce non-linearity. The final fully connected layer is designed to match the number of output classes, which is then followed by a softmax layer for classification. The network is trained using SGDM and its performance is validated on the test subset. Classifier models have been trained using a personal laptop having 8 GB RAM and Intel(R) Core (TM) Intel i5 processor with 3.11 GHz speed using single GPU environment. This work has been implemented using MATLAB R2022a. Once the offline training is complete, the online error detection phase begins. In this phase, unseen data is fed into the trained models to evaluate their ability to classify and detect errors in real-time. The models process the incoming µPMU data, extract features using the same CWT methodology, and then predict whether the signal is healthy or erroneous. The performance of the classifiers in this phase is assessed using metrics such as accuracy, precision, recall, and F1-score. By comparing the predicted outputs with the actual labels, the

effectiveness of the error detection system is evaluated. Fig. 7 shows the flowchart of the proposed CNN-based error detection approach. The µPMUs data is first used for offline training of the classifier models. The trained models are then used for online detection procedures. The approach for detecting errors in µPMU voltage measurements is outlined as follows:

1- First, the data measurements are obtained from the two µPMUs under different load conditions. Then, the error is injected into the voltage magnitude measurements randomly.
2- The measurements are divided into small windows (64 measurements per window). CWT is then applied to each window of µPMU measurements to obtain the wavelet coefficients using morlet.
3- Such coefficients with their corresponding classes are then inputted into the classifier models. Next, data partitioning is used to split the data into training data and test data.
4- For SVM and NN, the 4D data is then reshaped into a 2D array suitable for SVM and NN input.
5- Once the model is trained using training data, additional µPMU measurements are used beyond the training set to verify the classifier models accuracy. Through comparison, the confusion matrix is evaluated.

5.3 Evaluation metrics

Evaluation metrics in deep learning tasks are essential for optimizing classifiers [30]. In data classification problems, these metrics are applied at two key stages: the training stage and the testing stage. During the training stage, evaluation metrics guide the optimization of the classification model, helping to identify the best solution for improving the model's accuracy in predicting future outcomes. In the testing stage, the metrics serve as evaluators, measuring the effectiveness of the trained classifier when applied to new, unseen data. To assess the performance of the classifier models, commonly used metrics include accuracy, recall, precision, and F1 score [28]. These metrics, often applied in multiclass classification tasks, are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

where true negative (TN) and true positive (TP) are the correctly predicted negative and positive values, respectively, which are successfully classified. In addition, false negative (FN) and false positive (FP) are the incorrectly predicted negative and positive values, respectively.

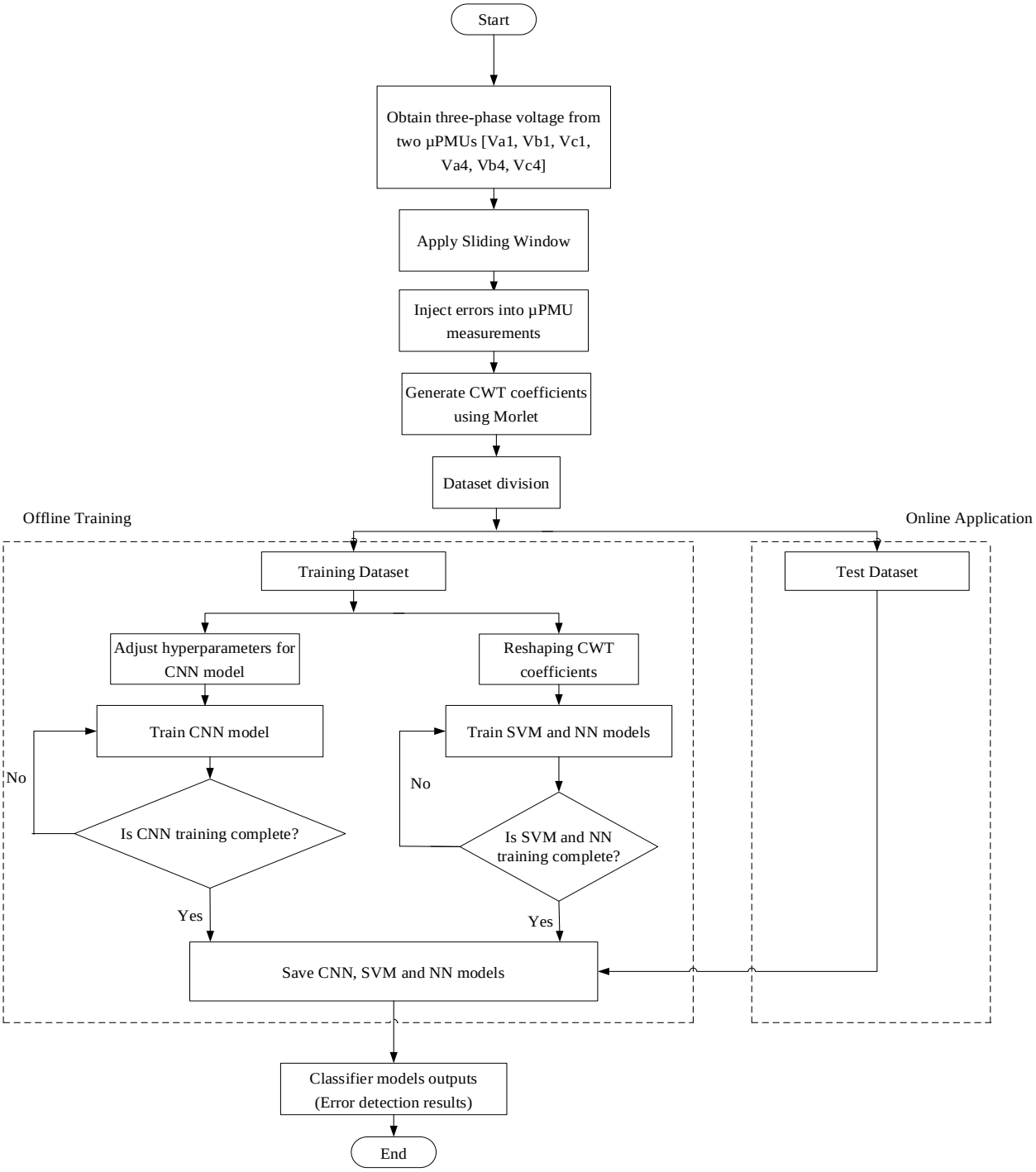


Fig. 7: The flowchart of the proposed technique for error detection in μPMU measurements.

5.4 Results and discussion

The proposed method has been tested on the test system described in section 4. A multi-class classification problem of 7-class has been presented for error detection in μPMU voltage measurements. Three classifier models have been trained using the CWT coefficient dataset, alongside their corresponding labels. The same training, validation and testing sets have been used for 7-way classification across all three models, as summarized in Table 3.

Table 3: Training, validation and testing dataset used for three models.

Class label	Class code	Training set	Validation set	Testing set
Healthy signal	Class 0	3200	800	800
Error in Va1	Class 1	3200	800	800
Error in Vb1	Class 2	3200	800	800
Error in Vc1	Class 3	3200	800	800
Error in Va4	Class 4	3200	800	800
Error in Vb4	Class 5	3200	800	800
Error in Vc4	Class 6	3200	800	800
Total		22400	5600	5600

The number of epochs used for CNN training is set to 10 and the maximum number of iterations is 3,500 where 350 iterations have been used for one epoch. Over a single GPU, the CNN model parameters are adjusted using SGDM with back-propagation through time. The training progress curve for the CNN model, shown in Fig. 8, highlights the model's performance in terms of accuracy and loss across 10 epochs. From the first epoch, the model achieved high accuracy, indicating that it effectively learned relevant features early in the training process. This rapid improvement suggests that the model architecture and parameter settings are well-suited for the task. However, training was set to run for a full 10 epochs to ensure the model's stability and convergence, allowing any potential minor fluctuations to be smoothed out. By the final epoch, the validation accuracy reached 99.71%, with minimal loss observed, confirming the model's strong generalization and robustness across both training and validation datasets. In addition, the total number of parameters in the CNN model is 943,287, which reflects the model's complexity and capacity to capture detailed features for accurate error detection in µPMU voltage measurements. The confusion matrices for three models are presented in Fig. 9, providing insight into the models' ability to classify the different classes accurately. While the model consistently identified all healthy cases correctly, it confused some of the error cases in the voltage measurements by misclassifying them as healthy cases.

The confusion matrix for the NN model shows a high overall classification accuracy, with all instances in Class 0 being correctly classified. However, the model makes some minor misclassifications in other classes, where certain instances are incorrectly labeled as Class 0. The highest misclassification rate was observed in Class 2 and Class 5, where 9 instances from each class were incorrectly classified as Class 0. This indicates that while the model performs well overall, there is a tendency to misclassify instances. The SVM confusion matrix illustrates the model's strong classification performance across all classes. SVM achieves high accuracy, with minimal misclassifications, particularly in Class 2 and Class 5, where a few instances are incorrectly classified as Class 0.

The CNN confusion matrix reveals the model's exceptional performance, with nearly perfect classification accuracy across all target classes. CNN correctly classifies almost all instances in each class, achieving 100% accuracy for Class 0 and Class 6. In Class 2, 4 instances are

misclassified as Class 0, resulting in a slight reduction in accuracy for this class to 99.5%. However, compared to the NN and SVM models, the CNN model demonstrates the highest overall performance. Table 4 presents a comparison of evaluation metrics for the three models. CNN achieved the highest accuracy, recall, F1-score, and precision among the three models. This indicates that the CNN model is highly effective for detecting errors in µPMU voltage measurements, demonstrating minimal misclassifications and strong consistency across all metrics. The time analysis revealed that generating the CWT coefficients took 2 ms, while testing each sample using the CNN model required 5 ms, equivalent to 320 ms per batch. Consequently, the total time for data generation and testing amounts to 7 ms, demonstrating the model's efficiency and suitability for real-time error detection in µPMU voltage measurements.

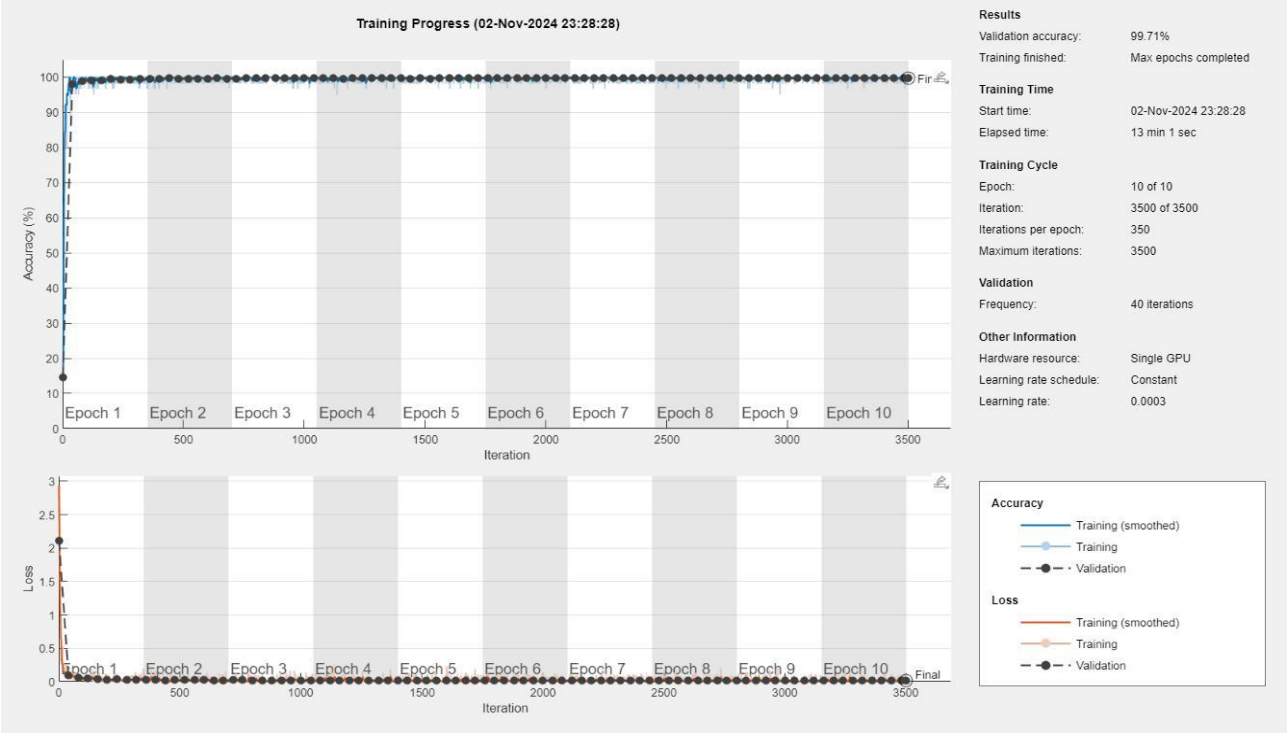
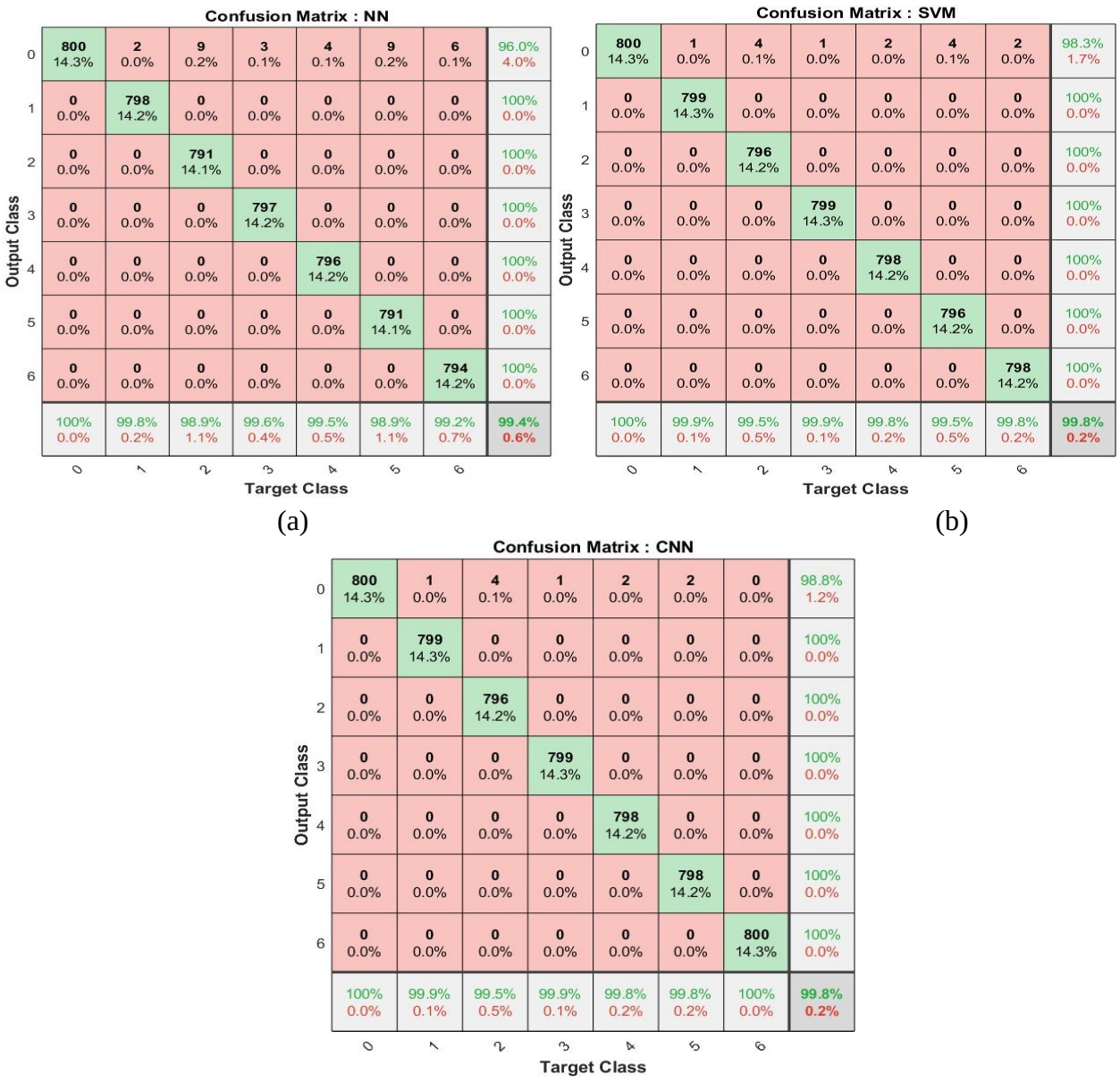


Fig. 8: Training progress of the CNN model.

Table 4: Evaluation metrics for three models for error detection in µPMU voltage measurements.

Models	**Accuracy (%)**	**Recall (%)**	**F1-Score (%)**	**Precision (%)**
NN	99.4107	99.434	99.415	99.411
SVM	99.75	99.754	99.751	99.75
CNN	99.821	99.824	99.822	99.821



(a) (b)

(c)

Fig. 9: The confusion matrices of classifier models for error detection in voltage measurements of two µPMUs a) NN, b) SVM, and c) CNN.

6. Conclusions and future work

This paper presents a novel approach for error detection in µPMU voltage measurements using a combination of CWT and CNN. By utilizing CWT coefficients instead of raw voltage signals, the proposed method effectively captures the distinguishing characteristics of the signals, leading to improved feature extraction and pattern recognition. Based on the results presented in this study, the proposed CNN-based approach for error detection in µPMU voltage measurements demonstrates superior performance, achieving a high classification accuracy of 99.821%, which surpasses other machine learning models like SVM and NN due to its powerful capability of extracting features and latent information. This means that CNN not only has an outstanding capability in dealing with the image but also has the potential to deal with CWT coefficients. The effectiveness of the model is further validated through

metrics such as precision, recall, and F1-score, which consistently confirm its robust classification capabilities across all seven classes. The model's ability to generalize is evident, as it was tested on a separate dataset not included in the training set, confirming its robustness on unseen data. A time analysis was conducted to address real-time application concerns, measuring the time required for data generation and testing, with a total time of 7 ms. These results provide insight into the model's performance and potential applicability in real-world smart grid environments. Therefore, combining CWT coefficients with a CNN model has shown promising results in detecting and classifying errors in μPMU voltage measurements, contributing to more accurate and stable monitoring in modern power grids. In future work, this approach will be extended to error detection in μPMU current measurements. Given the difficulty in distinguishing between error occurrences and load variations in current measurements, more complex architectures, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, may be necessary. These advanced models could improve error detection in current measurements by better capturing temporal patterns and dependencies in the data.

APPENDIX A

The cable, transformer and load data for three-phase unbalanced distribution grid sub-feeder NBS_8 of Assiut university are given in Table 5, Table 6 and Table 7, respectively.

Table 5: Cable data for unbalanced NBS8 sub-feeder.

Cable Data					
From	To	Z_s (Ω)	Z_m (Ω)	Y_s (μS)	Y_m (μS)
NBS8	46	0.13521+ 0.412i	0.0077+ 0.3604i	19.733i	-3.867i
46	47	0.0311+ 0.0873i	0.00053+ 0.0751i	4.767i	-0.933i
47	48	0.1071+ 0.3238i	0.00517+ 0.2829i	15.83i	-3.0٦7i
48	49	0.1211+ 0.3675i	0.00641+ 0.3215i	17.7٦7i	-3.4٣3i

Table 6: Transformer data for unbalanced NBS8 sub-feeder**.**

Transformer Data	
Transformer	Z^a_{trlv}=Z^b_{trlv} = Z^c_{trlv} in (Ω)
46tr	0.001435+0.0081791i
47tr	0.0033144+0.012761i
48tr	0.0033144+0.012761i
49tr	0.001435+0.0081791i

Table 7: Load data for unbalanced NBS8 sub-feeder.

Load Data						
Bus No.	Sa (MVA)	Sb (MVA)	Sc (MVA)	p.f.a	p.f.b	p.f.c
46l	0.144	0.145	0.146	0.91	0.91	0.91
47l	0.11	0.111	0.112	0.89	0.89	0.89
48l	0.112	0.113	0.111	0.9	0.90	0.9
49l	0.144	0.145	0.146	0.92	0.92	0.92

References

[1] J. A. Momoh, Smart grid: fundamentals of design and analysis, Vol. 33, John Wiley & Sons, 2012.

[2] A. E. Saldaña-González, A. Sumper, M. Aragüés-Peñalba, and M. Smolnikar, "Advanced distribution measurement technologies and data applications for smart grids: A review," Energies, vol. 13, no. 14, 3730, July 2020.

[3] A. Von Meier, E. Stewart, A. McEachern, M. Andersen and L. Mehrmanesh, "Precision micro-synchrophasors for distribution systems: A summary of applications", IEEE Transactions on Smart Grid, vol. 8, no. 6, pp. 2926-2936, 2017.

[4] M. Jamei, A. Scaglione, C. Roberts, E. Stewart, S. Peisert, C. McParland, and A. McEachern, "Anomaly detection using optimally placed μPMU sensors in distribution grids," IEEE Trans. on Power Systems, vol. 33, no. 4, pp. 3611–3623, July 2018.

[5] Y. Zhou, R. Arghandeh, and C. J. Spanos, "Partial knowledge datadriven event detection for power distribution networks," IEEE Trans. on Smart Grid, vol. 9, no. 5, pp. 5152–5162, Sep. 2018.

[6] A. Aligholian, A. Shahsavari, E. Cortez, E. Stewart, and H. MohsenianRad, "Event detection in micro-pmu data: A generative adversarial network scoring method." 2020 IEEE Power & Energy Society General Meeting (PESGM). IEEE, 2020. pp. 1-5.

[7] A. A. Elsayed, M. A. Abdellah, M. A. Mohamed, and M. A. E. Nayel, "μPMU Hardware and Software Design Consideration and Implementa-tion for Distribution Grid Applications," Advances in Science, Technology and Engineering Systems Journal, vol. 7, no. 4, pp. 59-71, 2022.

[8] P. A. Pegoraro, K. Brady, P. Castello, C. Muscas, and A. von Meier, "Compensation of systematic measurement errors in a PMU-based monitoring system for electric distribution grids," IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 10, pp. 3871-3882, 2019.

[9] H. Zhang and A. A. Arkadan, "B–H Curve-Based Model for CT Error Evaluation," in IEEE Transactions on Magnetics, vol. 55, no. 6, pp. 1-4, June 2019.

[10] D. Shi, D. J. Tylavsky, and N. Logic, "An adaptive method for detection and correction of errors in PMU measurements," IEEE Transactions on Smart Grid, vol. 3, no. 4, pp. 1575-1583, Aug. 2012.

[11] L. Vanfretti, J. H. Chow, S. Sarawgi, and B. Fardanesh, "A phasor data-based state estimator incorporating phase bias correction," IEEE Trans. Power Syst., vol. 26, pp. 111–119, Feb. 2011.

[12] I. N. Kolosok, E. S. Korkina, and A. E. Mahnitko, "Detection of systematic errors in PMU measurements by the power system state estimation methods," 2015 56th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), IEEE, pp. 1-4, 2015.

[13] X. Wang, D. Shi, Z. Wang, C. Xu, Q. Zhang, X. Zhang, and Z. Yu, "Online calibration of phasor measurement unit using density-based spatial clustering," IEEE Trans. Power Del., vol. 33, no. 3, pp. 1081–1090, Jun. 2018.

[14] K. Mahapatra, N. R. Chaudhuri, and R. Kavasseri, "Bad data detection in PMU measurements using principal component analysis," *2016 North American Power Symposium (NAPS)*, IEEE, pp. 1-6, 2016.

[15] J.L. Rueda, C.A. Juarez, and I. Erlich, "Wavelet-based analysis of power system low-frequency electromechanical oscillations," IEEE Transaction on Power System, vol. 26, no. 3, pp. 1733-1743, August 2011.

[16] W. Gao and J. Ning, "Wavelet-based disturbance analysis for power system wide-area monitoring," IEEE Trans. Smart Grid, vol. 2, no. 1, pp. 121–130, Mar. 2011.

[17] I. Kolosok and L. Gurina, "Wavelet analysis of PMU measurements for identification of cyber attacks on TCMS," 2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), IEEE, 2018, pp. 1-4.

[18] K. Mahapatra, N. R. Chaudhuri, and R. Kavasseri, "Online bad data outlier detection in PMU measurements using PCA feature-driven ANN classifier," *IEEE Power & Energy Society General Meeting*, IEEE, pp. 1-5, 2017.

[19] M. A. Merazy, R. Hasan, M. A. Mohamed, and M. Nayel, "Error Detection in μPMU Voltage Measurements Using Artificial Neural Network," 2021 22nd International Middle East Power Systems Conference (MEPCON). IEEE, 2021, pp. 569-574.

[20] G. Khare, A. Mohapatra, and S. N. Singh, "A real-time approach for detection and correction of false data in PMU measurements," Electric Power Systems Research, vol.191, 2021, 106866.

[21] Y. Zhu, X. Xu, and Z. Yan, "Hybrid clustering-based bad data detection of PMU measurements," *Energy Conversion and Economics*, vol. 2, no. 4, 2021, pp. 235-247.

[22] Z. Yang, H. Liu, T. Bi, and Q. Yang, "Bad data detection algorithm for PMU based on spectral clustering," Journal of Modern Power Systems and clean energy, vol. 8, no. 3, 2020, pp. 473-483.

[23] N. C. F. Tse, "Practical application of wavelet to power quality analysis," In: 2006 IEEE Power Engineering Society General Meeting, Montreal, Que., Canda, June 2006, pp. 5-10.

[24] F. Shiri and B. Mohammadi-ivatloo, "Identification of interarea oscillations using wavelet transform and phasor measurement unit data," Int. Trans. Electr. Energ. Syst., 2014.

[25] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 2, pp. 645–657, Feb. 2017.

[26] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in Neural networks: Tricks of the trade. Springer, 2012, pp. 437–478.

[27] Y. Wang, Y. Li, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," Applied Sciences, vol. 10, no. 5, pp. 1-20. Mar. 2020.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] A. A.Elaziez, M. A. Mohamed, M. AbdelRaheem and M. A.Nayel, "Optimal μPMU Placement and Current Channel Selection Considering Running Cost for Distribution Grid," *2020 IEEE International Conference on Power Electronics, Smart Grid and Renewable Energy (PESGRE2020)*, pp. 1-8, 2020.

[30] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information processing & management, vol. 45, no. 4, pp. 427–437, 2009.