Power Grid Online Surveillance through PMU-Embedded Convolutional Neural Networks

Shiyuan Wang, *Student Member, IEEE*, Li Li, *Student Member, IEEE*, and Payman Dehghanian, *Member, IEEE*

Abstract—Power grid operation continuously experiences state transitions caused by the internal and external uncertainties, e.g., equipment failures and weather-driven faults. This prompts an observation of different types of waveforms at the measurement points (substations) in power systems captured by the phasor measurement units (PMUs) and intelligent electronic devices (IEDs) embedded with PMU functionality, e.g., digital relays and fault recorders. The PMU should be, hence, equipped with either one synchrophasor estimation algorithm (SEA) that is accurate and robust to many different types of signals any time across the network, or should adaptively select the promising SEA, among an embedded suite of algorithms. This paper proposes a PMU-embedded framework that can ensure real-time grid surveillance and potentially enables adaptive selection of SEA for more accurate synchrophasor estimation. Our proposed framework is consisted of two components: (i) a pseudo continuous quadrature wavelet transform (PCQ-WT) algorithm using a modified Gabor wavelet transform, which generates the featured-scalograms; and (ii) a convolutional neural network (CNN), that classifies the events based on the extracted features in the scalograms. Our experiments demonstrate that the proposed framework achieves state-of-the-art classification accuracy on multiple types of prevailing events in power grids, through which an enhanced grid-scale situational awareness in real-time can be realized.

Index Terms—Convolutional Neural Network (CNN); feature extraction; phasor measurement unit (PMU); waveform classification; wavelet transform (WT).

I. INTRODUCTION

With the widespread deployment of synchrophasor technology in modern power grids, system monitoring and control settings have been revolutionized into a new era with high-resolution measurements, fulfilling an enhanced grid situational awareness [1]–[4]. Synchrophasor measurements, captured across the network via phasor measurement units (PMUs), have transformed many applications, e.g., power system model validation, state-estimation, dynamic stability, online monitoring, protection, control of the grid, and post-event analysis [5]–[7]. IEEE standard C37.118.1-2011 [8] has defined the PMU expected outputs—i.e., magnitude, phase angle, frequency, and rate of change of frequency (ROCOF)—and their corresponding desirable accuracy. These outputs are obtained from synchrophasor estimation algorithms (SEAs) which are primarily driven by mathematical approximations. In most cases, and irrespective of the focused application, marketplace PMUs are typically furnished with one SEA tool, each unleashing distinctive advantages and limitations, solely valid to one or a few certain applications [9]–[12]. Typically,

the waveforms fed into PMUs have variant behaviors; for instance, phasor magnitudes and phase angles go through step changes during faults, and the waveform measurements could be noisy. Besides, unbalanced load, voltage surge or sag, harmonics, and frequency drift are also common phenomena in electrical power networks [13]–[15]. In dealing with the above conditions, dynamic SEA based on time-domain signal processing techniques were applied. Some research efforts have proposed a single PMU equipped with only one sophisticated SEA, which is deemed to respond to various prevailing conditions [16]–[18]. Laboratory tests and field observations have revealed how inefficient the PMU measurements could be, if this "one-size-fits-all" SEA is applied to capture both static and dynamic features and peculiarities, when facing different operating states [19]–[21]. To meet the growing demand for high-speed, low-latency, and yet absolutely accurate measurements in PMUs, a more efficient mechanism that provides online event detection and assists selecting the right SEA at the right time is desired.

Waveform patterns and signatures can be extracted from voltage and current signals via wavelet transforms (WT), where transitions of power grid to different operating states and events reveal unique features in time-frequency domain [13], [22], [23]. Built on the WT-extracted features, a machine learning mechanism can be employed to detect and classify the events scalograms [14], [24]. Motivated by the success of CWT in signal processing and convolutional neural networks (CNNs) in pattern recognition, we propose a novel online surveillance framework that classifies different events in power grids and potentially enables an adaptive SEA selection in PMUs for more accurate phasor measurements. The paper's main contributions are as follows:

- We propose a pseudo-continuous quadrature wavelet transform (PCQ-WT) that effectively captures the power waveform features corresponding to different events.
- A CNN mechanism is developed to classify the scalograms generated by PCQ-WT, achieving a high event classification accuracy (92.84±1.20%) for twelve distinct modes of grid prevailing conditions.
- The combination algorithms of the two steps process the input signals in a real-time manner (2.24±0.54 ms), which leaves a large time margin to the subsequent SEA modules within PMUs to function.
- The proposed framework shares the exact same input signals of the PMU, with no additional devices or investments, making it an economically viable technology to be embedded within the existing PMUs.

S. Wang, L. Li, and P. Dehghanian are with the Department of Electrical and Computer Engineering, The George Washington University, Washington, DC 20052 USA (e-mails: [shiyuan1225; lili1986; payman]@gwu.edu).

This paper is organized as the following: Section II introduces a background on continuous wavelet transforms (CWT) and the pattern classification through convolutional neural networks (CNNs). Section III details the proposed online surveillance framework, consisted of (i) feature extraction from voltage and current signals via pseudo-continuous quadrature wavelet transform (PCQ-WT), and (ii) event classification via CNN. Case studies and experiment results are analyzed in Section IV, and the conclusions come in Section V.

II. BACKGROUNDS AND MOTIVATIONS

A. Power Waveform Modeling

To solve a classification problem, a mathematical representation of signals in power grid is needed. The three phase time-domain sinusoidal signals fed into the PMU can be represented by the following:

$$x_{ph}(t) = A_{ph}(t)cos\left(2\pi \int_0^t F(\tau)d\tau + \phi_{ph}(t)\right) \quad (1)$$

where $x_{ph}(t)$ is a one-dimension (1-D) waveform measured from each phase; $A_{ph}(t)$, $F(\tau)$, and $\phi_{ph}(t)$ are the instantaneous magnitude, fundamental frequency, and phase angle in each phase respectively. During both transient and steady state, the waveform in each phase can be expressed by a summation of different orders of harmonic components. Thus, the actual waveform in each phase is

$$x_{ph}(t) = \sum_{h=1}^{H} A_{ph,h}(t)cos\left(2\pi \int_0^t F_h(\tau)d\tau + \phi_{ph,h}(t)\right) \quad (2)$$

where h is the order of harmonics, and H is the maximum order of harmonic of interest. In a particular grid operation condition, different values of $A_{ph,h}(t)$, $F_h(\tau)$, $\phi_{ph,h}(t)$ will appear in the three-phase power signal, which lead to patterns and peculiarities. Therefore, an event can be detected and classified accordingly. To simplify the time-domain sinusoidal signals analysis and without acquiring the rotating reference frame in Park transformation [25], Clarke transformation is applied to convert the three phase signal from ABC to $\alpha\beta$-frame [26] by the following equations:

$$x_{\alpha\beta}(t) = x_\alpha(t) + jx_\beta(t), \quad (3)$$

and

$$\begin{bmatrix} x_\alpha(t) \\ x_\beta(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} x_A(t) \\ x_B(t) \\ x_C(t) \end{bmatrix} \quad (4)$$

Since power waveforms contain different frequency components, multi-resolution waveform analysis techniques are suitable to extract the features, i.e., amplitude, frequency and phase. The most commonly-used technique is short-time Fourier transform (STFT) [2], [3], [27], [28] and wavelet analysis [29]–[31]. However studies show that STFT is attributed a higher computational burden and cost more time which leads to a lower time resolution than wavelet analysis in time-frequency domains [32], [33]. Also, comparing the spectrogram in Fig. 1(a) and (b) with the scalogram of CWT in Fig. 1(c)(d)(e) and (f), it can be seen that STFT
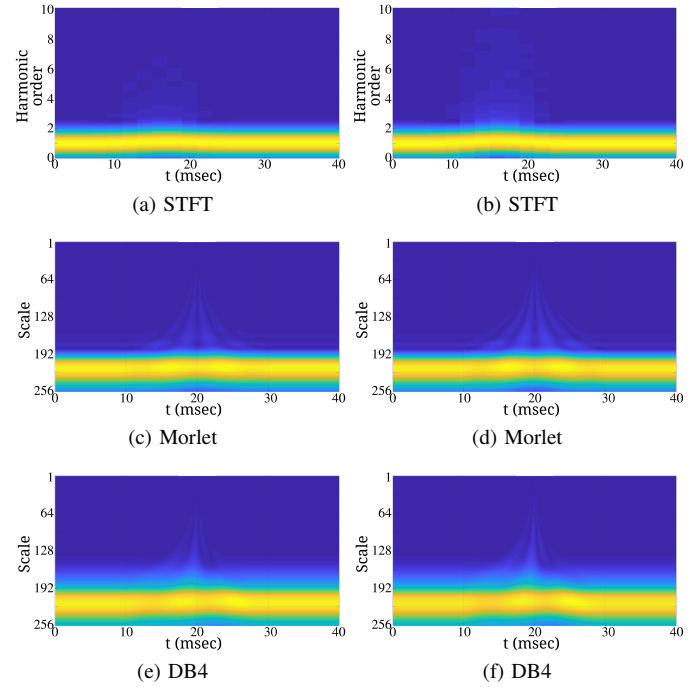

Fig. 1. Comparison of the STFT vs. Morlet CWT vs. DB4 CWT during: (a),(c),(e) 2Hz frequency jump; (b),(d),(f) 40° phase jump.

outperforms CWT in frequency accuracy, while CWT provides more conspicuous results for feature extraction. Therefore, wavelet analysis is chosen as the mathematical tool of interest for feature extraction and online waveform monitoring.

B. CWT and Pseudo-CWT (PCWT)

The wavelet transform is obtained by computing the cross-correlation between the signal of interest—$x_{\alpha\beta}(t)$ and designated wavelets. This process is defined as follows:

$$X_{\alpha\beta}(\omega|a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x_{\alpha\beta}(t)\Psi^*(\frac{t-b}{a})dt \quad (5)$$

where $\Psi(t)$ is the mother wavelet, $*$ denotes the complex conjugate, a and b are the scaling factors and the time shift, and $\Psi(\frac{t-b}{a})$ is one of the "daughter wavelets" of $\Psi(t)$ [34], [35]. With different selections of a and b, a wavelet bank is then determined. By selecting proper intervals for the *continuous* scaling factor along with the time shift, a continuous-wavelet transform (CWT) is achieved [36], [37]. In a PMU, the real-time signals are sampled, and discrete signal processing is actually applied. Due to the limited computational capacity of hardware, the number of scaling factors are finite, thus the mathematical behavior of the CWT within the processor is *pseudo* continuous with a set of discrete scaling factors. So, here, the pseudo-CWT (PCWT) with one of the discrete scaling factors is defined as follows:

$$X_{\alpha\beta}[\omega|a_k,b_k] = \frac{1}{\sqrt{|a_k|}} \sum_{n=0}^{W-1} x_{\alpha\beta}[n]\Psi^*[\frac{nT_s - b_k}{a_k}] \quad (6)$$

where T_s denotes the sampling interval, and W stands for the window (buffer) length. In a CWT, each daughter wavelet

needs to cover a designated frequency range that reflects the features in time-frequency analysis. The central frequency of the daughter wavelets can be approximated by the following relationship with the scaling factor:

$$f = F_c/a_k \qquad (7)$$

where F_C is the central frequency of the mother wavelet [36]. When a vector of scaling factors with length K is chosen, the wavelet bank Ψ and the extracted features at time instant n are expressed by the following:

$$\Psi^{K \times W} = \begin{bmatrix} \Psi\left[\frac{nT_s - b_1}{a_1}\right] \\ \vdots \\ \Psi\left[\frac{nT_s - b_k}{a_k}\right] \\ \vdots \\ \Psi\left[\frac{nT_s - b_K}{a_K}\right] \end{bmatrix}, \quad \mathbf{X}_{\underline{\omega}}^{K \times 1}[n] = \begin{bmatrix} X_{\omega 1}(a_1, b_1) \\ \vdots \\ X_{\omega k}(a_k, b_k) \\ \vdots \\ X_{\omega K}(a_K, b_K) \end{bmatrix}. \qquad (8)$$

To cover a sufficiently wide range of frequency and provide adequate pattern information for time-frequency analysis on any given operating condition, the central frequency of mother wavelets F_C and largest scaling factor a_K must satisfy the following condition:

$$\frac{F_c}{a_K} < f_0 < F_c, \qquad (9)$$

where f_0 is the frequency of the signal of interest. F_c needs to be chosen from a higher frequency range than the maximum frequency of interest and scaled down by a_k. Hence, as a_k increases, the corresponding frequency of the PCWT output decreases in the frequency domain.

Once the scaling factors are chosen, then (8) is able to generate the wavelet bank and a series time bin of $\mathbf{X}_{\underline{\omega}}^{K \times 1}$ along the time instant, i.e. a scalogram of PCWT is obtained.

C. Convolutional Neural Networks (CNNs)

At this stage, we consider the obtained scalogram as 2-D images, and the process of event classification turns to an image classification. The conventional paradigm for image classification is to manually design the feature extractor and reduce the dimensionality of the data, the second phase is to employ a classifier to classify the lower dimensional features. This paradigm highly depends on the design of the feature extractor, and manually designing features for a complex task requires a great deal of human time and effort; it can take decades for an entire community of researchers [38]. In contrast, Convolutional Neural Networks (CNNs) are able to learn the feature extractor automatically and have been proven very successful in the broad image-related tasks [39]–[43]. By definition, CNNs are simply neural networks that use convolution in place of general matrix multiplication in that least on of their layers [38]. In general, the implementation of the convolution are actually cross-correlations and defined by

$$s^p(m, n) = \sum_u \sum_v \sum_w I^u(m+v, n+w) K^p(v, w), \qquad (10)$$

where $s^p(m, n)$ is the output of the convolutional layer at position (m, n) and p-th channel, I^u is the u-th channel of the image/data volume, and K^u is the u-th convolutional kernel. A complex convolutional layer is composed by small number of complex layers [38] and expressed by the following:

$$I_l = pool\left(\sigma(s)\right), \qquad (11)$$

where I_l represents the output volume of the l-th layer, $\sigma(\cdot)$ is the non-linearity of the neurons, and $pool(\cdot)$ is a down sampling procedure. By stacking the convolutional layers, the abstraction capacity of the network generally increases [44].

The representations of the last convolutional layer are expanded to vectors and processed by the general fully-connected layers, which transform the representations with more nonlinearities and into spaces with different (higher or lower) dimensions. The final layer of a CNN usually reduces the dimensionality of the representations to the number of the classes; cross-entropy [45] is then employed to measure the "goodness" of the classification (Kullback-Leibler divergence between the predicted distribution and the target distribution). Finally, gradients of the cross-entropy loss function with respect to the parameters in the CNN are used to train the CNN by back-propagation.

III. Proposed Feature Extraction and Event Classification by CNNs

A. PCQ-WT Based Feature Extraction

Gabor wavelets have been widely used in two-dimension (2-D) pattern recognition [46]–[48]. In order to simplify the design and increase the computational efficiency, a modified complex Gabor wavelet from [48] is adopted in this paper and written as

$$\Psi(t) = \underbrace{exp\left(j\omega_c(t - b)\right)}_{\text{Periodic Component}} \cdot \underbrace{exp\left(-\frac{(t - b)^2}{\alpha_0^2}\right)}_{\text{Gaussian Envelope}} \qquad (12)$$

where w_c is the central frequency in radiant. The Fourier transform of this Gabor wavelet is

$$\mathbf{F}_{\Psi}(\omega) = \alpha_0\sqrt{\pi} \cdot exp(-j\omega b) \cdot exp(-\frac{\alpha_0^2}{4}(\omega - \omega_c)^2) \qquad (13)$$

One can see that the Fourier transform of Gabor wavelet is also a function on the theme of Gabor wavelet. Although they do not have the orthogonal property, because of

$$|\mathbf{F}_{\Psi}(\omega_c \pm \epsilon)| \neq 0, \qquad (14)$$

where ϵ is a small value; according to (13), this Gabor wavelet has a characteristic of predictable narrow-bandwidth. By properly selecting α_0, one of the PCQ-WT can cover a desired range of frequencies. And the time shift b plays no magnitude impact on (13) and (14). Therefore to ease the derivation, let $b = 0$, then the CWT of Gabor wavelet computation turns to

$$\begin{aligned} X_{\alpha\beta}(\omega_0|a, b=0) &= \int_{-\infty}^{\infty} x_{\alpha\beta}(t)\Psi^*(\frac{t}{a})d\tau \\ &= \int_{-\infty}^{\infty} exp\left(j(\omega_o - \frac{\omega_c}{a})t - \frac{t^2}{a^2\alpha_0^2}\right)d\tau \end{aligned} \qquad (15)$$
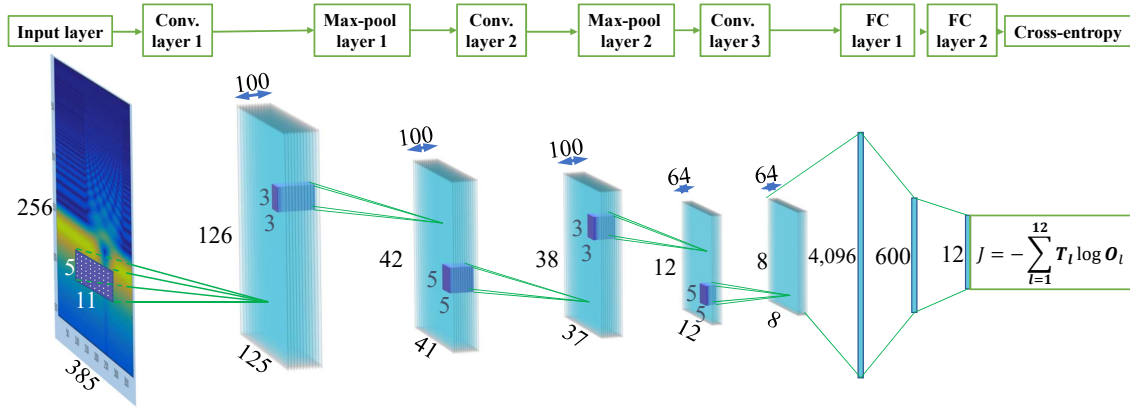
Fig. 2. The proposed classification CNN architecture for the scalogram extracted from PCQ-WT. J is the cross-entropy loss, $\boldsymbol{T}_l$ is the true probability distribution over labels and $\boldsymbol{O}_l$ is the predicted probability distribution by the networks [45].

According to Hubbard–Stratonovich transformation [49]:

$$exp(-\frac{\alpha}{2}x^2) = \sqrt{\frac{1}{2\pi\alpha}} \int_{-\infty}^{\infty} exp\left(-\frac{y^2}{2\alpha} - jxy\right) \mathrm{d}y \quad (16)$$

The CWT of Gabor wavelet in (15) becomes

$$X_{\alpha\beta}(\omega_0|a, b=0) = a\alpha_0\sqrt{\pi}exp\left(-\frac{\alpha_0^2}{4}(a\omega_0 - \omega_c)^2\right). \quad (17)$$

It can be seen that when $\omega_0 = \omega_c/a$, (17) reaches its maximum, the dominant feature of the expected frequency is revealed. To make each frequency of interest sharing an equivalent maximal magnitude, make

$$a\alpha_0 = \omega_c/\gamma, \quad (18)$$

where γ is a constant. By this expression and according to (12), the Gaussian envelope in Gabor wavelet is adaptive to different frequencies. The discrete form of the Gabor wavelet used in this paper is

$$\Psi[n|a_k, b_k] = exp\left(j\frac{\omega_c T_s(n - b_k)}{a_k}\right) exp\left(-\frac{T_s^2(n - b_k)^2}{a_k^2\alpha_0^2}\right), \quad (19)$$

When applying the complex Gabor wavelet with a set of discrete scaling factors, we achieve the proposed PCQ-WT, and it can be written as

$$X_{\alpha\beta}(\omega_k|a_k, b_k = 0) = \sum_{n=0}^{W-1} x_{\alpha\beta}[n]\Psi^*[-\frac{T_s m}{a_k}]$$
$$= \sum_{n=0}^{W-1} x_{\alpha\beta}[n]exp\left(-j\frac{\omega_c}{a_k}T_s n - \frac{T_s^2 n^2}{a_k^2\alpha_0^2}\right). \quad (20)$$

If one determines the frequencies of interest and designs the Gabor wavelet bank properly, a vector $\mathbf{X}_{\alpha\beta,\underline{\omega}}$ that consists of a set of PCQ-WTs can be obtained, and it is able to conduct time-frequency analysis and generate scalograms, extracting features from the waveform in frequencies of interest.

B. Event Classification by CNNs

Pursuing development of an event detection mechanism in power systems, one needs to understand that the scalograms of the waveforms generated by PCQ-WT convey valuable information on the events; the process of event detection is therefore converted to a supervised classification problem on the scalograms. However, the classification for the 2-D scalograms is challenging due to their high dimensionality. Specifically, every frame of the obtained scalogram has $scales \times time\ bins$ dimension (usually hundreds by hundreds); such high dimensional data is prohibitive for most of the conventional pattern classification approaches. We treat the PCQ-WT scalograms as 2-D images and propose a CNN-based architecture to classify the events concealed in the scalograms (images). As the scalogram classification is not with very high abstraction level, we did not transfer any very deep neural networks to the task; instead, the proposed CNN has a simple architecture that meets the requirements of a synchrophasor, yet with very fast test times.

Our proposed CNN contains five layers: three convolutional (Conv.) layers and two fully-connected (FC) layers. The architecture of the CNN can be seen in Fig. 2, and its specifications will be introduced in Section IV-B. The overall proposed online surveillance framework embedded within PMUs is demonstrated in Fig. 3. This framework can work as an standalone event detection and classification tool within PMU, or it can assist the phasor processor to select a proper SEA, if a set of SEAs were equipped and available within the PMU.

IV. CASE STUDY AND EXPERIMENTS

A. PCQ-WT Parameter Settings

The sampling frequency we used in this paper is $Fs = 9600$Hz. Up to 50th order (3000Hz) of harmonic is considered; therefore, mathematically, a frequency spectrum that ranges from 1Hz to 3000Hz is requisite. This paper chooses the mother wavelet's central frequency as 0.32 times of F_s, in other words $F_c = 0.32$ (normalized) for PCQ-WT. The scaling factor should be in the range of [1, 3072]. However, the total 3072 of PCQ-WT could lead to a high computational burden and large memory demanding; hence, a down-sampling of the scaling factor is much preferred. As the power waveform has a main energy concentration at around 60Hz, this frequency has
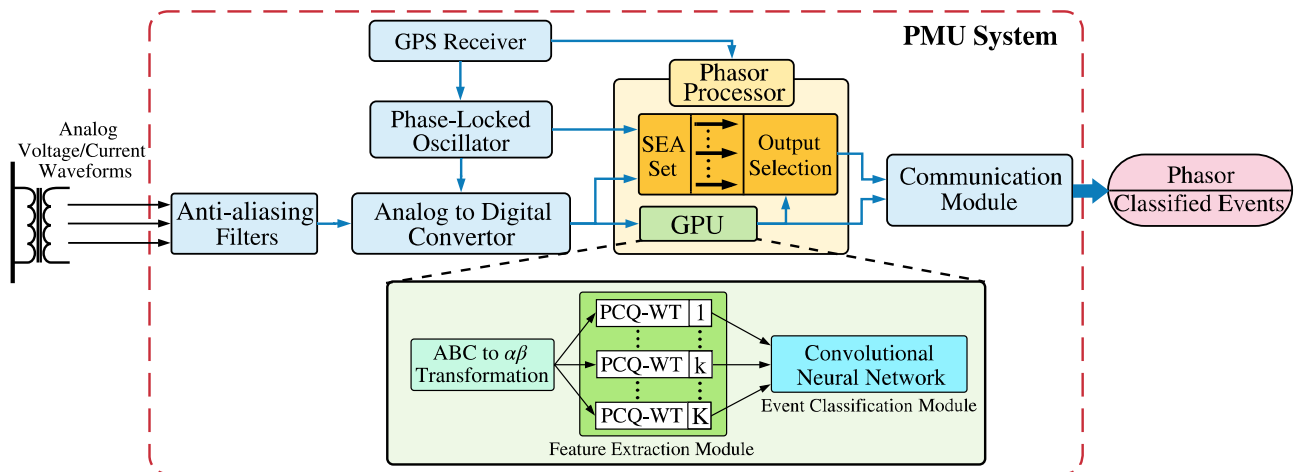
Fig. 3. Diagram of the proposed framework embedded within PMU.

TABLE I
PMU INPUT TEST WAVEFORM PARAMETER SPECIFICATION

Test Name	Input Range	Test Name	Input Range
Signal to Noise Ratio (SNR)*	40dB	Frequency Jump	-5Hz to 5Hz
Magnitude Jump	0.1-2pu	Phase Jump	$\pm\pi/18$ radians
Harmonic Distortion	0.5%-10%THD; order up to 50th	Out-of-Band Interference	10Hz to 120Hz; level 0.01-0.1pu
Amplitude Modulation	0.1Hz to 5Hz; level 0.005-0.1pu	Angle Modulation	0.1Hz to 5Hz; level 0.005-0.1pu
Frequency Ramp	±0.01Hz/s to ±1Hz/s, within ±5Hz	Single-line-to-ground (SLG) fault	Magnitude drop 0.2-1pu
Line-to-line (LL) fault	Magnitude drop 0.1-1pu**	Line-to-line-to-ground (LLG) fault	Magnitude drop 0.1-1pu

*Occurs in all test signals
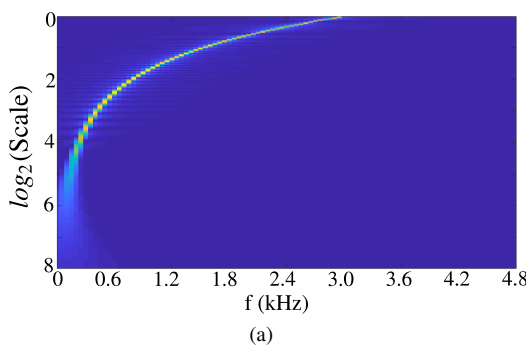**Phase shift occurs at lines with faults

(a)

Fig. 4. Spectrum of the proposed wavelet bank, the scaling factor is plotted by log_2.

gained the most attention during monitoring. So, we chose to compress the high frequency range and neglect the low frequency portion; a dyadic scaling factor range [1, 256] is chosen. Finally, we set the value of the constant $\gamma = 2$ and chose a fixed window size with 20ms (192 time bins) associated with a fixed time shift $b_k = 100$. By setting these parameters, the PCQ-WT used in our experiments is obtained according to (19). The corresponding spectrum of the Gabor wavelet bank is shown in Fig. 4. The final scalogram fed into the CNN has a duration of 40ms (385 time bins) including one historical window.

The test power waveforms' parameters specification for CNN training are selected according to [19]. All test wave-

forms are polluted by Gaussian background noise with a signal to noise ratio of 40dB. With the eleven types of test waveforms plus a normal waveform, a total twelve types of waveforms are simulated. Each type of waveform has been generated randomly according to the parameter specifications in Table. I. The input parameters are uniformly located in the designated ranges. The occurrence time of events is randomly located within the 20ms simulation run-time window.

B. Proposed CNN Configuration

As can be seen in Fig. 2, the CNN for the scalogram classification has the following architecture: **Input(256×385)– Conv(100, 5×11)–Max-pool(3×3)–Conv(100, 5×5)–Max-pool(3×3)–Conv(64, 5×5)–FC(600)–FC(12)**. Unlike the ordinary images which have homogeneous units on the two axes, the axes of scalograms are with different units. We chose a wide-shape kernel in the first convolutional layer that could extract more information for transitions of the scalogram along the time axis, and the stride of the convolution operation in the first layer is (2, 3)—other convolutional layers' strides are (1, 1). Besides the last FC layer, batch normalization [50] is used in each Conv and FC layers. Dropout [51] was adopted in the third convolutional layer and the first FC layer to prevent over-fitting. Rectified Linear Unit (ReLU) were chosen as nonlinearities in the neural net. The CNN used cross-entropy as the loss function.
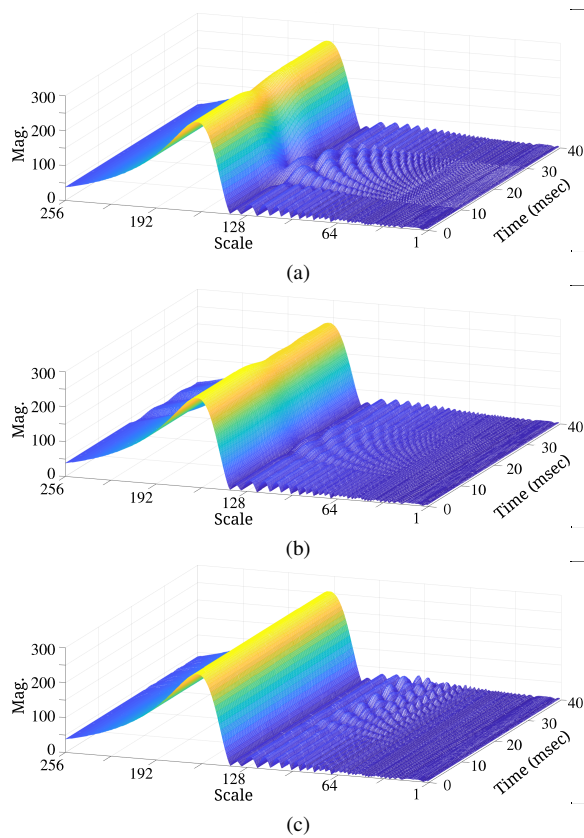
Fig. 5. Test waveform simulation results: (a) a 2Hz frequency jump occurs at $t = 10$ ms, (b) a high resistance line-to-line fault happens at $t = 10$ ms, (c) harmonic waves with orders of 5, 7, 9 and 11 are injected at $t = 10$ ms.

Fig. 6. The confusion matrix for the best validation results.

We experimented 120,000 samples of the wavelet scalogram for twelve types of events are simulated in the Matlab environment, wherein 96,000 samples are used as the training dataset, 12,000 samples for validation, and 12,000 for testing. We used Adam [52] as the optimizer, which has initialized learning rate of 1×10^{-3}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The CNN was trained 120 epochs, and the learning rate was decayed 1/10 in every 30 epochs. The best validated model was recorded and tested.

C. Experimental Results and Analysis

Three patterns extracted by PCQ-WT from three selected events are demonstrated in Fig. 5. One can see, from Figure 5(a), that the magnitude of the highest energy concentration which stands for fundamental frequency in the scalogram remains almost constant except at $t = 15$ ms which is 5 ms later after a frequency jump occurs, a small drop of of the peak can be observed. Among the high frequency range—scaling factor from 1 to 128—the pattern appears almost immediately as the event happens. Figure 5(b) shows a pattern caused by a high resistance line-to-line fault. This pattern has different features comparing to Fig. 5(a), and the highest energy concentration part has a significant drop after the fault happens. Similarly, this drop is around 5ms after the fault; the features in the high frequency range also show up immediately. When harmonics exist, on can see that the features in the high frequency range are still unique in Fig. 5(c), and the fundamental frequency feature remains unchanged. Similar to

the events in Fig. 5(a) and (b), the harmonic injection can be detected almost immediately.

We trained the CNN that classifies the scalograms five times. The best test results are summarized in the confusion matrix (Fig. 6), and the best test classification accuracy is 92.84±1.20% (mean±std). As a time-critical infrastructure, a PMU (for 60Hz system) usually has to process the sampled signal and report the phasor measurement results at the rate of 60 frames/sec [8]. The computational time of the PCQ-WT and CNN should be investigated. We used a workstation with an Intel Core i7-8700K CPU and Nvidia GeForce GTX 1080Ti GPU as the computational platform; Matlab 2016b and Pytorch 0.4.1 [53] as the implementation tools for PCQ-WT and CNN, respectively. The 12,000 test samples were adopted for testing the *computational* time of the proposed CNN. As a result, generating a 256×385 scalogram by PCQ-WT takes 1.20±0.23 ms, and the CNN takes 1.04±0.31 ms to process every scalogram. Therefore, the in-total computational time of our proposed framework is 1.24 ± 0.54. In this timing experiment, the timer starts from when the data were loaded into the CPU and GPU memory and ends when the classification results were given out. The test uses 32-bit precision. This short computational time satisfies the real-time constraints with large margin and leaves plenty of time to the subsequent SEA analytics to function.

From Fig. 6, it can be seen that for most test waveforms, the proposed framework detects the event accurately. However, the frequency jump test case has the lowest accuracy; the reason is that the range of frequency jump in simulations is assumed to be within [-5, 5] Hz. Thus, the values of frequency jump could be in the vicinity of zero; in such circumstances, the features in the scalogram is too weak to be detected by the CNN, and these instances can be featured a "normal" signature as the frequency jump step is extremely small.

The classification accuracy of amplitude and angle modulation are not ideal by still acceptable. The explanations for the miss-classifications on these two categories could be twofold: (i) The physical duration in a scalogram proved to CNN is 40 ms, which is too short to capture the entire transition patterns; (ii) The lower the modulation frequencies are, the smaller the modulation magnitudes appear. The above explanations are mathematically explained in the following:

$$x_{\alpha\beta}(t) = (1 + \underbrace{M_\Delta(t - t_\Delta)\sin(\omega_\Delta(t - t_\Delta)t)}_{\text{Modulation}}) \quad (21)$$
$$\cdot \cos\left(2\pi \int_0^t F(\tau)d\tau + \phi(t)\right)$$

$$M_\Delta(t) = \begin{cases} C_M & t > 0, \quad C_M \in [0.005, 0.1] \\ 0 & t < 0 \end{cases}$$
$$\omega_\Delta(t) = \begin{cases} C_A & t > 0, \quad C_A/2\pi \in [0.1, 5] \\ 0 & t < 0 \end{cases} \quad (22)$$

where $M_\Delta(t)$ and $\omega_\Delta(t)$ are the modulation magnitude and frequency, respectively; and t_Δ is the event occurrence point. As the duration is 40 ms, i.e., $t_{end} = 40$ ms, and t_Δ should satisfy $0 < t_\Delta < 40$ so that the event can be shown in the scalogram, thus $t_{end} - t_\Delta$ turns into a very small value. With small C_M and C_A, the modulation can be written as

$$Mod(t) = C_M \sin(2\pi C_A(t - t_\Delta))$$
$$\approx 2\pi C_M C_A(t - t_\Delta) \quad (23)$$
$$\approx 0, \quad t \in [t_\Delta, t_{end}]$$

From (23), one can see that if the event occurs at a time that is very close to t_{end}, the modulation $Mod(t)$ is extremely small under the condition that C_M and C_A are small; meanwhile, the duration of modulation in the scalogram is very short. Thus, a scalogram with 40 ms duration would not contain sufficient features for the occurrence of amplitude modulation, and the classification accuracy for such events are relatively lower than the others. Similar arguments hold for analyzing the results under angle modulation scenarios.

Thus, increasing the window size and the modulation strength may be beneficial to the classification accuracy, but the overall performance, especially for the real-time capacity of the PMU measurement, is harmed. A trade-off between the awareness and the accuracy transients exists here. As the proposed scheme aims to be deployed as a fast online grid monitoring tool and an awareness mechanism to assist the PMUs selecting an appropriate SEAs, we kept the parameters that are specified in Subsection IV-A.

V. CONCLUSION

This paper introduced novel PMU-embedded analytics for power grid online surveillance, that consists of the PCQ-WT feature extraction and CNN-based event classification mechanisms. The proposed framework aims to effectively extract the waveform features and efficiently classify multiple types of events in the grid. Our experiments demonstrated that the proposed analytics achieved high accuracy for real-time event classification. This framework would be a foundation for intelligent PMUs that could adaptively select an appropriate

SEA and achieve higher phasor measurement accuracy. Future work will be focused on integration of SEAs in the proposed framework to leverage the PMU overall functionality.

REFERENCES

[1] J. A. de la O Serna, "Synchrophasor measurement with polynomial phase-locked-loop taylor–fourier filters," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 328–337, Feb 2015.

[2] M. Bertocco, G. Frigo, C. Narduzzi, C. Muscas, and P. A. Pegoraro, "Compressive sensing of a taylor-fourier multifrequency model for synchrophasor estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3274–3283, Dec 2015.

[3] D. Petri, D. Fontanelli, and D. Macii, "A frequency-domain algorithm for dynamic synchrophasor and frequency estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 10, pp. 2330–2340, Oct 2014.

[4] T. Becejac, P. Dehghanian, and M. Kezunovic, "Analysis of pmu algorithm errors during fault transients and out-of-step disturbances," in *IEEE PES Transmission & Distribution Conference and Exposition-Latin America*, 2016, pp. 1–6.

[5] N. H. Abbasy and H. M. Ismail, "A unified approach for the optimal pmu location for power system state estimation," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 806–813, May 2009.

[6] S. Chakrabarti, E. Kyriakides, and D. G. Eliades, "Placement of synchronized measurements for power system observability," *IEEE Transactions on Power Delivery*, vol. 24, no. 1, pp. 12–19, Jan 2009.

[7] S. Azizi, G. B. Gharehpetian, and A. S. Dobakhshari, "Optimal integration of phasor measurement units in power systems considering conventional measurements," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 1113–1121, June 2013.

[8] "Ieee standard for synchrophasor measurements for power systems," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, Dec 2011.

[9] A. Z. Amanci and F. P. Dawson, "Synchronization system with zero-crossing peak detection algorithm for power system applications," in *Power Electronics Conference (IPEC), 2010 International*. IEEE, 2010, pp. 2984–2991.

[10] S. Das and T. Sidhu, "Robust algorithm to estimate fault synchrophasor from fault-transient synchrophasor in phasor data concentrator," *IET Generation, Transmission Distribution*, vol. 9, no. 2, pp. 124–132, 2015.

[11] C. Thilakarathne, L. Meegahapola, and N. Fernando, "Static performance comparison of prominent synchrophasor algorithms," in *2017 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, Dec 2017, pp. 1–6.

[12] S. Das and T. Sidhu, "A new algorithm to compute fault synchrophasor from transient state synchrophasor in pdc," in *2013 IEEE Power Energy Society General Meeting*, July 2013, pp. 1–5.

[13] Á. Ortega and F. Milano, "Comparison of different pll implementations for frequency estimation and control," in *Harmonics and Quality of Power (ICHQP), 2018 18th International Conference on*. IEEE, 2018, pp. 1–6.

[14] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1474–1483, Aug 2011.

[15] P. K. Dash, K. R. Krishnanand, and M. Padhee, "Fast recursive gauss-newton adaptive filter for the estimation of power system frequency and harmonics in a noisy environment," *IET Generation, Transmission Distribution*, vol. 5, no. 12, pp. 1277–1289, December 2011.

[16] A. J. Roscoe, I. F. Abdulhadi, and G. M. Burt, "P and m class phasor measurement unit algorithms using adaptive cascaded filters," *IEEE Transactions on Power Delivery*, vol. 28, no. 3, pp. 1447–1459, July 2013.

[17] I. Kamwa, S. R. Samantaray, and G. Joos, "Wide frequency range adaptive phasor and frequency pmu algorithms," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 569–579, March 2014.

[18] P. Castello, J. Liu, C. Muscas, P. A. Pegoraro, F. Ponci, and A. Monti, "A fast and accurate pmu algorithm for p+m class measurement of synchrophasor and frequency," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 12, pp. 2837–2845, Dec 2014.

[19] T. Becejac, P. Dehghanian, and M. Kezunovic, "Probabilistic assessment of pmu integrity for planning of periodic maintenance and testing," in *IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2016, pp. 1–6.

[20] ——, "Impact of the errors in the pmu response on synchrophasor-based fault location algorithms," in *2016 North American Power Symposium (NAPS)*, 2016, pp. 1–6.

[21] C. Qian and M. Kezunovic, "A power waveform classification method for adaptive synchrophasor estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 7, pp. 1646–1658, July 2018.

[22] F. A. S. Borges, R. A. S. Fernandes, I. N. Silva, and C. B. S. Silva, "Feature extraction and power quality disturbances classification using smart meters signals," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 824–833, April 2016.

[23] M. S. Manikandan, S. R. Samantaray, and I. Kamwa, "Detection and classification of power quality disturbances using sparse signal decomposition on hybrid dictionaries," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 27–38, Jan 2015.

[24] B. Tang, T. Song, F. Li, and L. Deng, "Fault diagnosis for a wind turbine transmission system based on manifold learning and shannon wavelet support vector machine," *Renewable Energy*, vol. 62, pp. 1–9, 2014.

[25] R. H. Park, "Two-reaction theory of synchronous machines generalized method of analysis-part i," *Transactions of the American Institute of Electrical Engineers*, vol. 48, no. 3, pp. 716–727, July 1929.

[26] W. Duesterhoeft, M. W. Schulz, and E. Clarke, "Determination of instantaneous currents and voltages by means of alpha, beta, and zero components," *Transactions of the American Institute of Electrical Engineers*, vol. 2, no. 70, pp. 1248–1255, 1951.

[27] J. A. de la O Serna and J. Rodríguez-Maldonado, "Taylor–kalman–fourier filters for instantaneous oscillating phasor and harmonic estimates," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 4, pp. 941–951, April 2012.

[28] M. Bertocco, G. Frigo, C. Narduzzi, C. Muscas, and P. A. Pegoraro, "Compressive sensing of a taylor-fourier multifrequency model for synchrophasor estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3274–3283, Dec 2015.

[29] D. P. Mishra, S. R. Samantaray, and G. Joos, "A combined wavelet and data-mining based intelligent protection scheme for microgrid," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2295–2304, Sept 2016.

[30] K. Thirumala, M. S. Prasad, T. Jain, and A. C. Umarikar, "Tunable-q wavelet transform and dual multiclass svm for online automatic detection of power quality disturbances," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3018–3028, July 2018.

[31] P. Rajaraman, N. A. Sundaravaradan, R. Meyur, M. J. B. Reddy, and D. K. Mohanta, "Fault classification in transmission lines using wavelet multiresolution analysis," *IEEE Potentials*, vol. 35, no. 1, pp. 38–44, Jan 2016.

[32] Y.-C. Su, K.-L. Lian, and H.-H. Chang, "Feature selection of non-intrusive load monitoring system using stft and wavelet transform," in *e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 293–298.

[33] S.-H. Ni, K.-F. Lo, L. Lehmann, and Y.-H. Huang, "Time–frequency analyses of pile-integrity testing using wavelet transform," *Computers and Geotechnics*, vol. 35, no. 4, pp. 600–607, 2008.

[34] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bulletin of the American Meteorological society*, vol. 79, no. 1, pp. 61–78, 1998.

[35] P. S. Addison, *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*. CRC press, 2017.

[36] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 617–643, March 1992.

[37] M. Stephane, *A wavelet tour of signal processing*. Elsevier, 1999.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[42] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[43] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[44] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[45] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[46] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, April 1998, pp. 454–459.

[47] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, Aug 1996.

[48] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, "Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification," *IEEE Transactions on Medical Imaging*, vol. 25, no. 9, pp. 1214–1222, Sept 2006.

[49] J. Hubbard, "Calculation of partition functions," *Physical Review Letters*, vol. 3, no. 2, p. 77, 1959.

[50] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[51] 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.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[53] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.