Power System Disturbance Classification with Online Event-Driven Neuromorphic Computing

Kaveri Mahapatra, *Student Member, IEEE,* Sen Lu, Abhronil Sengupta, *Member, IEEE,*
and Nilanjan Ray Chaudhuri, *Senior Member, IEEE*

Abstract—Accurate online classification of disturbance events in a transmission network is an important part of wide-area monitoring. Although many conventional machine learning techniques are very successful in classifying events, they rely on extracting information from PMU data at control centers and processing them through CPU/GPUs, which are highly inefficient in terms of energy consumption. To solve this challenge without compromising accuracy, this paper presents a novel methodology based on event-driven neuromorphic computing architecture for classification of power system disturbances. A Spiking Neural Network (SNN)-based computing framework is proposed, which exploits sparsity in disturbances and promotes local event-driven operation for unsupervised learning and inference from incoming data. Spatio-temporal information of PMU signals is first extracted and encoded into spike trains and classification is achieved with SNN-based supervised and unsupervised learning framework. In addition, benefits of deep spiking networks for complex multi-class event identification problem are presented by leveraging increasing dynamic neural sparse spiking events with network depth. Moreover, a QR decomposition-based selection technique is proposed to identify signals participating in the low rank subspace of multiple disturbance events. Performance of the proposed method is validated on data collected from a 16-machine, 5-area New England-New York system.

Index Terms—PMU, Spiking neural network (SNN), Event-driven operation, Signal selection, Disturbance classifier, Neuromorphic computing, MAC operations.

I. INTRODUCTION

ADVENT of Phasor Measurement Units (PMUs) is leading to a new era of wide-area monitoring with large-scale utilization of machine learning (ML) and deep learning (DL) techniques. Proliferation of PMUs offer better observability of the modes of the system and helps building situational awareness. However, this also brings along challenges in handling and utilization of higher volume of time-series data. Typically, PMU measurements are transmitted at high speeds to digital computing systems (e.g. DSP, microprocessor or computers with CPU/GPU capabilities) in the control center, which requires high computational power and consumes lots of energy while processing the data for monitoring applications. One such application is the classification of large-scale events using PMU data.

The conventional way to approach event classification problem involves two steps. The first step is to utilize many measurements and apply feature engineering to extract relevant features for events under consideration. Some examples are raw time-series, energy function approach [1]–[3], ellipsoid characteristics [4]–[7], frequency-domain details such as wavelets or shapelets [1], [7]–[11], principal component analysis [4], [12]–[14]. Extraction of features involve computations for every

new data window. In contrast, we propose a feature selection method to utilize a subset of raw signals directly into the spiking network framework so that the computations associated with feature extraction can be avoided.

The second step is the utilization of ML/DL based classification algorithms. Classification algorithms proposed in literature are broadly based on either statistical tests, correlation tests or thresholds [2], [9], [13], [15], [16], or different ML methods such as clustering [5], [17], decision tree [18]–[20], support vector machines [11], [14], [20]–[22], k-Nearest neighbor [8], [21], [23], artificial neural network (ANN) [3], [20], extreme learning machines [23], auto-encoders [14], and deep learning architectures [24] among others. These techniques inherently require a large computational effort in processing every incoming time-series data window for training and inference on a traditional computing platform. Especially, computations performed on an ANN platform for learning and inference is extremely hardware expensive and cannot leverage the sparse nature of disturbance events in PMU measurements.

In contrast, in the human brain, information is processed in networks composed of spiking neurons, which accepts dynamic binary spiking inputs as a function of time and with only parts of the network being active during any operation. A Spiking Neural Network (SNN) driven by the notion of biological neurons with computing by means of sparse binary spike signals over time is thus more biologically plausible, and supports unsupervised learning via event-driven hardware operation, consuming much less power. These features make SNN a suitable candidate for online learning and inference from PMU data for disturbance monitoring and thereby, necessitates development of SNN architecture for enabling algorithm-hardware co-design and encourages implementation on hardware for low power event-driven operation. In this work, we present a maiden application of neuromorphic computing platform in power systems for an event classification problem using PMU data streams. A SNN architecture is proposed to extract spatio-temporal information from PMU measurements in an efficient way for classification of different power system events without sacrificing accuracy obtained with conventional ML algorithms.

This paper is organised into six sections. Section II discusses the main motivation for shifting from ANN to a spiking domain for representing PMU data. Section III presents spike-based representation of a time series signal. Section IV introduces a feature selection approach to complement SNN classification framework to reduce the computational burden by enabling selection of candidate raw signals. Section V discusses different

types of SNN learning frameworks for event classification problem. Section VI presents the experimental set up and discusses the results of performance comparison between ANN and SNN. Section VII presents results with deeper spiking neural networks on event classification problem. Section VIII concludes the paper.

II. MOTIVATION

Disturbance event identification and characterization in smart grid using PMU data is a computationally expensive task for online operations. Availability of ML/DL resources on CPU/GPU at control centers helps in solving problems requiring training and inference using large PMU data sets. However, due to lack of event-driven operation and inability to exploit sparsity patterns in the data, the computational burden, energy consumption, and time required to train and deploy networks for this complex solution space is humongous. Addressing the high computational burden of machine learning hardware has been identified as one of the Grand Challenges in Computing for the next decade by the US government and Federal agencies [25], [26]. For instance, training common large Deep Learning models can result in more than 626,000 pounds of carbon emissions - the equivalent of 5 times the lifetime emission of an average car [27]. Therefore, power grid's secure operation also comes at a cost of significant increase in its carbon footprint, which counters the sustainability goals in today's world. This just shows the inefficiency of our current computational platforms (CPU/GPUs/cloud clusters) and how these can be improved with neuromorphic computing. The goal of this paper is to address this challenge for the smart grid community.

An under-explored fact here is that a power transmission system remains in ambient/quasi steady state operating condition most of the time and switches to transient state only during disturbance events for a brief period. Typical disturbance events such as line faults (voltage events), generator-load unbalancing (frequency events) due to generator outages or load shedding, and so on, occur infrequently over the span of the year. Table I shows two metrics from the North American Electric Reliability Corporation (NERC) report [28] for years 2015-2018, which indicate the average probability of occurrence of transmission line outage events in a year is $\approx 0.237\%$ and also the number of occurrences of contingency events related to balancing resources/generation with demand following a disturbance ranges between $150 - 425$ per year.

TABLE I
FINDINGS FROM NERC STATE OF RELIABILITY REPORT [28]

Years	2015	2016	2017	2018
Transmission Unavailability (%)	0.22	0.27	0.24	0.22
Number of Disturbance Control Standard Events (Range)	350--400	375--425	150--200	175--225

In addition, based on a 15s window of simulated data from the New England-New York test system [29], a typical histogram of distribution of signal magnitudes for 363 events is shown in Fig. 1(a). Signal magnitude here refers to the deviation from nominal values per unit magnitudes of phasor voltages ($|V|$), currents ($|I|$) or frequency ($f/60$) of various network variables in the system following a disturbance. Each of these windows contains 5s of pre-disturbance and 10s

of post-disturbance data from each event. For more details on the data generation, please refer to Section VI-A. The characteristic of these deviations alters with post-disturbance operating condition. Under quasi-steady-state pre-disturbance (also called ambient) condition, the signal magnitudes are represented by zero mean gaussian noise. During disturbance, the fluctuations in these magnitudes are highest and they typically manifest electromechanical oscillations of smaller amplitude following the disturbance , which damp out if the system is stable. Therefore, the period of high magnitude fluctuations (0.95-1 times signal magnitude of 1 pu) from steady state value (of.1 p.u.) for most disturbances is short and thus such fluctuation values have a less probability of occurrence in the window right after disturbance.

This implies there is a higher probability of samples in the signals of the window to have a lower range of 'magnitude deviation from the prefault values' that lies on the left side of histogram. Note that these magnitude deviations are caused by oscillations. Due to temporal sparsity of events, most of the oscillations occur only at the starting of a disturbance event and dies down after a small duration. In addition, due to spatial sparsity of PMUs, the signals which are closer to the disturbance location undergoes more significant deviations during disturbance compared to others which are farther (see 4.2.2 of [30]). Moreover, spatial sparsity brings in dispersion in the observability of modes reflected in post-disturbance oscillation magnitudes. As a result, in the event window, the high magnitude samples are less frequent and can be characterized by a sparse distribution with close to 0 magnitude samples persisting mostly in the time horizon.

Interestingly, our brain computes by means of sparse, event-driven temporal signals and algorithms inspired by such computational principles can be an ideal fit here. Inspired by this property, we propose to leverage a disturbance event-driven neuromorphic computing platform, which can exploit sparsity in signal representation in spike domain and thus reduces the computational effort leading to energy savings.

SNN is a class of ANN-based computational model, which closely mimics our brain's ability to naturally encode and process information. Its efficiency comes from the "learning over time" feature as well as computation and communication in terms of spike in the operating model. The main advantages of SNN are as follows: (1) sparse signal processing, (2) event-driven hardware operation, and (3) lower energy consumption [31]. Recent developments show that SNN now has the potential to achieve similar accuracy as of other ML techniques with much less power consumption [32]. Incoming data represented by a sparsely-distributed input spike train can significantly reduce energy consumption during periods of ambient condition of the grid.

Figure 1(b) describes the proposed architecture for processing PMU signals at a control center in spiking domain. After receiving PMU data at phasor data concentrator (PDC), a moving window containing present and past samples of data is extracted. This data is first encoded into spike trains and then processed in SNNs for event detection and classification purposes. The main contributions of this work are as follows.

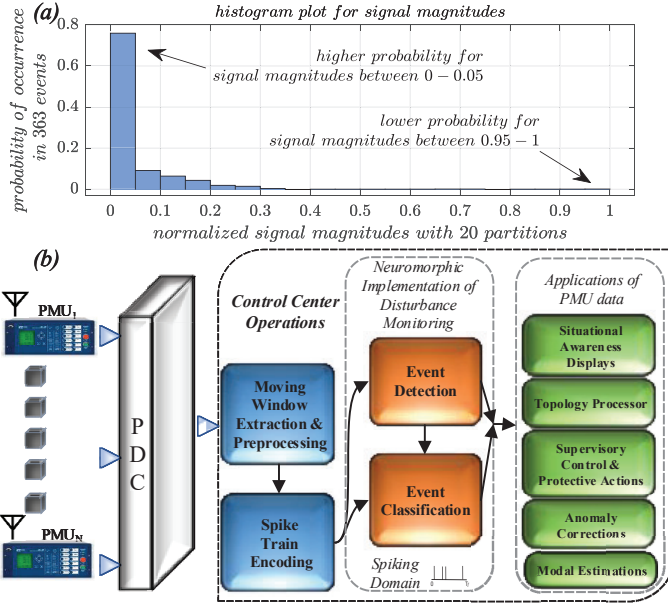- A spike-based representation of PMU measurements and

Fig. 1. (a) A histogram plot for a 15s measurement data window of 10 signals with disturbance starting at 5s over 363 disturbance events, where X-axis indicates 20 normalized magnitude ranges of magnitude deviations from ambient prefault values and Y-axis indicates the probability of any signal taking a magnitude in that range at any instant. The events are simulated in NE-NY system; (b) Proposed architecture for power system disturbance classification.

event-driven supervised and unsupervised power system event classification through a SNN architecture that can replace the CPU/GPU-based processing of state-of-the-art classification algorithms.

- A signal selection technique to eliminate dependency on processing many signals without loosing any information and thus, reduce the computations during classification by operating directly on measurements instead of derived features, when the number of observed signals is large.

The following sections present these contributions.

III. Spike-Train Representation of a Time-Series Signal

A spike train is a mode of transmitting information in the nervous system. This can be interpreted as a sequence of recorded times at which a neuron fires an action potential [33]. The irregularity in arrival time of successive action potentials can be modelled as a random process. Under an independent spike generation hypothesis, the random process in the neocortex is assumed to follow a Poisson distribution. This is typically utilized in a process called "rate coding" [34] to convert an input sample to a certain number of spikes while including an element of stochasticity.

The ith sample $y(i)$ of a discrete time-series measurement $[\bar{y}], i \in [1, N]$ is represented by first associating it with an input neuron i. With a homogeneous Poisson process assumption, the spike generation rate λ_i of this neuron over an interval of spiking takes a fixed continuous analog value and is set to be proportional to the amplitude of the sample $y(i)$. In other words, the neuron i is driven by a stimulus of $y(i)$ over the interval $[0, \mathtt{T}]$ to generate spikes at a rate $\lambda_i = y(i)$. The spike train interval $\mathtt{T}$ is subdivided into shorter intervals $\delta\mathtt{t}$ such that $\mathtt{t} = j\delta\mathtt{t}$, where $j = 0, 1, 2, ..., \mathtt{T}/\delta\mathtt{t}$. The probability of spikes occurring during each $\delta\mathtt{t}$ is set to $r\delta\mathtt{t}$. For each $\delta\mathtt{t}$, a

uniformly distributed random number $\mathtt{x}(\delta\mathtt{t})$ is generated. This is followed by a spike generation as per the following rule.

$$\mathtt{s}(j\delta\mathtt{t}) = \begin{cases} 1, & \mathtt{x}(j\delta\mathtt{t}) \leq r\delta\mathtt{t} \\ 0, & otherwise \end{cases} \Rightarrow \mathtt{s} = \sum_{\mathtt{T}_f} \delta(\mathtt{t} - \mathtt{t}_f)$$
(1)

where, $\mathtt{s}$ is the spike train generated over the interval $[0, \mathtt{T}]$ with spiking instances $\mathtt{t}_f$ for sample $y(i)$. $\mathtt{s}$ can also be represented as a set of Dirac-δ functions at time instants $\mathtt{t}_f \in \mathtt{T}_f$. These spike trains are given as input to the excitatory layer of SNN and has been demonstrated in Fig. 2. This type of binary encoding of stimulus requires a single bit for computing as well as for communicating a spike. With address-event representation (AER)-based communication protocol, sparse neural events can be communicated efficiently in event-driven neuromorphic hardware [31].

A power system follows a quasi-steady state trajectory during most of the time of operation and only switches to a transient state trajectory during a disturbance. The spike generation rate follows a steady-state value proportional to signal magnitude during the steady state operation and fluctuates in presence of disturbance samples. Assuming that under steady state conditions of the system, the spike rate is set to a low value, the distribution in spike occurrence will be sparse over the interval of stimulus presentation to the neurons. This leads to inactivity in the regions of SNN, which can be exploited on a neuromorphic hardware through "power gating" [35] by shutting off portions of the circuit, which are not being triggered by spiking events. Only upon the onset of an event, sufficient spikes are generated to activate the network enabling event-driven operation, and eliminating computational effort. Thus, a spiking representation of signals and the algorithm-hardware co-design of SNN and implementation on neuromorphic hardware could potentially result in significantly low power consumption.

IV. Signal Selection

PMUs provide discrete synchronous time-series measurements of electrical quantities $y(t)$ such as voltage, current, power and frequency at any time t. The system response Y ($N \times m$) containing N samples for $t = [0, (N-1)Ts]$ of m measured signals can be approximated by low dimensional feature set could potentially overcome the dependency on processing all PMU signals online. However, it comes at a cost of SVD computations every time a new sample arrives, and thus imposes a complexity of $\mathcal{O}(mN)$ for online feature extraction from a $N \times m$ data matrix. To overcome this, a feature selection procedure based on economy QR decomposition [36] is proposed to identify PMU signals which participate in forming the low-rank subspace under multiple disturbance events that could be used as input to the SNN.

To that end, response matrix Y can be expressed as a product of Vandermonde matrix V and complex transformation matrix F_o. Please see Appendix-A for details.

$$Y^T = F_o V \Rightarrow Y = V^* F_o{}^*$$
(2)

The rectangular matrix Y can be decomposed into two factors using economy QR decomposition [36], which rearranges signals present in Y in decreasing order of priority for obtaining a linearly independent set of signals in the columns of $\overleftarrow{Y}$.

$$YP = \overleftarrow{Y} = QR \Rightarrow Q^T \overleftarrow{Y} = R \qquad (3)$$

where, Q represents an orthonormal basis for subspace of $\overleftarrow{Y}$ and R represents an upper triangular matrix $r_{k,l} = \bar{q}_k^T \bar{y}_l$, P represents permutation matrix. When all the signals present in Y are linearly independent, the rank of subspace U of Y is m implying at least m distinct modes (v_i) of V are contributing in forming U. Since, Y is low rank, the rank of U becomes $n < m << N$. Since $Q \subseteq \mathcal{R}(Y)$, where $\mathcal{R}$ indicates the column space, it can be written as a mapping from V^* as follows. We know that columns of Q form the orthonormal basis of the subspace U in $\mathbb{R}^N$ and represents a similar transformation of trajectories in V, thus can be written as follows.

$$Q = V^* G_o^* \Rightarrow Q^T = G_o V \qquad (4)$$

where, G_o in (4) can be interpreted as a transformation matrix similar to F_o such that $\bar{q}(t) = \chi_{g_o}(\bar{v}(t))$ and $\bar{q}(t)$ is the image of $\bar{v}(t)$ under mapping χ_{g_o}.

$$Q^T Q = I = GVV^* G^* = G\hat{V}G^*$$
$$\Rightarrow G\hat{V} = (G^*)^\dagger = (G^\dagger)^* \qquad (5)$$

where, $G^\dagger$ represents the pseudoinverse of matrix G. Now substituting expression for $G^{\dagger*}$ in (3) for R leads to

$$R = Q^T \overleftarrow{Y} = GVV^* F^* = G\hat{V}F^* = (G^\dagger)^* F^* \qquad (6)$$

This implies R is the projection of F^* onto G^*, which are both mapping complex exponential time series in Vandermonde matrix to the signal subspace of Y. Each entry $r_{k,l}$ of R is equivalent to $\bar{g}_k^T \bar{f}_l$, which is the projected component of $\bar{f}_l$ in the direction of $\bar{g}_k$ of Q.

$$\bar{g}_k^T \bar{f}_l = 0, \forall l, k = l + 1 : m$$
$$\bar{g}_1^T \bar{f}_1 \geq \bar{g}_2^T \bar{f}_2 \geq ...\bar{g}_n^T \bar{f}_n \geq ... \geq \bar{g}_{m-1}^T \bar{f}_{m-1} \geq \bar{g}_m^T \bar{f}_m \qquad (7)$$
$$\bar{g}_k^T \bar{f}_k = 0, n < k \leq m$$

Note that the projected component in $r_{k,l}$ are proportional to the F^* matrix. A total of n independent column vectors of F^* corresponding to top n diagonal entries of R implies n linearly independent signals forming the subspace U of all m signals in Y. This set forms a minimal group, which at the least ensures observability of high energy modes responsible for oscillations observed through Y.

Since, the objective of this work is to detect transient disturbances with low frequencies of oscillation (persisting over longer duration) carrying more energy and the number of low frequency modes present in the system is $\tilde{n} < n$,–a very low dimensional subspace $\hat{Q}$ (tall matrix with $\tilde{n}$ orthonormal columns) need to be extracted from Y. In reality, both F^* and V^* are very sensitive to operating condition and disturbance as well. By applying the above analysis, we can obtain a linearly independent minimal group of $\tilde{n}$ signals for each disturbance scenario.

To select signals, a metric is proposed to find the importance of each signal among a group of signals based on its contribution in low rank subspace formation under multiple disturbance scenarios. Given a set of training examples of multiple disturbance events obtained by a contingency analysis on the system, we propose the following heuristic to identify a group of signals of priority and to represent the lower dimensional signal subspace.

- **Input** Disturbance event data set $Y_h \in \mathbb{R}^{N \times m}$, with $h \in [1 : H_{train}]$, where H_{train} is the number of training examples obtained under 3 types of disturbance events; Bus faults (Class-A), generator outage (Class-B), load trip (Class-C); **Output**: Important signal set S; **Parameters**: Threshold ν_1 for selecting important signals in each Y_h;
- **Initialization**
    - Set initial counter $CC(j)$ for each signal $j = 1 : m$.
    - Set thresholds $\nu_1 = 0.1$.
    - Set $h = 1$, set of all signals $M = [1 : m]$,
- **while** $h \leq H_{train}$
    1) Apply economy QR decomposition on Y_h to obtain permutation matrix P, orthonormal matrix Q, R.
    2) Find the permutation vector p from P containing arrangement of signals in Y used for economy QR decomposition.
    3) Find set $\hat{S} = \{s | s \in p, r_{s,s} \geq \nu_1 r_{1,1}\}$.
    4) Update counter $CC(j) = CC(j) + 1$ for all $j \in \hat{S}$.
    5) Increment h by 1 and go to step (1).
- Find importance score Sc for each signal j as follows. $Sc(j) = CC(j)/H_{train}$.
- Select the top n signals having highest scores in Sc and save those in set S.
- **Output** Set S containing signal indices important for multiple disturbance event data Y_h for $h = 1 : H_{train}$.

Given m number of signals for any event in Y_h, signals in $\hat{S}_h$ forming the low-rank subspace U_h is important for the event h. Cardinality ($\tilde{n}$) of set $\hat{S}_h$ is an estimate of the number of high energy modes $\lambda_i s$ present in m signals. Given H_{train} events, a score $Sc(j)$ for each of signal indicates how many such events, signal j has participated in forming their low-rank subspace. Thus, signals with higher scores become potential candidates for representing multiple disturbance events and thus, can faithfully ensure the information contained in signals not selected during this process is also preserved.

V. SPIKING NEURAL NETWORK

A. Learning using Unsupervised Spiking Neural Network

A typical architecture of spiking neural network [37] consists of input layer and excitatory layers as shown in Fig. 2. The number of neurons present in input layer is a function of number of signals and corresponding samples of the data window. Each neuron in the input layer is fully connected to the neurons in the excitatory layer. A connection between two neurons is called a synapse. After receiving the spike trains from input layer, excitatory postsynaptic potentials are generated. The synaptic weights between input to excitatory layer are then trained based on time difference between pre- and post-synaptic spike occurrences through a process called spike time dependent plasticity (STDP), which promotes temporal correlation in the firing activities of the connecting neurons. The excitatory layer has inhibitory connections in which information from each excitatory neuron is simultaneously backward-propagated to other neurons in the excitatory layer. This is called "lateral inhibition" whose purpose is to discourage simultaneous firing of multiple excitatory neurons and promotes
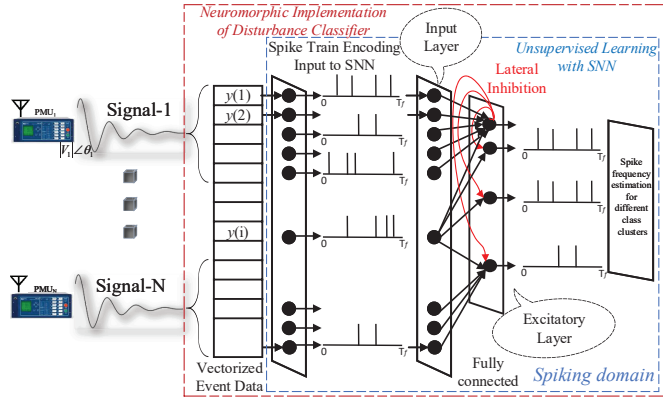
Fig. 2. Spike train representation of incoming PMU signals communicated to a Spiking Neural Network (SNN) architecture for disturbance event classification.
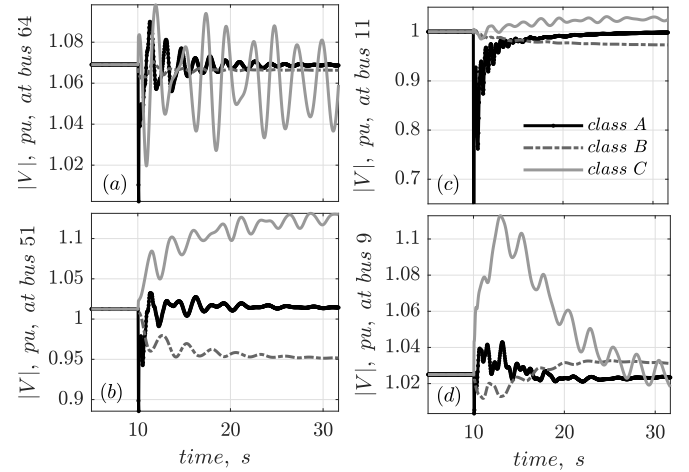

Fig. 3. Trajectories of 4 voltage measurements ($|V|, pu$) at buses 64, 51, 11, 9 during disturbance applied at 10s. Three traces show the type of disturbance; Class-A: bus fault, Class-B: generator outage, Class-C: load trip cases.

competition among neurons for learning different input patterns. The firing potential or threshold of each neuron in the excitatory layer is increased (referred to as 'Homeostasis') every time it spikes to prevent single neuron from dominating the firing pattern. Each excitatory neuron learns to represent a class and spikes at a higher frequency as compared to other neurons when examples of that class are presented as input to the SNN.

As shown in Fig. 2, the input layer accepts samples of measurement data window and produces spike train as output, which in turn is sent to the excitatory layer. The synapse response in terms of excitatory postsynaptic potential (EPSP) for a spike train input to a neuron in the excitatory layer can be described as follows.

$$EPSP(\mathtt{t}) = \begin{cases} 1, & \mathtt{t}_c \leq \mathtt{t} \leq \mathtt{t}_d, \mathtt{s}(\mathtt{t}_c) = 1 \\ 0 & otherwise \end{cases} \quad (8)$$

The temporal dynamics of membrane potential u_{post} of a leaky integrate-and-fire (LIF) neuron in the excitatory layer in response to EPSP is given by

$$\tau_{mt} \frac{du_{post}(\mathtt{t})}{d\mathtt{t}} = -u_{post}(\mathtt{t}) + w(\mathtt{t}) \sum_{\mathtt{T}_f} \text{EPSP}(\mathtt{t}) - I \quad (9)$$

where, τ_{epsp} and τ_{mt} are the EPSP and membrane time constants, respectively. u_{post} is being reset after reaching a threshold u_{th} and results in spiking of post-synaptic neuron in the excitatory layer. Immediately after this event, u_{post} is not allowed to change for a refractory period. In addition, Homeostasis was adopted to prevent single neuron from dominating the response pattern, where u_{th} for the spiking neuron is updated to a higher value at a rate of θ_r as follows.

$$u_{th} \Leftarrow u_{th} + \theta_r u_{th} \quad (10)$$

In this framework, a biologically plausible unsupervised learning mechanism based on STDP [37] is presented. STDP is a form of Hebbian learning, which supports event-driven learning and weight updating only upon arrival of post synaptic spikes during learning phase and can provide low power on-chip learning. The main idea behind STDP is the adjustment of synaptic weights based on the the temporal correlation between the pre- and post-synaptic spike occurrences. STDP-based weight updation for each connection between input to excitatory layer is described as follows.

$$\Delta w = \begin{cases} \varsigma_+ e^{(-\Delta \mathtt{t}/\tau_+)} & \Delta \mathtt{t} > 0 \\ \varsigma_- e^{(-\Delta \mathtt{t}/\tau_-)} & \Delta \mathtt{t} < 0 \end{cases} \quad (11)$$

where, $\Delta \mathtt{t} = \mathtt{t}_{post} - \mathtt{t}_{pre}$. $\mathtt{t}_{post}$ and $\mathtt{t}_{pre}$ are the pre- and

post-synaptic firing time instances, respectively. τ_+, τ_-, ς_+, ς_- are weight updating constants.

B. Learning using Supervised Spiking Neural Network

Although unsupervised SNN has the advantage of low power on-chip local learning without the requirement of any labelled data, supervised SNN gives much better performance and are currently more scalable to complex pattern recognition tasks [32]. Here, we explore a particular type of supervised SNN training, namely ANN-SNN conversion, for power system disturbance classification. In a supervised learning framework, a particular type of sparsely-firing event-driven spiking deep network [32], [38], [39] can be constructed by training an ANN using standard backpropagation and then converting to SNN. This technique exploits the relationship between number of spikes produced by integrate-and-fire (IF) spiking neuron over an interval in SNN and the output activation of Rectified Linear Unit (ReLU) network (ANN). Typically, the output activation v_j of a ReLU neuron j as a weighted sum of neuronal activations received from previous layer ($\forall y_i$) can be expressed as follows.

$$v_j = \max\left(0, \sum_i w_{ij} y_i\right) \quad (12)$$

Similarly, in case of IF neurons (without leak and refractory period), input y_i encoded as spike train $s^{y_i}(\mathtt{t})$, with expectation $E\left[s^{y_i}(\mathtt{t})\right] \propto y_i$, which is integrated over $\mathtt{T}$ to update membrane potential u_{mem}, which upon crossing the threshold u_{th} generates output spikes s^{v_j}.

$$u_{mem}(\mathtt{t}+1) = u_{mem}(\mathtt{t}) + \sum_i w_{ij} s^{y_i}(\mathtt{t}) \quad (13)$$

In this framework, a ReLU-Based Feed-Forward Neural Network is first trained. To ensure minimum loss in accuracy and faster convergence during the conversion process, a weight normalization is adopted [32], [38], [39] to optimize the ratio of neuron thresholds to synaptic weights. This is achieved by propagating the training set through the network and recording the maximum ReLU activations per layer. The weights are then scaled according to the maximum possible activation within the training set [38]. The final weight-normalized spiking network can be tested for classification of disturbances in the test set. For an extensive discussion on spiking neuron models and network architectures, readers are directed to reference [40].
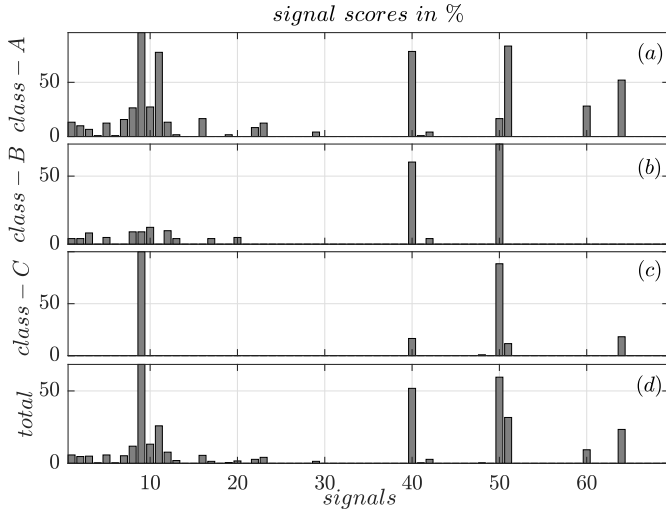
Fig. 4. For 69 voltage magnitude signals $|V|$, signal scores in % of total number of events was obtained over 3 classes of disturbance events and is presented in (a)-(c). The total score Sc as a sum of the 3 scores was calculated for each signal and is shown in (d).
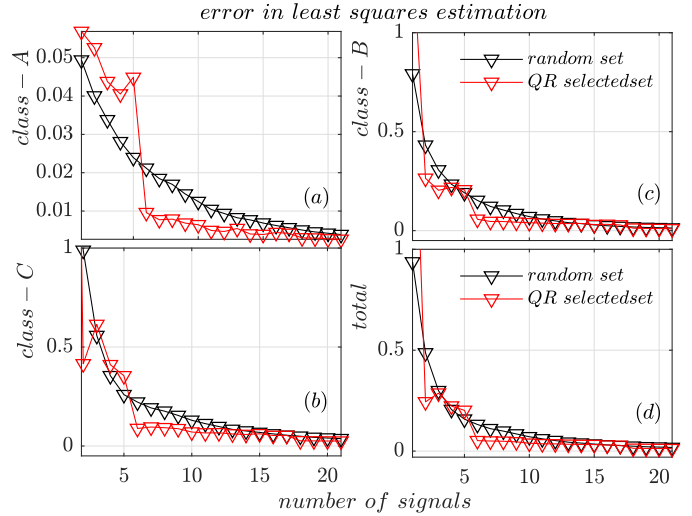

Fig. 5. Average error in LS estimation of signals in the set $\bar{S}_{W_r}(\bar{S})$ using the selected signal set $S_{W_r}(S)$, when the number of signals ($|S_{W_r}|(|S|)$) varies from 1-21 as shown in x-axis.

VI. RESULTS AND PERFORMANCE EVALUATION

A. Experimental Setup

A 16-machine, 5-area New England-New York system [29] is considered with PMUs installed at all 69 buses. A disturbance event data set consisting of (1) bus faults (BF), (2) generator outages (GO), and (3) load trip (LT) cases were created via a contingency analysis under 3 operating conditions with different loading scenarios. During each simulation, disturbance was applied at 10s and measurements of voltage magnitudes $|V|$ from all buses and frequencies f from 16 generator terminals were recorded for 40s at a sampling frequency of 30Hz. A total of 201 bus faults, 38 generator outages, and 156 load trip cases were generated. Figure 3 shows time-domain plots of 4 voltage magnitude measurement $|V|$ signals recorded under 3 types of disturbance cases.

■ *Pre-processing of voltage magnitudes and frequency measurements:* For classification purposes, the entire data set from the three classes were divided into 10-fold cross validation sets with 60% of the event examples used for training and 40% for testing. A window size of 15s was considered for extraction from starting of event at 10s while detrending and downsampling to 10 Hz was applied on each signal.

B. Signal Selection on Voltage Magnitude Measurements

The QR decomposition-based signal selection algorithm proposed in Section IV was applied on the data window Y with all $|V|$ measurements. Figure 4 shows the signal importance scores Sc calculated for 69 signals with 3 classes of event examples and their total scores. As can be seen from Fig. 4, signals $|V_9|, |V_{40}|, |V_{50}|$ are participating in forming low rank subspace of more number of events. A group of $\hat{n}$ signals with the highest total scores were selected from set S. An estimation error formula is presented next for performance evaluation of the proposed heuristics in preserving the information in the entire data set.

Given m signals with N ($>> 2m$) samples from each training event example Y_h, suppose the proposed heuristic selects $\hat{n}$ signals of higher importance scores present in S (with the rest signals in set $\bar{S}$). Then, given pN ($>> m$)

samples of all m signals, where p (≥ 0.5) shows a fraction of N samples, the rest $(1-p)N$ samples of $(m-\hat{n})$ signals in $\bar{S}$ can be estimated using $(1-p)N$ samples of $\hat{n}$ signals in S with a least squares (LS) estimation procedure as follows.

- Given training event examples $Y_h, h \in [1 : H_{train}]$
- Initialize sets $S, \bar{S}, D, \bar{D}$.
- *while $h \leq H_{train}$* (for each Y_h)
  1) Calculate $\beta = \left(Y^{(S,D)}\right)^{\dagger} Y^{(\bar{S},D)}$; $\dagger$ indicates pseudo-inverse of a matrix.
  2) Estimate $\hat{Y}^{(\bar{S},\bar{D})} = Y^{(S,\bar{D})}\beta$
  3) Calculate LS estimation error for example event Y_h; $e_h = ||Y^{(\bar{S},\bar{D})} - \hat{Y}^{(\bar{S},\bar{D})}||_2$
- Calculate mean error μ_X across examples $h \in [1, H_{train}]$

Here $Y^{(S,D)}$ denotes submatrix of Y_h with signals present in set S and samples present in set D. Subscript h in notation $Y^{(,)}$ is dropped for simplicity. D and $\bar{D}$ show the samples in the range $[1, pN]$ and $[pN + 1, N]$, respectively. Using the above steps, an estimation error μ_X can be calculated for H_{train} example events.

A Monte-Carlo simulation was then conducted for 100 randomly generated signal combinations $\mathbf{W}$, each with $\hat{n} = 21$ number of signals such that for each randomly generated set $W_r \in \mathbf{W}, |W_r| = \hat{n} = 21$. The effect of considering selection of different number of signals in set $S_{W_r} \subset W_r, (\bar{S}_{W_r} = W_r - S_{W_r})$, and each with multiple combination of signals on the estimation error for all disturbance events was then compared against the proposed set S obtained using the proposed heuristics. For each combination of signals in each random set W_r, the estimation errors were calculated by considering $|S_{W_r}| = 1$ ($|\bar{S}_{W_r}| = 69 - |S_{W_r}|$), and in each subsequent iteration, $|S_{W_r}|$ was increased by 1 by including signals from the combination in W_r. Error μ_X in estimation of the unselected signals in $\bar{S}_{W_r}(\bar{S})$ is calculated as a function of set $S_{W_r}(S)$ and the number of signals $|S_{W_r}|(|S|)$ for each signal set W_r and the proposed set of signals in S – see, Fig. 5. A decrease in the estimation errors is observed by including more number of signals in set $S_{W_r}(S)$. This reduction is more significant for proposed set S as compared to any randomly selected set S_{W_r}. This shows the ability of the proposed set
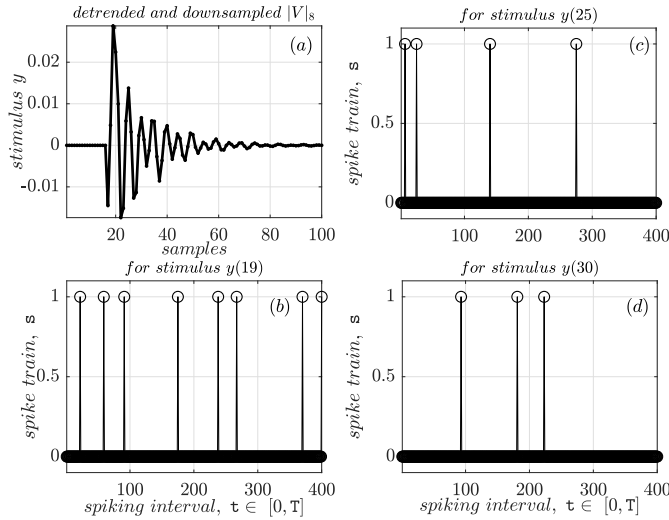
Fig. 6. Preprocessed voltage signal $|V_8|$ as stimulus y presented as input to SNN is shown in (a). (b)-(d) shows spike train representation of samples $y(19), y(25), y(30)$, respectively over a spike train duration of 400 samples.

S to contain most information present in the entire data set. Therefore, in this work, $|S| = \hat{n} = 10$ number of signals in S were selected to provide input to event detection and classification algorithms.

C. Spike Train Representation

In this step, the entire window of data matrix for each event example was converted to a 1-D vector by stacking the signal samples present in the window, which is further encoded as Poisson spike trains with an average spiking rate proportional to the maximum value of the data window. Figure 6(a) shows the detrended and downsampled version of voltage magnitude measurement signal from bus 8. The corresponding spike train representation for 19th, 25th, and 30th samples are shown in Fig. 6(b), (c), and (d), respectively. Spike train generated for each example event data was then presented as input to SNN.

D. SNN's Classification Performance Analysis

The experiment starts with training and inference on 10 cross-validation (CV) sets in the following supervised and unsupervised learning frameworks – 1) Unsupervised learning framework: a) U_1: Self-organising maps (SOM); b) U_2: LIF based SNN; 2) Supervised learning framework: ANN-SNN conversion based SNN. The following sections present classification results obtained with supervised and unsupervised learning techniques.

1) Unsupervised Learning: In this framework, a network of size $[\hat{n}N, 10]$ is presented, where $\hat{n}N$ represents the number input neurons as well as the number elements in the 1-D vector, which leads to $\hat{n}N = 1090$ for $|V|$ signals and $\hat{n}N = 2512$ for f signals. The number of neurons in output (excitatory)

layer of SOM (SNN) is kept 10. The average accuracy for event detection with an SNN on 10 CV tests is 96.5% for $|V|$ and 97.7% for f signals, which ensures unsupervised online event-driven operation. The accuracy for disturbance event classification is shown in Table II. The average accuracy on the test set is 92.4% with U_1 and 92.9% with U_2 for $|V|$ and 93.4% with U_1 and 96.8% with U_2 for f signals.

2) Supervised Learning: In this framework, a ReLU based ANN-SNN of size $[\hat{n}N, h_1, h_2, r]$ as described in Section V-B, is considered for supervised classification using voltage and frequency information of the simulated events with $\hat{n}N$ input nodes, two hidden layers each with h_1, h_2 nodes, respectively and r output nodes with all fully connected layers. Maximum number of epochs was kept to 30 and a mini batch size of 100 was adopted during training ANN through standard backpropagation. The weights from the ReLU network is then scaled to obtain the SNN version of the trained ANN. The results obtained with ANN-SNN conversion are presented in Table III, which shows the average classification accuracy is 96.3% for voltage magnitudes and 93.6% for frequency measurements and proves the effectiveness of SNN in supervised classification tasks. The following sections present a discussion on the computational performance analysis of both ANN and SNN algorithms.

E. Evaluation of Computational Burden

1) Unsupervised Learning: Although an estimation of the entire hardware overhead or actual energy consumption is outside the scope of the work, a comparison of number of computations per synaptic operations with techniques U_1 and U_2 are presented based on the number of multiply-accumulate (MAC) operation. A typical ANN with i input neurons and r output neurons requires ir multiplication and addition operations with one forward pass of the fully connected feed forward network. In contrast, SNN performs only accumulate (AC) operations over the entire spiking interval $[0, T]$ in layer j only upon receiving a spike from previous layer $j - 1$. Number of AC operation per layer j becomes a product of average number of accumulated spikes per neuron $s_m(j)$ from previous layer $j - 1$ during the spiking interval T and the number of synaptic operations (ir) between two consecutive layers. The ratio of number of MAC operations in SOM to AC operations in SNN is estimated to be ≈ 1.2, which shows a 20% reduction in number of computations with SNN.

2) Supervised Learning: After ANN-SNN conversion, the total number of computations in SNN during testing becomes $\hat{n}Nh_1s_m(2) + h_1h_2s_m(3) + h_2rs_m(4)$, which is much less as compared to $\hat{n}Nh_1 + h_1h_2 + h_2r$ operations involved with the trained ANN in the process. The ratio the number of MAC operations involved in ANN to the AC operations in SNN is ≈ 1.1. This ratio is expected to further increase with more

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY BETWEEN SOM (U_1) AND SNN (U_2) WITH UNSUPERVISED LEARNING USING $|V|$ AND f FEATURES

Feature type	CV Sets	1	2	3	4	5	6	7	8	9	10	Average accuracy
Voltage $|V|$	Training (SOM)	93.6	96	93.7	96.5	97	96.6	95	92.8	92.5	96	94.9
Voltage $|V|$	Test (SOM)	93.3	93	96	92	92	91.6	89	87.5	92.1	98	92.4
Voltage $|V|$	Training (SNN)	90	87	90	90.3	91.5	85	86	90.1	86	90.1	88.6
Voltage $|V|$	Test (SNN)	92.5	92	92	93	94.6	91	94	94.2	93	93	92.9
Frequency f	Training (SOM)	96	94	93	93.4	93.1	97	92	96.4	95	94.2	94.4
Frequency f	Test (SOM)	95	88	96.2	95.8	98	90	85	95.4	96.2	95	93.4
Frequency f	Training (SNN)	93	92.3	94	95.31	92.8	94	97	93.7	92.6	96.14	94.08
Frequency f	Test (SNN)	98	96.7	99.2	98.7	95.8	95	99	93.7	95.4	96.7	96.82

TABLE III
CLASSIFICATION ACCURACY OF SNN OBTAINED USING ANN-SNN BASED CONVERSION (S1) FOR SUPERVISED LEARNING

Feature type	CV Sets	1	2	3	4	5	6	7	8	9	10	Average accuracy
Voltage $|V|$	Training	98	99	98	98.6	98	99	99	97.3	98.3	99	98.42
Voltage $|V|$	Test	97	97	97.1	95.8	95	97.9	97.2	91.2	97.2	98	96.34
Frequency f	Training	98.3	99.5	99	98	97.8	99	99	97.3	97.6	99.1	98.4
Frequency f	Test	98	91.2	96.2	89	91	96.7	90	94.6	91.2	98	93.6

layers in networks [32], which is due to exponential increase in neuron spiking sparsity with network depth. This could be very useful at control centers for larger datasets. This proves the capability of spiking neurons in achieving similar accuracy as conventional techniques in disturbance classification with less computational burden.

Moreover, SNN is more efficient even for the worst case scenarios when an input pattern demands more binary AC operations than real-valued MAC operations, since SNNs can be implemented in energy efficient neuromorphic hardware [41], [42]. For instance, an estimated ratio of order of magnitude of energy consumption per MAC operation (with one 32-bit ADD and 32-bit MULT) to AC operation (with 32-bit ADD) is 5.1 for floating point computations [43]. A summary of projected energy savings for disturbance classification has been presented in Table IV, which shows SNN deployed for disturbance classification operations can save upto 83.6% of energy which is wasted by ANN in the unsupervised and upto 82% in supervised framework. This implies by deploying SNNs, which are inherently low energy networks and incurs less computations, the energy savings will significantly increase with deep networks. While an actual estimate is outside the scope of the work, the evidence presented through AC operations encourages implementation of the event-driven classifier in a neuromorphic computing hardware for realization of a spiking environment enabling low energy computations and efficient data processing without sacrificing accuracy.

The following section demonstrates the potential benefits of deep SNNs over ANNs in a new experiment to show the impact of training with more data at scale and corresponding effects on classification accuracy and computational performance.

VII. SCALING TO DEEPER NETWORKS

A. Experimental Setup

While the results in the previous section have provided motivation for SNN usage in power system disturbance classification, computational burden reduction becomes a significant issue in complex problem spaces where deep networks exceeding a million parameters become necessary to achieve state of the art accuracy. To substantiate the benefits of SNN usage in such scenarios, a new experiment was conducted with additional class data. Due to limitations in the availability of real-time PMU data and an annotated dataset containing information on the type of disturbance in the general power system community, a set of simulations on the existing test

TABLE IV
COMPARISON OF PERFORMANCE MEASURES BETWEEN ANN AND SNN

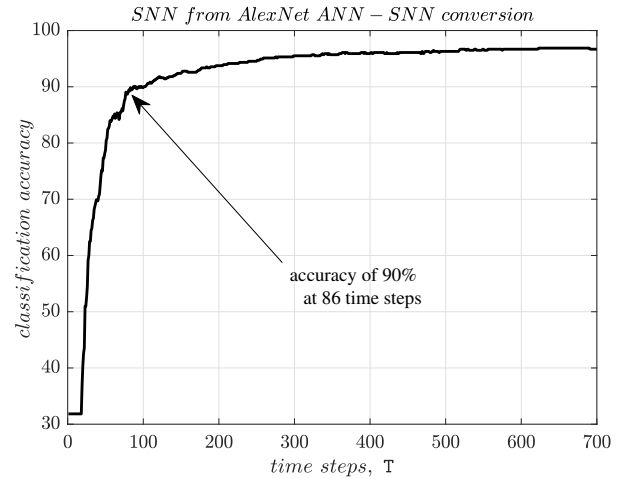| For each MAC (ANN) to AC (SNN) operation	Unsupervised Learning	Supervised Learning
Ratio of number of operations	1.2	1.1
Ratio of Energy consumption	6.12	5.61
% of Energy saving with SNN	83.67	82.18

Fig. 7. The classification accuracy of SNN after AlexNet based ANN-SNN conversion is plotted as a function of time steps T. SNN achieved 90% validation accuracy at 86 timesteps. The temporal behavior can be exploited for accuracy-efficiency tradeoff.
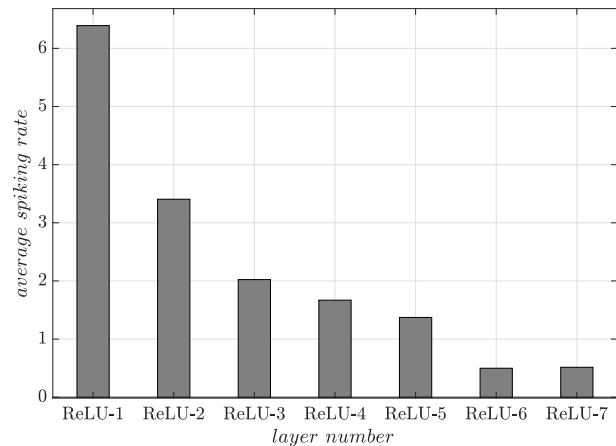

Fig. 8. Average spiking activity of the converted AlexNet is plotted as a function of network depth. The SNN spiking sparsity significantly increases as the network depth increases.

system were performed for collecting time series measurements involving transient data from disturbance events. Four classes of events considered in this experiment are: (1) bus faults (BF), (2) generator outages (GO), (3) load trip (LT) and (4) line outage (LO) cases under 7 different operating conditions with varying loading levels across the network. This resulted in 483 BF, 112 GO, 490 LT, 490 LO labeled cases. A 30 seconds of simulation data was extracted from the selected $|V|$ signals and preprocessed for providing inputs to the neural network.

B. Supervised Training Methodology

Next, we trained a convolutional network architecture by *supervised learning* and subsequently converted that to a spiking network using ANN-SNN conversion. The SNN simulation is performed using a modified version of BindsNet [44], a PyTorch based package. The AlexNet [45] architecture, an 8-layer deep network consisting of 5 convolutional layers and 3 fully

connected layers was selected. Without loss of generality, the network was trained on voltage magnitude features. The training can be extended for other features as well. It is worth noting here that the convolutional layers used in our work are 1D, ideal for time-series data processing. The dataset was then randomly split into two parts: 2/3 for training and 1/3 for validation. The network was trained with a learning rate of 0.001 and a batch size of 32, and subsequently converted to spiking mode for inference. To reduce the high inference latency observed in deep SNNs, a fixed percentile (96.4% in our case) from the maximum activation histogram was used during the weight normalization process. Such a relaxed normalization scheme significantly reduces the timesteps involved in inference without any observable degradation in classification accuracy [46]. For an extensive discussion on constraints involved in ANN-SNN conversion and design-time/run-time optimizations, readers are referred to reference [46].

C. Advantages with Deeper SNN

Figure 7 depicts the SNN classification accuracy as a function of timesteps T. This temporal behavior, where the accuracy of the network increases over timesteps, can be leveraged for accuracy-efficiency trade-off - appealing for resource-constrained applications. For instance, the SNN achieves 90% accuracy at 86 time steps and subsequently saturates in terms of accuracy improvement. Hence, computations involved in the SNN can be significantly reduced by small relaxation in the final accuracy requirement. Using the methodology described in the previous section, the SNN is estimated to be $2.71\times$ energy efficient than the corresponding ANN to achieve a final accuracy of 90%. It is worth mentioning here that SNNs are power efficient due to sparse event-triggered neural operation. However, the power advantage is achieved at the expense of inference delay.

While instantaneous power consumption is a key metric driving computational cost overhead for resource constrained devices, resultant energy efficiency (delay x power) will be a trade-off between inference latency (impacting delay) and sparsity of the spike train (impacting power). In this particular example, the power efficiency of the SNN is estimated to be $233\times$ in contrast to the ANN since the inference takes 86 timesteps to reach 90% accuracy. Note that this is a first order estimate as mentioned before based on computational energy requirements.

Additionally, Fig. 8 demonstrates that the SNN spiking sparsity significantly increases as the network depth increases. This downward trend of spiking rate as layers go deeper implies SNN's even higher power/energy efficiency when implementing deeper architectures such as VGG [47] and ResNet [48], which are essential to solve more complex problems.

VIII. CONCLUSION

This work successfully demonstrated the ability of spiking neural networks (SNN) to process PMU signals for online event-driven disturbance classification. SNNs can exploit the sparsity patterns of events to achieve classification accuracy comparable to ANNs while significantly reducing the energy consumption. In addition, a signal selection technique to determine the signals participating in low rank subspace in multiple disturbance event scenarios was proposed, whose effectiveness was demonstrated by reconstructing samples of multiple signals using the selected signals. Performance comparison of ANN and SNN from classification tests on 16-machine, 5-area New England-New York system shows that SNN has the capability to achieve 96.8% disturbance classification accuracy with upto 20% reduction in number of computations in terms of MAC to AC operations. Event-driven SNNs in unsupervised (supervised) learning framework results in a projected estimate of energy savings upto 83.6% (82.18%) with respect to standard ANNs. In addition, our results on deep SNNs also highlighted the fact that dynamic sparsity benefits would be best realized for deeper networks which, in turn, is a necessity for complex machine learning tasks. Hence, the energy and power benefits of SNNs are expected to further improve with increasing complexity of the power grid disturbance classification problem and in real time operational scenarios. This work opens up the new possibility of utilization of neuromorphic computing algorithms and hardware at control centers for real time disturbance monitoring.

IX. APPENDIX

A. Measurement data

$y(t)$ can be expressed in terms of the linearized system matrices (A_o, B_o, C_o, D_o) of the network [49] at an operating condition (o).

$$\bar{y}(t) = C_o \Phi_o L_o \bar{z}(t) \tag{14}$$

where, Φ_o is the modal matrix of A_o, columns of which contains the right eigenvectors $\bar{\phi}_i$ of A_o. $\bar{z}(t)$ represents the complex exponential associated with n eigenvalues $[\lambda_1, ... \lambda_n]$ of the system at the operating condition and can be expressed as

$$\bar{z}(t) = \begin{bmatrix} e^{\lambda_1 t} & ... & e^{\lambda_n t} \end{bmatrix}^T \tag{15}$$

Each row in matrix L_o contains vector $\bar{l} = [l_1 l_2 ... l_n]$, in which each element l_i is a function of initial condition $\nabla x(0)$ and left eigenvectors $\bar{\psi}_i$ of A_o.

$$l_i = \bar{\psi}_i \nabla x(0) \tag{16}$$

where, Ψ_o contains the left eigenvectors $\bar{\psi}_i$. Discretized system response Y $(N \times m)$ containing N samples for $t = [0, (N - 1)Ts]$ of m measured signals at an operating condition (o), can be expressed as follows.

$$Y^T = C_o \Phi_o L_o V = F_o V \Rightarrow Y = V^* F_o^* \tag{17}$$

where, V is a Vandermonde matrix $(n \times N)$ with n distinct complex modes $[v_1, ..., v_n]$ in the z-plane inside the unit circle corresponding to eigenvalues $\lambda_i = F_s ln(v_i)$ in continuous domain. Therefore, V^* contains n columns with n complex exponential time series signals $\bar{v}_i$ representing the trajectories of modal signals.

$$V^* = \begin{bmatrix} v_1^0 & \cdots & v_n^0 \\ \vdots & \ddots & \vdots \\ v_1^{N-1} & \cdots & v_n^{N-1} \end{bmatrix} \tag{18}$$

The data window Y can be written as a function of V in (17), where, rows $\bar{y}(t) \in \mathbb{R}^m$ of Y are a complex transformation of each row $\bar{v}(t) \in \mathbb{C}^n$ of V^* by transformation matrix F_o^* for each $t \in [0, (N - 1)Ts]$. So we can write $\bar{y}(t) = \chi_{f_o}(\bar{v}(t))$, where $y(t)$ is the image of $\bar{v}(t)$ under mapping χ_{f_o}. For distinct v_i's, the columns of V^* forms a basis of a set of linearly

independent vectors of complex exponential time series signals and defines the dimension of Y as n when $m >= n$.

B. Comparison of classification accuracy of SNN with other machine learning techniques

Due to the "architectural similarity" between ANN and SNN and the applicability of ANNs in learning from large-scale datasets, ANN is presented for comparison of accuracy and computational efficiency. Nevertheless, the results with other classification methods are presented here.

TABLE V
COMPARISON OF CLASSIFICATION ACCURACY

Accuracy (%)	DT	SVM	RF	SNN
Training ($\|V\|$)	98.6	95.5	95.8	98.4
Testing ($\|V\|$)	96.6	97.01	93	96.3
Training (f)	98.7	99.2	94.4	98.4
Testing (f)	96.2	95.08	94.5	93.6

Note that the objective here is to attain a similar accuracy as other machine learning methods with reduced computational overhead. For real-time operations, ANNs or deep ANNs are preferable for large data set as compared to other classification techniques due to the capability of ANNs in representing highly nonlinear functions/decision boundaries. The results in Table V indicate that SNN has the capability to achieve similar accuracy as other machine learning techniques such as decision tree (DT), support vector machines (SVM), random forest (RF).

Remarks: Note that for large-scale problems, deep neural networks are known to outperform the traditional ML techniques such as SVM, DT, RF etc. in regression or classification tasks. It is a known fact that traditional ML techniques are not suitable in achieving generalization while learning from large datasets. Reference [50] shows how the scale of the problem drives machine learning progress and highlights the limitation of traditional ML techniques.

While latest deep neural networks present the best platform to learn from large datasets, a significant improvement in computational efficiency can be achieved through the use of alternative neuromorphic SNN architectures. Thus, deep SNNs are proposed for large power system event detection problems, which can achieve similar accuracy as ANNs, while saving significant energy consumption due to its sparse, event-driven, temporal operation.

REFERENCES

[1] S. Brahma, R. Kavasseri, H. Cao, N. Chaudhuri, T. Alexopoulos, and Y. Cui, "Real-time identification of dynamic events in power systems using pmu data, and potential applications—models, promises, and challenges," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 294–301, 2017.

[2] R. Yadav, A. K. Pradhan, and I. Kamwa, "Real-time multiple event detection and classification in power system using signal energy transformations," *IEEE Transactions on Industrial Informatics*, 2018.

[3] Y. Tang and J. Yang, "Dynamic event monitoring using unsupervised feature learning towards smart grid big data," in *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 1480–1487, IEEE, 2017.

[4] G. Gajjar and S. Soman, "Auto detection of power system events using wide area frequency measurements," in *National Power Systems Conf.(NPSC), Guwahati, India*, pp. 1–6, 2014.

[5] O. P. Dahal, S. M. Brahma, and H. Cao, "Comprehensive clustering of disturbance events recorded by phasor measurement units," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1390–1397, 2014.

[6] O. P. Dahal and S. M. Brahma, "Preliminary work to classify the disturbance events recorded by phasor measurement units," in *Power and Energy Society General Meeting, 2012 IEEE*, pp. 1–8, IEEE, 2012.

[7] S. Lavand, G. Gajjar, S. Soman, and R. Gajbhiye, "Mining spatial frequency time series data for event detection in power systems," 2016.

[8] M. Aravind and A. T. Mathew, "Pmu data based post disturbance analysis for a large grid using wavelets and lyapunov exponent," in *Power Systems Conference (NPSC), 2016 National*, pp. 1–6, IEEE, 2016.

[9] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, and Y.-J. Shin, "Wavelet-based event detection method using pmu data," *IEEE Transactions on Smart grid*, vol. 8, no. 3, pp. 1154–1162, 2017.

[10] S. M. Bhuiyan, J. Khan, and G. Murphy, "Wpd for detecting disturbances in presence of noise in smart grid for pq monitoring," *IEEE Transactions on Industry Applications*, vol. 54, no. 1, pp. 702–711, 2018.

[11] M. Biswal, Y. Hao, P. Chen, S. Brahma, H. Cao, and P. De Leon, "Signal features for classification of power system disturbances using pmu data," in *Power Systems Computation Conference (PSCC), 2016*, pp. 1–7, IEEE, 2016.

[12] Y. Chen, L. Xie, and P. R. Kumar, "Power system event classification via dimensionality reduction of synchrophasor data," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014 IEEE 8th*, pp. 57–60, IEEE, 2014.

[13] M. Rafferty, X. Liu, D. M. Laverty, and S. McLoone, "Real-time multiple event detection and classification using moving window pca," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2537–2548, 2016.

[14] I. Niazazari and H. Livani, "Disruptive event classification using pmu data in distribution networks," in *Power & Energy Society General Meeting, 2017 IEEE*, pp. 1–5, IEEE, 2017.

[15] H. Okumus and F. M. Nuroglu, "Event detection and classification algorithm using wide area measurement systems," in *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, pp. 230–233, IEEE, 2018.

[16] D. Phillips and T. Overbye, "Distribution system event detection and classification using local voltage measurements," in *Power and Energy Conference at Illinois (PECI), 2014*, pp. 1–4, IEEE, 2014.

[17] T. Guo and J. V. Milanović, "Online identification of power system dynamic signature using pmu measurements and data mining," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1760–1768, 2016.

[18] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao, and X. Liang, "Smart grid line event classification using supervised learning over pmu data streams," in *Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International*, pp. 1–8, IEEE, 2015.

[19] Y. Zhang and K. Tomsovic, "Real-time transient instability detection based on decision trees," in *Intelligent System Applications to Power Systems, 2009. ISAP'09. 15th International Conference on*, pp. 1–6, IEEE, 2009.

[20] T. Yin, S. S. Wulff, J. W. Pierre, D. Duan, D. J. Trudnowski, and M. Donnelly, "Initial investigation of data mining applications in event classification and location identification using simulated data from minniwecc," in *North American Power Symposium (NAPS), 2016*, pp. 1–6, IEEE, 2016.

[21] O. P. Dahal, H. Cao, S. Brahma, and R. Kavasseri, "Evaluating performance of classifiers for supervisory protection using disturbance data from phasor measurement units," in *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES*, pp. 1–6, IEEE, 2014.

[22] K. Seethalekshmi, S. Singh, and S. Srivastava, "A classification approach using support vector machines to prevent distance relay maloperation under power swing and voltage instability," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1124–1133, 2012.

[23] M. Biswal, S. M. Brahma, and H. Cao, "Supervisory protection and automated event diagnosis using pmu data," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1855–1863, 2016.

[24] Y. Zhu, C. Liu, and K. Sun, "Image embedding of pmu data for deep learning towards transient disturbance classification," in *2018 IEEE International Conference on Energy Internet (ICEI)*, pp. 169–174, May 2018.

[25] "Nanotechnology-Inspired-Grand-Challenges." https://www.nano.gov/grandchallenges.

[26] "IBM simulates 4.5 percent of the human brain, and all of the cat brain." https://www.scientificamerican.com/article/graphic-science-ibm-simulates-4-percent-human-brain-all-of-cat-brain/.

[27] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[28] "NERC-State-of-Reliability-Report,June,2019." www.nerc.com/pa/RAPA/PA/Performance/20Analysis/20DL/NERC_SOR_2019.pdf.

[29] B. Pal and B. Chaudhuri, *Robust control in power systems*. Springer Science & Business Media, 2006.

[30] M. H. Bollen, *Voltage SagsCharacterization*, pp. 139–251. 2000.

[31] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, pp. 82–99, January 2018.

[32] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, 2019.

[33] "Spike train." http://www.stat.cmu.edu/ kass/papers/SpikeTrain.pdf.

[34] G. A. Orban, *Neuronal operations in the visual cortex*, vol. 11. Springer Science & Business Media, 2012.

[35] Z. Chen, M. Johnson, L. Wei, and W. Roy, "Estimation of standby leakage power in cmos circuit considering accurate modeling of transistor stacks," in *Proceedings. 1998 International Symposium on Low Power Electronics and Design (IEEE Cat. No. 98TH8379)*, pp. 239–244, IEEE, 1998.

[36] G. Golub and C. Van Loan, "Matrix Computations, 3rd edition, The John Hopkins University Press," *Baltimore, MD*, 1996.

[37] A. Sengupta, A. Banerjee, and K. Roy, "Hybrid spintronic-cmos spiking neural network with on-chip learning: Devices, circuits, and systems," *Physical Review Applied*, vol. 6, no. 6, p. 064003, 2016.

[38] P. U. Diehl, D. Neil, J. Binas, M. Cook, S. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2015.

[39] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, pp. 54–66, May 2015.

[40] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. M. McGinnity, "A review of learning in biologically plausible spiking neural networks," *Neural Networks*, vol. 122, pp. 253–272, 2020.

[41] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, p. 774, 2018.

[42] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch, C. di Nolfo, P. Datta, A. Amir, B. Taba, M. D. Flickner, and D. S. Modha, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences*, vol. 113, no. 41, pp. 11441–11446, 2016.

[43] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *CoRR*, vol. abs/1506.02626, 2015.

[44] H. Hazan, D. J. Saunders, H. Khan, D. Patel, D. T. Sanghavi, H. T. Siegelmann, and R. Kozma, "Bindsnet: A machine learning-oriented spiking neural networks library in python," *Frontiers in Neuroinformatics*, vol. 12, p. 89, 2018.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[46] S. Lu and A. Sengupta, "Exploring the connection between binary and spiking neural networks," *arXiv preprint arXiv:2002.10064*, 2020.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[49] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*, vol. 7. McGraw-hill New York, 1994.

[50] A. Ng, "Machine learning yearning," *URL: https://www.deeplearning.ai/content/uploads/2018/09/Ng-MLY01-12.pdf*, 2017.

Kaveri Mahapatra (S'11-S'13-S'20) received her Ph.D. degree from Pennsylvania State University, USA in 2020 in Electrical Engineering. In 2019, she worked with General Electric (GE) Global Research Center, NY, USA as a research fellow intern. She has received her B. Tech. and M. Tech. degree in Electrical Engineering from KIIT University in 2011 and SOA University, India, in 2013, respectively. Presently, she is working as a power systems research engineer with Pacific Northwest National Laboratory, WA, USA. Her current research interests include wide area monitoring, protection and control, power system dynamics and resilience, cyber physical security, machine learning, data analytics and optimization.
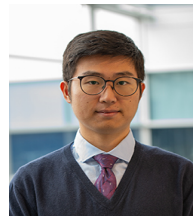
Sen Lu received the Computer Science Bachelor's degree from the Department of Computer Science and Engineering, College of Engineering, Pennsylvania State University, University Park, USA, in 2019.

Sen is currently pursuing a Ph.D. degree in Computer Science and Engineering, in the College of Engineering, Pennsylvania State University, USA. He joined the Neuromorphic Computing Lab of Penn State in 2019. His current research focuses on Computer Vision, Deep Learning, and Neuromorphic Algorithms.

Abhronil Sengupta (S'15-M'18) received the B.E. degree from Jadavpur University, Kolkata, India, in 2013, and the Ph.D. degree in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 2018.

He is an Assistant Professor with the School of Electrical Engineering and Computer Science, Penn State University, University Park, PA, USA. He was as a DAAD (German Academic Exchange Service) Fellow with the University of Hamburg, Hamburg, Germany, in 2012, and as a Graduate Research Intern with Circuit Research Labs, Intel Labs, Hillsboro, OR, USA, in 2016 and Facebook Reality Labs, Redmond, WA, USA, in 2017. He is pursuing an interdisciplinary research agenda at the intersection of hardware and software across the stack of sensors, devices, circuits, systems, and algorithms for enabling low-power event-driven cognitive intelligence. He has published over 70 articles in referred journals and conferences and holds 4 granted/pending U.S. patents.

Prof. Sengupta was a recipient of the IEEE Circuits and Systems Society Outstanding Young Author Award in 2019, the IEEE SiPS Best Paper Award in 2018, the Schmidt Science Fellow Award Nominee in 2017, the Bilsland Dissertation Fellowship in 2017, the CSPIN Student Presenter Award in 2015, the Birck Fellowship in 2013, and the DAAD WISE Fellowship in 2012.

Nilanjan Ray Chaudhuri (S'08-M'09-SM'16) received the Ph.D. degree in power systems from Imperial College London, London, UK in 2011. From 2005 to 2007, he worked in General Electric (GE) John F. Welch Technology Center. He came back to GE and worked in GE Global Research Center, NY, USA as a Lead Engineer during 2011 to 2014. Presently, he is an Associate Professor with the School of Electrical Engineering and Computer Science at Penn State, University Park, PA. He was an Assistant Professor with North Dakota State University, Fargo, ND, USA during 2014-2016. He is a member of the IEEE and IEEE PES. Dr. Ray Chaudhuri is the lead author of the book Multi-terminal Direct Current Grids: Modeling, Analysis, and Control (Wiley/IEEE Press, 2014). He served as an Associate Editor of the IEEE TRANSACTIONS ON POWER DELIVERY (2013 – 2019) and IEEE PES LETTERS (2016 - present). Dr. Ray Chaudhuri was the recipient of the National Science Foundation Early Faculty CAREER Award in 2016 and Joel and Ruth Spira Excellence in Teaching Award in 2019.