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. 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

A DVENT 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 [8], [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 [9], [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] 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. Therefore, power grid’s secure operation comes at a cost of significant increase in its carbon footprint, which counters the sustainability goals in today’s world.

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 \[25\] 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 | |
|----------------------------------------------------------|---------------------------|
| **Transmit Unavailability (%)**                          | 0.22 | 0.27 | 0.24 | 0.22 |
| **Number of Disturbance Control Standard Events (Range)**| -250 | -575 | -150 | -175 |
|                                                         | -400 | -425 | -200 | -225 |

In addition, based on a 15s window of simulated data from the New England-New York test system \[25\], a typical histogram of distribution of signal magnitudes for 363 events is shown in Fig. 1(a). It indicates the probability of occurrence of absolute values of signal magnitudes in different ranges after normalization. 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. 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 \[27\]. Recent developments show that SNN now has the potential to achieve similar accuracy as of other ML techniques with much less power consumption \[28\]. 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 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 \([29]\). 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” \([30]\) to convert an input sample to a certain number of spikes while including an element of stochasticity.

The \(i^{th}\) sample \(y(i)\) of a discrete time-series measurement \([\hat{g}, i \in [1, N]]\) is represented by first associating it with an input neuron \(i\). With a homogeneous Poisson process assumption, the spike generation rate \(\lambda_i\) 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 \([0, T]\) to generate spikes at a rate \(\lambda_i = y(i)\). The spike train interval \(T\) is subdivided into shorter intervals \(\delta t\) such that \(t = j\delta t\), where \(j = 0, 1, 2, ..., T/\delta t\). The probability of spikes occurring during each \(\delta t\) is set to \(r \delta t\). For each \(\delta t\), a uniformly distributed random number \(\chi(\delta t)\) is generated. This is followed by a spike generation as per the following rule.

\[
\begin{cases}
1, & \chi(\delta t) \leq r \delta t \\
0, & \text{otherwise}
\end{cases}
\Rightarrow s = \sum_{j=0}^{T/\delta t} \delta (t - tj)
\]

where, \(s\) is the spike train generated over the interval \([0, T]\) with spiking instances \(t_j\) for sample \(y(i)\). \(s\) can also be represented as a set of Dirac-\(\delta\) functions at time instants \(t_j\) for \(j = 1, 2, ..., T/\delta t\). These spike trains are given as input to the excitatory layer of SNN and has been demonstrated in Fig. [3] 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 \([27]\).

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” \([31]\) 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 lower power consumption.

IV. Signal Selection

PMU data 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 \(O(mN)\) for online feature extraction from a \(N \times m\) data matrix. To overcome this, a feature selection procedure based on economy QR decomposition \([32]\) 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.

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 expressed as a product of Vandermonde matrix \(V\) and complex transformation matrix \(F_o\).

\[
Y^T = F_oV \Rightarrow Y = V^*F_o^*
\]

The rectangular matrix \(Y\) can be decomposed into two factors using economy QR decomposition \([32]\), which rearranges signals present in \(Y\) in decreasing order of priority for obtaining a linearly independent set of signals in the columns of \(\hat{Y}\).

\[
YP = \hat{Y} = QR \Rightarrow Q^T\hat{Y} = R
\]

where, \(Q\) represents an orthonormal basis for subspace of \(\hat{Y}\) and \(R\) represents an upper triangular matrix \(r_{k,l} = \hat{q}_k^T \hat{y}_l\). \(P\) represents permutation matrix, which permutes the columns of \(Y\) to get nonincreasing diagonal entries in \(R\). 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 \((v_1, v_2, ..., v_m)\) of \(V\) are contributing in forming \(U\). Since, \(Y\) is low rank, the rank of \(U\) becomes \(n < m << \infty\). Since \(Q \subseteq R(Y)\), where \(R\) indicates the column space, it can be written as a mapping from \(V^*\) as follows.

\[
Q = V^*g_o^* \Rightarrow Q^T = G_oV
\]

\[
Q^TQ = I = GV^*G^* = GVG^*
\]

\[
GV = (G^*)^T = (G^T)^T
\]

Now substituting expression for \(G^T\) in \((3)\) for \(R\) leads to

\[
R = Q^T\hat{Y} = GVV^*F^* = GV^*F^* = (G^T)^*F^*
\]

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 \(\hat{g}_k^T f_l\), which is the projected component of \(\hat{f}_l\) in the direction of \(\hat{g}_k\) of \(Q\).

\[
\hat{f}_l = 0, \forall l, k = l + 1 : m
\]

\[
\hat{g}_k^T f_l \geq \hat{g}_{k+1}^T f_l \geq ... \hat{g}_m^T f_l \geq ... \geq \hat{g}_{m-l}^T f_{m-l} \geq \hat{g}_{m-l+1}^T f_{m-l+1} \geq ... \hat{g}_m^T f_m
\]

\[
\hat{g}_k^T f_k = 0, n < k \leq m
\]

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 \( n < n_m \), a very low dimensional subspace \( Q \) (tall matrix with \( n_m \) 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 \( n \) 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 \( h \in [1 : H_{\text{train}}] \), where \( H_{\text{train}} \) 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 \( \nu_1 \) 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 \( h < H_{\text{train}} \)
    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 \( Y \) for economy QR decomposition.
    3) Find set \( \hat{S} = \{ s \mid s \in p, r_{s,s} \geq \nu_1 r_{1,1} \} \).
    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.
  \( Sc(j) = CC(j) / H_{\text{train}} \).
- 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 \( h = 1 : H_{\text{train}} \).

Given \( m \) number of signals for any event in \( Y_h \), signals in \( S_h \) forming the low-rank subspace \( U_h \) is important for the event \( h \). Cardinality \( (\hat{n}) \) of set \( S_h \) is an estimate of the number of high energy modes \( \lambda_s \) present in \( m \) signals. Given \( H_{\text{train}} \) 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 [33] 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 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.

\[
\text{EPSP}(t) = \begin{cases} 
1, & t_c \leq t \leq t_d, s(t_c) = 1 \\
0, & \text{otherwise}
\end{cases}
\]  

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

\[
\tau_{\text{mem}} \frac{du_{\text{post}}(t)}{dt} = -u_{\text{post}}(t) + w(t) \sum \limits_{j} \text{EPSP}(t) - I
\]

where, \( \tau_{\text{mem}} \) and \( \tau_{\text{int}} \) are the EPSP and membrane time constants, respectively. \( u_{\text{post}} \) is being reset after reaching a threshold \( u_{\text{th}} \) and results in spiking of post-synaptic neuron in the excitatory layer. Immediately after this event, \( u_{\text{post}} \) is not allowed to change for a refractory period. In addition, Homeostasis was adopted to prevent single neuron from dominating the response pattern, where \( u_{\text{th}} \) for the spiking neuron is updated to a higher value at a rate of \( \theta_r \) as follows.
This technique exploits the relationship between number of load trip cases were created via a contingency analysis event data set consisting of bus faults, generator outages, and load trip cases. A disturbance can be tested for classification of disturbances in the test set. The final weight-normalized spiking network scaled according to the maximum possible activation within the maximum ReLU activations per layer. The weights are then propagating the training set through the network and recording of neuron thresholds to synaptic weights. This is achieved by normalization is adopted to optimize the ratio of synaptic weights based on 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.

\[
\Delta w = \begin{cases} 
    \zeta_+ e^{(-\Delta t/\tau_+)} & \Delta t > 0 \\
    \zeta_- e^{(-\Delta t/\tau_-)} & \Delta t < 0 
\end{cases}
\]

where, \(\Delta t = t_{\text{post}} - t_{\text{pre}}\), \(t_{\text{post}}\) and \(t_{\text{pre}}\) are the pre- and post-synaptic firing time instances, respectively. \(\tau_+\), \(\tau_-\), \(\zeta_+\), \(\zeta_-\) 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. 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 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.

\[
v_j = \max(0, \sum_i w_{ij} y_i) \quad (12)
\]

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

\[
u_{\text{mem}}(t + 1) = u_{\text{mem}}(t) + \sum_i w_{ij} s^{y_i}(t) \quad (13)
\]

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 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. The final weight-normalized spiking network can be tested for classification of disturbances in the test set.

### VI. RESULTS AND PERFORMANCE EVALUATION

#### A. Experimental Setup

A 16-machine, 5-area New England-New York system is considered with PMUs installed at all 69 buses. A disturbance event data set consisting of bus faults, generator outages, and load trip cases were created via a contingency analysis.

\[
\Rightarrow u_{th} = u_{th} + \theta_u u_{th} \quad (10)
\]

*In this framework, a biologically plausible unsupervised learning mechanism based on STDP 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 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.

#### B. Signal Selection on Voltage Magnitude Measurements

During each simulation, disturbance was applied at 10s. Three traces show the type of disturbance; Class-A: bus fault, Class-B: generator outage, Class-C: load trip cases. The total score \(Sc\) for each signal and is shown in (d). The total score \(Sc\) is calculated for each signal and is shown in (d).

#### 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 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 4 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 $n$ 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 $n$ signals of higher importance scores present in $S$ (with the rest signals in set $S$). Then, given $pN$ ($> m$) samples of all $m$ signals, where $p \geq 0.5$ shows a fraction of $N$ samples, the rest $(1 - p)N$ samples of $(m - n)$ signals in $S$ can be estimated using $(1 - p)N$ samples of $n$ 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 $S, S, D, D$.
- while $h \leq H_{train}$ (for each $Y_h$)
  1) Calculate $\beta = (Y^{(S,D)}\dagger)Y^{(S,D)}; \dagger$ indicates pseudo-inverse of a matrix.
  2) Estimate $Y^{(S,D)} = Y^{(S,D)}\beta$
  3) Calculate LS estimation error for example event $Y_h$; $\epsilon_h = \|Y^{(S,D)} - Y^{(S,D)}\|_2$
- Calculate mean error $\mu_X$ 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 $Y^{(\cdot)}$ is dropped for simplicity. $D$ and $D$ show the samples in the range $[1, pN]$ and $[pN + 1, N]$, respectively. Using the above steps, an estimation error $\mu_X$ can be calculated for $H_{train}$ example events.

A Monte-Carlo simulation was then conducted for 100 randomly generated signal combinations $W$, each with $n = 21$ number of signals such that for each randomly generated set $W_r \in W$, $|W_r| = n = 21$. The effect of considering selection of different number of signals in set $S_{W_r} \subset W_r, (S_{W_r} = W_r - S_{W_r})$, 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 $|S_{W_r}| = 1 (|S_{W_r}| = 69 - |S_{W_r}|)$, and in each subsequent iteration, $|S_{W_r}|$ was increased by 1 by including signals from the combination in $W_r$. Error $\mu_X$ in estimation of the unselected signals in $S_{W_r}(S)$ 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 $S$ to contain most information present in the entire data set. Therefore, in this work, $|S| = n = 10$ 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 downsampling 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 ANN.

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.

1) Unsupervised Learning: In this framework, a network of size $[nN, 10]$ is presented, where $nN$ represents the number input neurons as well as the number elements in the 1-D vector, which leads to $nN = 1090$ for $|V|$ signals and $nN = 2512$ 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
7.7% for f signals, which ensures unsupervised event-driven operation. The accuracy for disturbance event classification is shown in Table I. The average accuracy on the test set is 92.4% with U1 and 92.9% with U2 for |V| and 93.4% with U1 and 96.8% with U2 for f signals.

2) Supervised Learning: In this framework, a ReLU based ANN-SNN of size \([n,N,h_1,h_2,r]\) as described in Section V-B, is considered for supervised classification using voltage and frequency information of the simulated events with \(nN\) 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 ANN version of the trained ANN. 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.

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 \(\approx 1.2\), 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 \(\hat{n}Nh_1s_m(2) + h_1h_2s_m(3) + h_2r_s_m(4)\), 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 \(\approx 1.1\). This ratio is expected to further increase with more layers in networks [28], 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 classification accuracy as conventional techniques in disturbance classification with less AC operations.

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 [36, 37]. 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 [38]. 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 up to 83.6% of energy which is wasted by ANN in the unsupervised and up to 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.

VII. 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 up to 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 up to 83.6% (82.18%) with respect to standard ANNs. This work opens up the new possibility of utilization of neuromorphic computing algorithms and hardware at control centers for real-time disturbance monitoring.

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] E. 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. IEEE, 2017, pp. 1480–1487.
[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, 2014, pp. 1–6.
[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. IEEE, 2012, pp. 1–8.
[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. Aarvind and A. T. Mathew, “Pmu data based post disturbance analysis for a large grid using wavelets and layapunov exponent,” in Power Systems Conference (NPSC), 2016 National. IEEE, 2016, pp. 1–6.
[9] Y. Li, J. T. Chu, X. H. Zong, G. Lee, and Y. J. Shinn, “Wavelet-based event detection method using pmu data,” IEEE Transactions on Smart Grid, vol. 8, no. 3, pp. 1154–1162, 2017.
[10] S. M. Bhiyani, 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. IEEE, 2016, pp. 1–7.
[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. IEEE, 2014, pp. 57–60.
[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. Niazzari and H. Livani, “Disruptive event classification using pmu data in distribution networks,” in Power & Energy Society General Meeting, 2017 IEEE. IEEE, 2017, pp. 1–5.
[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). IEEE, 2018, pp. 230–233.
[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. IEEE, 2014, pp. 1–4.
[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. IEEE, 2015, pp. 1–8.
[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. IEEE, 2009, pp. 1–6.
[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 minnievac,” in North American Power Symposium (NAPS), 2016. IEEE, 2016, pp. 1–6.
[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. IEEE, 2014, pp. 1–6.
[22] K. Senthaleshmi, 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), May 2018, pp. 169–174.
[25] “NERC-State-of-Reliability-Report.June,2019,” www.nerc.com/pa/RAPA/PA/Performance/20Analysis/20DL/NERC-SOR-2019.pdf

[26] B. Pal and B. Chaudhuri, “Robust control in power systems.” Springer Science & Business Media, 2006.
[27] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Islam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaitkitty, 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, no. 1, pp. 82–99, January 2018.
[28] 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.
[29] “Spike train,” http://www.stat.cmu.edu/~kass/papers/SpikeTrain.pdf

[30] A. Wild, 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.
[31] “Spike train,” http://www.stat.cmu.edu/~kass/papers/SpikeTrain.pdf

[32] A. Wild, 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