IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 November 2022, accepted 26 December 2022, date of publication 29 December 2022,
date of current version 25 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3233219

RESEARCH ARTICLE

Automated Power System Fault Prediction and Precursor Discovery Using Multi-Modal Data

MOHAMMAD ALQUDAH[1], **(Student Member, IEEE),**
MLADEN KEZUNOVIC[2], **(Life Fellow, IEEE), AND**
ZORAN OBRADOVIC[1], **(Senior Member, IEEE)**
[1]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[2]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA

Corresponding author: Zoran Obradovic (zoran.obradovic@temple.edu)

This work was supported by the Department of Energy under Award DE-OE0000913.

ABSTRACT Electric power system operators monitor large multi-modal data streams from wide service areas. The current data setups stand to get more complex as utilities add more smart-grid sensors to collect additional data from power system substations and other in-situ locations. We propose a methodology to utilize multi-modal data for automated power system fault prediction, and precursor discovery that takes advantage of not only the utility owned measurements but also an abundance of data from other related databases such as weather observation systems. The process is automated to help operators analyze multi-modal data that may be impossible to process manually due to the size and variety. We automatically preprocess multi-source data and learn a joint latent representation from collocated streamed, sparse, and high-dimensional data collected from Phasor Measurement Units and external weather data. Then we utilize multi-instance learning to predict events and discover precursors simultaneously without relying on post-mortem studies of fault signatures. We apply the proposed methodology to provide early predictions of faults in the U.S. Western Interconnection. AU-ROC of 0.94 is achieved in predicting events by utilizing information 5 hours before event time using season-specific models. We show how precursors can be extracted from multi-modal data and interpreted for predicted events.

INDEX TERMS Big data, weather, event detection, event precursors, machine learning, phasor measurement units, power system faults, smart grids, time series analysis.

I. INTRODUCTION

With the deployment of Phasor Measurement Units (PMUs) and the introduction of various other monitoring and recording systems, the amount of available data in the power systems has reached a challenging level for utilities [1]. PMUs provide measurements at 30-120 frames per second for voltage and current phasors, as well as frequency and the rate of change of frequency. Such streaming data is critical when conducting post-mortem analysis of power system disturbances and failures, as well as for system restoration and predictive analytics. To fully exploit the value of such large datasets, new machine learning techniques are introduced that can provide more automated and proactive responses. In recent years, extensive efforts have been shown to develop data models utilizing PMU and other data to enhance power system event detection and classification, leading to improved power system resilience [2], [3], [4], [5], [6], [7], [8], [9]. Utilizing data from different measuring systems (weather, PMU, etc.) can significantly improve the analysis of the current and future status of the power system. On the other hand, utilizing such measurements by power system operations is challenging due to the large size and hidden correlations not being easily captured and processed by operators.

Power systems data models characterizing events can be categorized as reactive or predictive. Reactive models are triggered after the actual events happen and can be used for post-mortem analysis of events to learn what may have gone wrong and determine the correct course of action in

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

the future. In such cases, the event has already happened, and data models assist in event post-mortem management, including event detection, classification, and intelligent alarm processing. In predictive models, data is used to anticipate the future status of the power system and allow power system operators the chance to react ahead of time to mitigate the impacts. We introduce a predictive model for anticipating power system faults ahead of time by identifying leading event precursors. This model utilizes multi-modal data in a sparse setting.

A. PROBLEM STATEMENT AND OBJECTIVES

Effective event prediction and precursor discovery in a power system relies on understanding the *internal* state of the system and any *external* factors interacting with the system and affecting its state. Our hypothesis is that event prediction and precursor identification accuracy can be improved by collocating and combining data representing the power systems' internal state with external factors affecting the system's operations. For this purpose, the internal power systems' state is represented by the PMU dataset, and the external factors by the weather affecting the power system.

The design of a solution to event prediction and precursor discovery is constrained by the sparsity and complexity of the data multi-modal data and the lack of specified leading indicators and precursors. That can inform the model if an event might happen or when. Consequently, we address the following research questions:

1. *how to predict power system failures* using sparse and multi-modal datasets representing the internal status and external factors affecting the power system, and
2. *how to design an automated prediction system for fault anticipation* to provide time for power system operators to react with optimal mitigation measures

These methods cover large spatial areas using sparsely mounted sensors providing streaming raw measurements. The objective is to feed complex and multi-modal data into automated systems to produce interpretable and actionable results. In summary, we provide a mathematical formulation to answer the following broad questions:

1. *when* (event prediction): when an event will happen.
2. *why* (precursors discovery): given a prediction, help power system operators identify specific examples or instances in large amounts of collected data, pointing to probable leading indicators of the provided prediction.

B. RELATED WORK

Several studies investigated PMU data for event detection and classification in power systems. An Extreme Learning Machine is used for a Fast variant of the Discrete S-Transform feature extraction [2]. A dimensionality reduction technique is introduced for event detection [3]. A wavelet-based method is also proposed to detect faults in PMU data [4]. Such methods rely on feature engineering and domain knowledge to design features. In automated feature learning and detection approach, Convolutional Neural Networks are used to detect faults from PMU data collected from the U.S. western interconnection [5]. Another method is also developed to detect and classify events using sparse PMU data [6]. Such methods use event signatures to detect events as they happen. Event prediction is studied in many other domains, such as the stock market, disease outbreaks, and crimes [7]. Methods used include regression, time series analysis, spatial analysis, and neural networks [7]. Example [8] uses a time series forecasting technique to predict events from simulated PMU data. A different approach, Collaborative Logistic Ensemble Classifier (CLEC), is introduced to classify events using weather forecasts [9].

While some of such approaches can predict events ahead of time using weather forecasts, very few, provide a methodology to identify event precursors. Some methods are developed to detect event precursors while at the same time estimating the likelihood of events. For instance, a methodology to predict labels for the events and at the same time, learn on the instance level is proposed in [10]. This methodology introduced the Group-Instance Cost Function (GICF) as a loss function that propagates information and distributes group labels to deep features. GCIF was expanded by introducing Nested Multi-Instance Learning (nMIL), where a nested data approach is used to distribute labels over multilayered instances. nMIL allowed labels to be predicted at the group level and nested levels [11]. nMIL shows promising results when used to predict societal events using public social media data [11].

C. PURPOSE AND NOVELTY

Event prediction, particularly fault prediction, is a critical task for power system control. It is an essential decision-making tool for control room operators that provides the time to plan mitigation measures to avoid or reduce the impact of forced outages. The power system monitoring data is dynamic, noisy, and exhibits many correlated factors. This burdens control room operators to analyze such correlated factors efficiently within an appropriate response time. Furthermore, correlating external data sources with internal datastores is inherently different from what the operators are tasked to do today. A large amount of such data makes the power system monitoring task even more challenging. Precursor analysis aims to identify important event anticipators before the actual event happens. Precursor analysis differs from event detection and classification since it focuses on what happened before the actual event, not the post-mortem event signature itself. Event precursors are important in understanding events that have often been discounted, dismissed, or never understood. Also, understanding precursors helps define near consequential events, which can help design better mitigation methods leading to more reliable power system operation.

The novelty reported in this paper is in jointly using sparse PMU data and sparse weather data for event prediction and fault precursor discovery. We provide a methodology to automatically preprocess large amounts of high-dimensional spatiotemporal data to predict faults and identify the precursors, which to the best of our knowledge, has not been reported before. We achieve that goal by introducing automated methods to learn latent embeddings from complex, multi-modal, and high-dimensional spatiotemporal data. The learned embeddings can be used to detect events and provide event precursors. Furthermore, we use multi-instance learning (MIL) to formulate a joint event prediction and precursor discovery model that does not rely on the pre-identification of specific leading indicators and precursors in the data. Rather, precursors are discovered as we develop and train the model.

D. CONTRIBUTION AND ADVANTAGES

There are several nuisances in event prediction and precursor discovery: (1) there are no standard out-of-the-box models for precursor discovery evaluation, compared to classification and forecasting where evaluation is more standardized, and (2) there are no labels for precursors, which is another challenging aspect for a machine learning model construction, and (3) there is no standard database syntax or semantics where the multi-modal datasets have standardized resolutions and representations.

In practical utility field settings, additional challenges include a lack of information on PMU location and power network topology for confidentiality reasons, big data paradox (lots of data, but few data for events of interest), bad data, temporal and spatial data dependency problems, and data model interpretability. The advantages of the approach we describe are as follows:

- tracking event prediction from raw streaming data allows proactive monitoring of the power system dynamic status progression compared to using triggered data for post-mortem reactive analysis.
- providing power system operators with an automated prediction tool for interpreting large amounts of multi-modal data allows for making timely mitigation decisions.
- defining a methodology for exploiting multi-modal data for fault prediction and precursor discovery in power systems using sparse PMU and weather data.
- utilizing automated techniques to preprocess large amounts of high-dimensional spatiotemporal data to reduce data dimensionality without extensive data wrangling and feature engineering.
- deploying multi-instance learning, where labels for precursors are not required, and information can be propagated through the data to predict labels and discover precursors.

E. PAPER ORGANIZATION

The remainder of this paper is organized as follows. Section II describes the methodology used for fault prediction and precursor discovery. Section III discussed the data used and preprocessing steps conducted. Section IV presents the experimental setup and results. Section V concludes the paper and section VI discusses future work. We provide two appendices to discuss the details of various data-related tasks. References are provided at the end.

II. EVENT PREDICTION AND PRECURSOR DISCOVERY

The intuition behind our formulation for power system event detection is similar to multi-instance learning (MIL). In MIL, the labels are assigned to the bag level where individual instances inside the bag do not have explicit labels. In MIL, trained classifiers aim to learn labels for the individual instances inside the bags. In a power system setting, for an event at time t, data for k hours before t is used and considered the bag. Data inside the bag can be further grouped depending on the modeling choice and application.

There are two important distinctions between what is presented in this paper and event detection:

1. *event signature at time t is not used*. In event detection, the actual signature at the time of the event is used to train a model to distinguish events from nonevents. Here, we are interested in predicting events ahead of time and identifying significant precursors. This task is performed *without* the use of actual event signatures. Formally we define the time period for an event at time t as $[t-k, t)$ to indicate the exclusion of time t.
2. *individual instances of data inside the bags are unlabeled*. In the current formulation of the problem, labels are assigned only to the bag level. Data instances within the bag are not assigned any labels at training time. Here, we *aim to identify precursors* from individual instances despite their lack of labels.

A. EVENT PREDICTION MODEL

This section introduces the basic information propagation learning paradigm utilizing multi-instance learning. Then two models are discussed, namely GICF and nMIL, where nMIL is considered an extension of GICF. Lastly, using these models, we introduce the precursor discovery methodology.

1) INFORMATION PROPAGATION THROUGH MULTI-INSTANCE LEARNING

For each event $\mathbb{Y}_n \in \mathcal{Y}$, where $\mathbb{Y}_n$ occurs at time t_n, we collect bags of data $\mathbb{B}_n \in \mathcal{B}$ where each bag $\mathbb{B}$ represent data in the time period $[t_n - k, t_n)$. Data representing $\mathbb{B}$ is collected from multiple sources (i.e., PMU, weather, etc.). In its basic representation, each bag $\mathbb{B}$ represents an unordered set of data instances $\{x_j\}$. A label is assigned for $\mathbb{Y}_n \in \{0, 1\}$ where 1 indicates that an event occurred at time t and 0 indicates that no events occurred at time t_n. As discussed in section II, actual event data at time t_i is not utilized in this study. When we discuss event precursors, we aim to identify individual data instances (x_j) which have great importance in predicting events $\mathbb{Y}_n \in \mathcal{Y}$.

We are given a training set $\mathfrak{D}$, which consists of a set of data bags $\mathbb{B}_n \in \mathcal{B}$ and their associated labels $\mathbb{Y}_n \in \mathcal{Y}$. $\mathfrak{D}$ is formally defined as:

$$\mathfrak{D} = \{(\mathbb{B}_n, \mathbb{Y}_n)\}_{n=1,\ldots,N} \tag{1}$$

and $N = |\mathcal{Y}|$. We train an unknown function $\mathbb{F}$ with parameters w, where:

$$\mathcal{F}\left(\mathbb{B}_n \,|\, w\right) \rightarrow \mathbb{Y}_n \forall n \in [1, N]. \tag{2}$$

$\mathbb{B}_n$ represents an unordered set of instances $\{x_j\}$. Each instance x_j represents a data vector collected for the time period $[t_n - k, t_n)$. In this setting, data vectors can be PMU data, weather data, or other data types. The $\mathcal{F}$ function is modeled as a logistic binary classifier on the instance level, and w is a learned weight vector. In logistic functions, $\mathcal{F}$ learns a probabilistic mapping learned from target labels $\mathbb{Y} \in \mathcal{Y}$. Each vector x_j is assigned a probability p_j that represents its relation to the target label $\mathbb{Y}$. Hence, p_j is defined as:

$$p_j = \sigma\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_j\right) = \frac{1}{1 + e^{-w^T x_j}} \tag{3}$$

where σ is the sigmoid function. Here, labels $\hat{y}_j$ can be inferred for p_j using pre-assigned thresholds. As discussed in section II, there are no ground truth labels for the instance level. The predicted label $\hat{\mathbb{Y}}_n \in [0, 1]$ is calculated by examining the probability value $\mathbb{P}_n$, where $\mathbb{P}_n$ is calculated by applying an aggregation function $\mathcal{A}$ on $p_j \forall x_j \in \mathbb{B}_n$. More formally:

$$\mathbb{P}_n = \mathcal{A}(\mathcal{F}\left(\mathbb{B}_n \,|\, w\right)) \tag{4}$$

The goal of the function $\mathcal{F}$ is to assign probabilities to individual instances x_j using *information propagation* from bag labels $\mathbb{Y}_n$ to predicted instance labels $\hat{y}_j$. This information propagation is controlled through a specialized cost function $\mathcal{J}(w)$. Furthermore, using the learned labels $\hat{y}_j$ for individual instances x_j, we train the classifier to infer labels for unseen bags as well as labels of the individual instances of the unseen bags. The cost function $\mathcal{J}(w)$ is the key tool to control information propagation from bag labels $\mathbb{Y}_n$ to instance labels $\hat{y}_j$ which constructs predicted bag labels $\hat{\mathbb{Y}}_n$. Sections II.A.2 and II.A.3 discuss two specific cost functions $\mathcal{J}(w)$.

2) GROUP-INSTANCE COST FUNCTION (GICF)

The first cost function $\mathcal{J}(w)$ we consider is the Group-Instance Cost Function (GICF), which combines bag costs and instance similarity costs. GICF is defined as the sum of bag level error and instance similarity costs. GCIF aims to distribute information from bag labels $\mathbb{Y}_n$ down to individual instances x_j. The assumption behind instance similarity costs is that similar instances (measured through a similarity function) should have relatively similar probabilities. Formally, equation 5 defines the cost function of GICF as:

$$\mathcal{J}_{GICF}\left(w\right) = \mathcal{J}\left(w\right)_{Bag} + \mathcal{J}\left(w\right)_{Instance} \tag{5}$$

where $\mathcal{J}\left(w\right)_{Bag}$ is defined as

$$\mathcal{J}\left(w\right)_{Bag} = \frac{\lambda}{N}\sum_{n=1}^{N}\Delta_1\left(\mathbb{Y}_n, \hat{\mathbb{Y}}_n\right) \tag{6}$$

and $\mathcal{J}\left(w\right)_{Instance}$ is defined as

$$\mathcal{J}\left(w\right)_{Instance} = \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B}\mathcal{K}\left(x_i, x_j\right)\Delta_2\left(\hat{y}_i, \hat{y}_j\right) \tag{7}$$

λ is a balancing factor between group penalty and instance penalty and $B = |\mathbb{B}_n|$. In instance level costs, $\mathcal{K}$ is used as a similarity function and Δ represents a non-negative penalty of the difference of values. In this formulation, we are controlling probabilities while at the same time examining similar instances through $\mathcal{K}$. Here $\mathcal{K}$ can be cosine similarity, radial basis function (RBF), or other similarity functions. To define Δ, we can use functions such as squared loss or log-loss. In conducted experiments, we use cosine similarity for $\mathcal{K}$ and squared loss for Δ_1, Δ_2. Squared loss is formally defined as:

$$\Delta_1\left(a, b\right) = \Delta_2\left(a, b\right) = (\mathrm{a} - \mathrm{b})^2 \tag{8}$$

where a and b are arbitrary values.

3) NESTED MULTI-INSTANCE LEARNING (NMIL)

The second cost function $\mathcal{J}(w)$ we consider is Nested Multi-Instance Learning (nMIL). This cost function extends GICF and introduces a nested data approach. The main advantage of introducing a nested approach is to account for temporal dependencies *within* the bag. A bag $\mathbb{B}$ in its basic definition represents an *unordered* set of data instances x_j. nMIL adjusts the definition of the bag from an unordered set of instances $\{x_j\}$ to a set of ordered temporal groups $\mathbb{B} = [\mathcal{X}_i]$, where $\mathcal{X}_i$ represents a temporal grouping (hourly) for a data collection $\mathcal{X}_i = \{x_{ij}\}$ and x_{ij} represents data instances at grouping time i for the $\mathrm{j} -$ th data source. Fig. 1 shows the data representation with temporal groups $\mathcal{X}_t$ and their associated probabilities P_i. Furthermore, Fig. 1 shows the data history and prediction leadtime for an event using the nested data modeling described earlier.

nMIL introduced a nested temporal cost, which accounts for temporal dependencies. This relies on the assumption that consecutive temporal groups $\mathcal{X}_t$ have similar probabilities P_i. The assumption is that the temporally adjacent instances will hold similar information and thus should have similar probabilities. nMIL penalizes temporal relations through a specific function g, which utilizes temporal dependency of data and ensures that the temporally adjacent instances hold similar information (measured through $\mathcal{K}$) and consequently have similar probabilities. In nMIL, g is defined as:

$$g = \mathcal{K}\left(\mathcal{X}_i, \mathcal{X}_{i-1}\right)\Delta_2\left(P_i, P_{i-1}\right) \tag{9}$$

where P_t (temporal grouping probabilities) is defined as an aggregate (using $\mathcal{A}$) of instance level probabilities for a temporal group $\mathcal{X}_i$, formally defined as:

$$P_i = \mathcal{A}(\mathcal{F}\left(x_{ij} \in \mathcal{X}_i \,|\, w\right)) \tag{10}$$

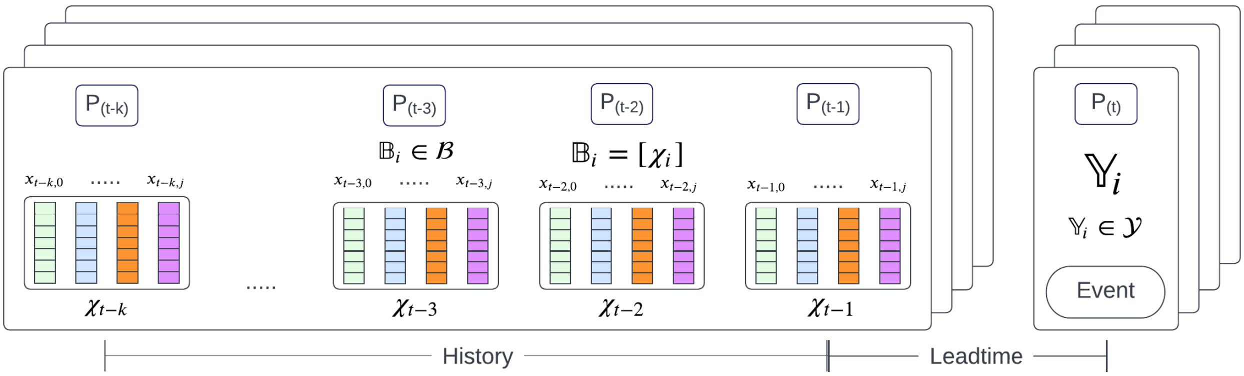
P(t-k)
P(t-3)
P(t-2)
P(t-1)
P(t)
$\mathbb{B}_i \in \mathcal{B}$
$\mathbb{B}_i = [\chi_i]$
$x_{t-k,0}$ ..... $x_{t-k,j}$
$x_{t-3,0}$ ..... $x_{t-3,j}$
$x_{t-2,0}$ ..... $x_{t-2,j}$
$x_{t-1,0}$ ..... $x_{t-1,j}$
χ_{t-k}
.....
χ_{t-3}
χ_{t-2}
χ_{t-1}
$\mathbb{Y}_i$
$\mathbb{Y}_i \in \mathcal{Y}$
Event
History
Leadtime

FIGURE 1. Data flow for event detection and precursor discovery. History represents how much data is used to predict an event. Leadtime represents the difference between the most recent data point used and the actual event time.

Since nMIL introduces temporal groups, the definition of $\mathbb{P}_n$ changes to:

$$\mathbb{P}_n = \mathcal{A}(P_i) \tag{11}$$

In addition, nMIL regularized loss by adding unsupervised hinge loss h and the L_2 regularization term $\gamma R(w)$. $\mathfrak{J}_{nMIL}\,(w)$ is a sum of bag loss, sequential instance loss, and regularization loss. Formally, equation 12 defines the cost function of nMIL.

$$\mathfrak{J}_{nMIL}\,(w) = \mathfrak{J}\,(w)_{Bag} + \mathfrak{J}\,(w)_{Instance} + \mathfrak{J}\,(w)_{Reg} \tag{12}$$

where $\mathfrak{J}\,(w)_{Bag}$, $\mathfrak{J}\,(w)_{Instance}$, $\mathfrak{J}\,(w)_{Reg}$ are defined as:

$$\mathfrak{J}\,(w)_{Bag} = \frac{\lambda}{N}\sum_{n=1}^{N} \Delta_3\left(\mathbb{Y}_n, \hat{\mathbb{Y}}_n\right) \tag{13}$$

$$\mathfrak{J}\,(w)_{Instance} = \frac{1}{\mathrm{N}}\sum_{i=1}^{B}\frac{1}{t}\sum_{t=1}^{T} g\left(\mathfrak{X}_{i,t}, \mathfrak{X}_{i,t-1}\right) \tag{14}$$

$$\mathfrak{J}\,(w)_{Reg} = \sum_{\substack{\mathbb{B}\,\in\,\mathfrak{B}\\ \mathfrak{X}_i\,\in\,\mathbb{B}\\ x_{ij}\,\in\,\mathfrak{X}_i}} \frac{1}{T}\sum_{i=1}^{T}\frac{1}{B}\sum_{j=1}^{B} h(x_{ij}, w) + \gamma R(w) \tag{15}$$

where h represents instance level hinge loss, here, h is formally defined in equation 16. m_0, p_0 are hyperparameters and sgn is the sign function.

$$h(x_{ij}, w) = \max(0, \mathrm{m}_0 - sgn\left(p_{ij} - p_0\right) w^T x_{ij}) \tag{16}$$

Lastly, Δ_3 is defined as:

$$\Delta_3\left(\mathbb{Y}_n, \hat{\mathbb{Y}}_n\right) = -(\mathbb{Y}_n \log\left(\mathbb{P}_n\right) + (1 - \mathbb{Y}_n)\log\left(1 - \mathbb{P}_n\right)) \tag{17}$$

The aggregate function $\mathcal{A}$ is defined as the arithmetic mean of the values. The loss functions are optimized using stochastic gradient descent with mini batch.

4) PRECURSOR DISCOVERY

Precursors are selected after an event is identified. In precursor discovery, we utilize nMIL since it accounts for temporal dependencies and has better regularization. To select precursors, we examine the probabilities p_{ij} of individual instances x_{ij} where $\hat{\mathbb{Y}}_n = 1$. Then we use a threshold η to determine if an individual instance is considered a precursor. If $p_{ij} \geq \eta$, then x_{ij} is a potential precursor.

III. DATA MANAGEMENT

Data management is an integral and important part of this approach. One of the main goals is to utilize multi-modal data sources for event prediction and precursor discovery without manual feature engineering. In this section, the two datasets used are described. Furthermore, the performed preprocessing steps on each dataset are discussed. Lastly, we discuss how we used deep representation learning to produce a unified lower dimensional and less noisy latent data representation.

A. DATA DESCRIPTION

This study utilizes two datasets, namely: PMU data and weather data.

1) PMU DATA

PMUs are sensors that monitor the power system sparsely. PMUs inputs are streams of samples taken from voltage and current signals describing the properties of the system. Usually, PMUs output the magnitude, angle, and frequency of the observed signals Through a transformation from sample to phasors. The output phasors are also post-transformed into a single positive sequence voltage or current. In addition, PMUs output the power system's fundamental frequency and rate of change of frequency readings. The data is synchronized in real-time between different locations using the timing clock signal obtained from Global Positioning System (GPS), which also reports precise UTC reference. PMUs report phasor data at high frequency, usually between 30Hz or 60Hz, and at an even higher rate.

PMU data is collected from 38 PMUs in the Western Interconnection in the United States. This dataset is proprietary and provided by the Department of Energy (DOE) for research purposes. This dataset is completely anonymized. to protect critical power system topology and infrastructure. The anonymization process included withholding all topological information, including PMU locations, which added significant challenges to this study.

PMUs collect PMU data at unknown locations for the years 2016 and 2017 and report data at 30Hz to 60Hz rates. Due to the high reporting rates and extended recording time, the resulting data file size for the two years of the U.S. Western Interconnection analyzed in our study is 6.87 Terabytes.

The PMU data used here has several challenges. The missing information about the power system topology (buses, voltage levels, etc.) and unknown PMU locations limited the ability to study events with specific spatiotemporal correlations. Since the PMU locations are undisclosed, we do not know the specific location or how far from the occurrence of an event a PMU is located.

For local events such as faults, some electrical signal properties caused by event occurrence are less prominent if measured at a distance from the event occurrence, but might be detected by considering multiple PMUs across vast geographical areas. We utilize this data property to learn a unified representation of PMU data. The representation learning process is introduced in Section II.C. The PMU dataset introduces a measurement sparsity problem, where it is known that the Western Interconnection has approximately 20,000 buses. The provided set of PMUs may have been unevenly collected covering certain areas much more densely than others. PMUs report multiple measurements simultaneously. In the current dataset, the data reporting rate is unified at 30Hz.

2) WEATHER DATA

Weather is an important factor when studying power system events. There are several sources of public weather data. We use weather data extracted from publicly available Automated Surface Observing Systems (ASOS) datasets [12]. ASOS is a joint effort of the National Weather Service (NWS), the Federal Aviation Administration (FAA), and the Department of Defense (DOD). ASOS contains weather data from more than 900 sites in the United States. The exact location of each weather station is known. ASOS data is reported on multiple resolutions, such as every 1-minute and 5-minutes. This resolution enables the reporting of rapidly changing weather conditions. ASOS stations report several basic measurements such as temperature, pressure, and sky conditions. We use the following weather measurements: air temperature in Fahrenheit, dew point temperature in Fahrenheit, wind direction in degrees from north, wind gust in knots, wind gust direction, visibility in miles, atmospheric pressure, precipitation total, and wind speed in knots.

B. DATA PREPROCESSING

As described in section III.A, we integrate information from two different datasets and each one comes with its reporting rate, dimensionality, and spatial settings. We unify the temporal resolution and address the missing spatial data in the following preprocessing steps.

1) GEOGRAPHICAL AREA SELECTION

For the case study, we consider the entire Western Interconnection service area which contains 136,000 miles of transmission lines, and spans 1.8 million square miles across 14 western U.S. states, in addition to parts of Canada and Mexico [13]. PMUs provide streaming data with sparse geographical area coverage. This characteristic of PMU data still allows for event prediction across sparse areas without the need to collect data from thousands of granular power system nodes (substations). For faults detection and precursors discovery we consider two important aspects of the selected region: (1) this area has diverse climatological topography, and (2) weather plays an important role in affecting power system stability [13]. Therefore, in addition to data from 38 PMUs described in Section A.1 we utilize a subset of weather data described in Section A.2 obtained from 529 weather stations distributed across the geographical area of the Western Interconnection.

2) SELECTING EVENTS OF INTEREST

Events are outages in the power system have effects on the stability of the power system as well as impacts on the power system users. The contributors of the PMU data accompanied the data with an event log. We utilize outage event set $\mathcal{E}$ from this event log provided for years 2016 and 2017 (those years are the years we have obtained PMU data for). For each event in the event set $\mathcal{E}$ we select a time period $[t_{i-k}, t_i)$ around it. When selecting events, we rely on the following criteria:

1. *selecting transmission line events*. Transmission line forced outages affect large geographical areas and can directly impact power system users.
2. *filtering out maintenance and planned outages*. Planned and maintenance events are not of interest since they are scheduled well ahead of time and do not need to be predicted
3. *selecting non-overlapping events*. This includes the period before the event t_i. Since we are interested in the period $[t_{i-k}, t_i)$ before the event, we ensure that the period $[t_{i-k}, t_i)$ which leads to the event does not contain any other events. This is done by cross-referencing time intervals of known events.

One important characteristic of the provided event logs is the complete lack of geographical information about the events. We were given neither the absolute nor relative locations of events. This limitation is introduced by the contributors of the PMU data, which presented a sparsity challenge for event detection and precursor discovery.

3) WEATHER DATA AGGREGATION

ASOS network has more than 900 stations in the United States, and we selected 529 of them for this study. This large amount of spatiotemporal data introduces challenges in predictive modeling. To mitigate the sparsity challenge for the weather data, we aggregate the weather data using the U.S. climatological divisions. Appendix I describes the process of aggregation. The climatological divisions divide the U.S. lower 48 states into geographical areas with similar weather conditions [14]. Using climatological division data proved useful in energy application research [14]. The National Oceanic and Atmospheric Administration (NOAA) publishes detailed maps of the climatological divisions [15]. We used these divisions to aggregate the weather stations' readings from weather stations in the same climatological divisions. Using this aggregation process, we lowered the number of spatial weather readings from 529 selected stations to 80 climatological divisions of interest. The 80 climatological divisions of interest were the divisions that intersect the Western Interconnection service area.

C. DATA REPRESENTATION LEARNING

Each data set (PMU data and weather data) has its own unique characteristics. This multi-modal data introduces several challenges to detection models: 1) different data reporting rates, and 2) the curse of dimensionality, where the high dimensionality, sparsity, and high reporting rates lower the effectiveness of downstream tasks and obscure relevant and important data characteristics 3) noisy and missing data remains a challenge even after the preliminary preprocessing steps. Representation learning is one common method used to handle data complexity issues.

Representation learning has been used across domains for many downstream tasks. Representation learning aims to learn latent low-dimensional representations of the original data, where the latent representations preserve the information within the original data and present it in a compact form, which eliminates the need to design and extract features from raw data manually. This is a useful property since feature engineering can be a time consuming task if done by subject matter experts. The deep learning models learn data-driven representations without requiring manual feature engineering. One category of representation learning is unsupervised representation learning, where latent representations are learned without the need for explicit labels or a specific downstream task. This type of representation learning is particularly useful in the datasets we are using since there are no labels on any of the individual data instances. Unsupervised representation learning is well-studied in computer vision and natural language processing domains [16].

Time series introduces additional problems to representation learning, such as high dimensionality, high frequency, and non-stationary. Temporal Neighborhood Encoding (TNC) was introduced to learn unsupervised representation for non-stationary and multivariate time series [16]. TNC utilizes stationary properties within temporal neighborhoods to define a distribution of similar windows. TNC utilizes statistical tests to define the boundaries of neighborhoods. To overcome potential bias when sampling negative data, TNC utilizes Positive Unlabeled Learning.

TNC learns a latent representation for each window W_q of data of length δ centered around time q. TNC learns an encoder $Enc\left(\cdot\right)$:

$$Enc\left(W_q^{[d\times\delta]}\right) \rightarrow \tilde{W}_q^{[\tilde{d}\times 1]} \tag{18}$$

which maps a window of size $[d \times \delta]$ (d is the dimensionality of the time series) to a latent representation $\tilde{W}$ of size $[\tilde{d} \times 1]$. The encoding size $\tilde{d}$ is pre-set before training. After encoding, TNC learns a discriminator $Des\left(\cdot\right)$ that estimates the probability of two encodings $\tilde{W}_1, \tilde{W}_2$ to belong to the same neighborhood. TNC uses a multi-headed binary classifier for $Des\left(\cdot\right)$. For the $Enc\left(\cdot\right)$ specifics, TNC is agnostic to its design. We use a Recurrent Neural Network (RNN) for the encoder with a multi-layer Gated Recurrent Unit (GRU). We use two layers with a hidden size of 100. Since TNC is an unsupervised representation learning model, the quality of the encodings is assessed by examining the performance of $Des\left(\cdot\right)$. The performance of the $Des\left(\cdot\right)$ is examined separately from downstream tasks. Each dataset is embedded separately into a new feature space. When extracting the final representation, the $Des\left(\cdot\right)$ is not used as the $Des\left(\cdot\right)$ is just used to control the learning process. The final embeddings are extracted by using the trained $Enc\left(\cdot\right)$ using the penalties imposed by the $Des\left(\cdot\right)$.

1) EMBEDDING PMU DATA

PMU data consists of three time series measurements (V, I, f) measured from 38 PMUs. The embedding is done by measurement (feature). For each measurement $\{V, I, f\}$, we use a different TNC model to learn a new representation of it. For each input window W, of the model size is $[d \times \delta]$ where $d = 38$. Embedding size $\tilde{d}$ is set at 60, which proved effective for different applications [16] and showed good performance using this data. δ can be adjusted to control the window being represented, but there are caveats when choosing δ. If δ is too big, the window crosses neighborhoods and the representation is not yielding good results for the $Des\left(\cdot\right)$. On the other hand, a very small δ is very sensitive to local changes in the time series and is not producing good results for the $Des\left(\cdot\right)$. After consulting with domain experts, we designed the embeddings to represent 1 hour of data. The final embedding is of size $[60 \times 1]$ for each 1 hour of data for one feature.

A major issue faced when embedding the PMU data is its sheer size. At the original reporting rate (30Hz), each hour of data for one signal has more than 4 million data points for 38 PMUs (30Hz × 3600sec × 38 PMUS). This large size causes two issues: (1) embedding size per hour is not rendering good accuracy for the $Des\left(\cdot\right)$, even with parameter tuning and more complex $Enc\left(\cdot\right)$, and (2) run times for the embedding model are not feasible, even when using high-end scientific cloud computing equipment (4x NVIDIA Tesla V100 GPUs with NVlink2, code is utilizing the GPUs when running, with 512GB of RAM). Using this equipment, the run time took around a week per experiment when doing parameter tuning, which is infeasible when developing models. To resolve this issue, we downsampled the data. Previous studies [5], [6], [17] show that PMU data can be downsampled for various applications without sacrificing the performance of downstream tasks. We use the Largest Triangle Three Buckets (LTTB) [18] to downsample PMU data. LTTB showed effectiveness in downsampling in multiple applications without losing any visual or inherent data characteristics. Using LTTB, we performed two downsampling experiments: downsampling to 60 samples/min and downsampling to 1 sample/min. Appendix II discusses examples of downsampled PMU data, and how effective LTTB is in downsampling data without losing its characteristics.

Visually, both variations of the downsampled datasets are satisfactory. When applying representation learning, the best performing dataset was the 1 sample/min. This was shown by examining the accuracy of the $Des(\cdot)$ and stability through the execution epochs. Fig. 2 shows the embedding performance of the two datasets. Fig. 2 diagram 1 shows the embedding performance using 1 sample/min for f. The graph shows that we had stable accuracy through the epochs and no underfitting. Diagram 2 in Fig. 2 shows the embedding performance for 60 samples/min, which had unstable performance. Both experiments used the same number of epochs (150). The behavior shown in Fig. 2 is compatible with the expected behavior from TNC and embedding models in general. TNC focuses on extracting compressed latent representations out of the raw data. Suppose TNC is applied to a higher frequency of data. In that case, the fundamental latent representations will be obscured by noise and local data fluctuations, which embeddings are trying to overcome.

2) EMBEDDING WEATHER DATA

Weather data was embedded similarly to PMU data. Each weather measurement is embedded separately, which results in each hour of data being represented as one embedding vector ($\tilde{d} = 60$). In the dataset we use, weather data is reported every 5 mins, which results in reasonable TNC model learning times. The final embedding is of size $[60 \times 1]$ for 1 hour of data for each weather measurement.

The final dataset is grouped by the hour from the two data sources. The total number of embedding vectors from all datasets is 12, with each hour represented by $[60 \times 12]$ matrix.

IV. EXPERIMENTAL EVALUATION

A. PREDICTION MODELS

We used three models, two of which are GICF and nMIL, which are introduced in sections II.A.2 and II.A.3, respectively. In addition, Logistic Regression (LR) is used as a baseline model. For comparison with the other models, we use LR with a Single Instance Learning (SIL) approach to learning (LR-SIL). In this approach, we distribute bag labels $\mathbb{Y}_n$ to instances x_j and train the LR model. When calculating bag labels, we use the $\mathcal{A}$ to aggregate p_j for bags and infer labels $\hat{\mathbb{Y}}_n$.

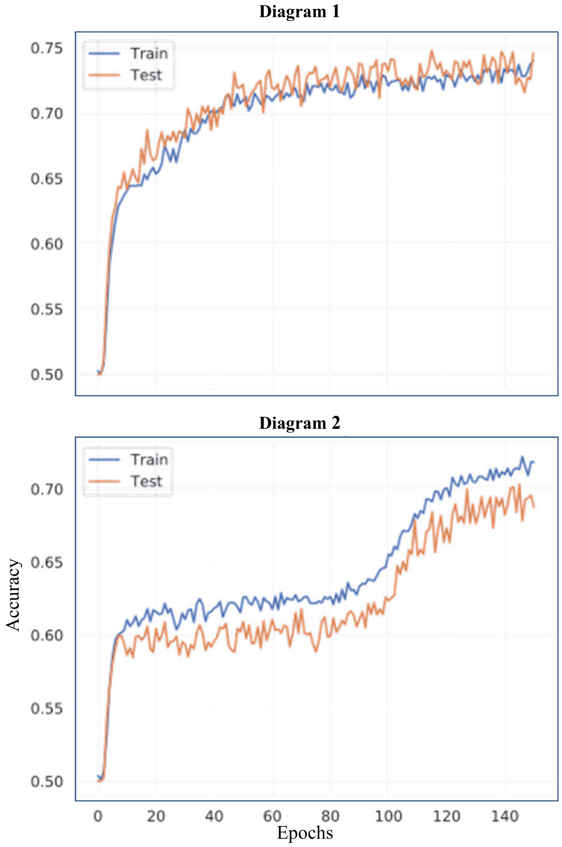

FIGURE 2. Performance of TNC embedding for f using 1 sample/min (Diagram 1) and 60 samples/min (Diagram 2).

B. EXPERIMENTAL SETUP

1) DATA SPLIT

The models are evaluated using two years of data from 2016 and 2017. All data from the two sources are preprocessed as described in section III.B. The data is split into training and testing datasets. The dataset is split temporally to capture seasonal patterns throughout the year and ensure the testing is performed on equal ground. We introduce two modeling scenarios, namely the *global model* and the *seasonal model*. In the global model, the data for the year 2016 is used for training, and the 2017 data is used for testing. In the seasonal model, four different models are trained and evaluated. Each of the four seasonal models is specialized for a specific season (Summer, Fall, Winter, Spring), where for example the Spring of 2016 is used for training, and the Spring of 2017 is used for testing. In the seasonal model, the season of 2016 is used for training, and the same season of 2017 is used for testing. In both seasonal and global models, the same geographical area of the Western Interconnection is used. The reason behind the seasonal models is to determine if season-specific models can further improve detection results in such a sparse setting. The number of positive cases corresponding to known power system events per the event log of the two years is 179. We supplemented these data with 204 negative cases where there are no events. The negative cases were selected by cross-referencing known event logs and ensuring that the periods do not contain or overlap with any known event. The negative events were selected from both years.

2) MODEL PARAMETERS

k is set to 5 hours and used to extract the time periods used for training and testing. The value of k is chosen after consulting with SMEs. The embedding dimension size $\tilde{d}$ is set to 60. Using guided hyperparameter tuning, λ is set to 2, 60 for nMIL and GICF, respectively, the batch size is 25, and the learning rate for mini batch is 0.01. For nMIL regularization m_0 and p_0 is kept at the default 0.5 and γ is set to 0.25. Except for m_0, p_0, all other hyperparameters are set through experiments. For LR-SIL, a linear solver with L2 penalty and C=0.0001 is used.

3) EVALUATION METRICS

We report precision and recall to evaluate the correctness of the used models. One of the main goals is to help power system operators navigate large amounts of data and for the proposed methodology to behave like an early warning system. By reporting precision and recall, we see how the model is measuring false positives and false negatives. The relevance of the probabilities of the events to the predictions is measured by reporting the Area Under the Receiver Operating Characteristic curve (AU-ROC). Since we have a slight imbalance in the data, we also reported Area Under the Precision-Recall Curve (AU-PRC). AU-ROC and AU-PRC provide a measure of robustness and tuning flexibility of the models, which are important in practical settings.

C. EVENT DETECTION PERFORMANCE

The performance of the event prediction is evaluated using the data split and metrics discussed in sections IV.B.1 and IV.B.2. This section discusses the performance of event detection using the seasonal and global models, as well as how early the model can detect events and how much data is needed. The last part of this subsection discusses the precursor discovery results and shows how they can be applied in practice. The results from the global model are shown in Table 1. During the experiments, we faced issues with the sensitivity of the detection models. To assess this, we conducted experiments using all the features we have and a search for best performing groups of features. Table 1 shows the prediction results of the best performing feature groups. Group 1 (G1) represents PMU voltage, wind gust, pressure, and precipitation. G2 represents wind gust, precipitation, air temperature, PMU voltage, PMU current, and PMU frequency. G3 represents all weather parameters considered and PMU voltage. We believe that the results presented here are in line with the behaviors expected from logistic-based models. Logistic models tend to overfit when there are correlations between input parameters. G1 represents a small subgroup of available features. The original set of features has inherent correlations between them. For example, system frequency is not a measured feature but rather a calculated feature based on voltage and current. In addition, weather measurements such as wind speed, wind gust, and pressure are correlated. In this application, using methodologies such as Principal Component Analysis for feature reduction would not be helpfull since learned features cannot be explained and related to precursor analysis. Table 1 shows that the best-performing global model was nMIL, described in section II.3, using feature group G2 described in Section III (though all groups that combine weather and PMU data have comparable results). The combination of power system information (PMU data) and weather data provided the most information. The high Recall shows that this model can positively detect events, which can help operators identify events. In addition to the discussed groups, in Table 1 we also show the performance of uni-modal data demonstrated by just using PMU data (frequency f or the voltage V) or just using weather data (W).

TABLE 1. Models performance for global model.

Model	Features	AU-ROC	AU-PRC	Precision	Recall
nMIL	G1	0.721	0.762	0.763	0.632
	G2	**0.728**	0.795	**0.849**	0.530
	G3	0.729	**0.800**	0.777	**0.624**
	f	0.593	0.674	0.690	0.419
	V	0.665	0.735	0.733	0.539
	W	0.665	0.665	0.765	0.640
GICF	G1	0.705	0.752	0.742	0.615
	G2	**0.709**	**0.758**	0.761	0.598
	G3	0.690	0.760	**0.768**	**0.624**
	f	0.588	0.678	0.734	0.496
	V	0.665	0.749	0.706	0.410
	W	0.683	0.746	0.713	0.577
LR-SIL	G1	0.642	0.759	**0.849**	0.239
	G2	0.656	0.762	0.800	0.410
	G3	**0.670**	**0.772**	0.810	0.291
	f	0.522	0.599	0.364	0.034
	V	0.634	0.689	0.750	0.231
	W	0.648	0.759	0.753	**0.470**

Bold values represent the best for each model. Bold and underlined values represent the best overall.

Table 2 shows results using the seasonal setup for the best performing model from Table 1 (nMIL). Here, four different seasonal models were trained and tested. From Table 2, seasonal models significantly improved the performance compared to the global model. For example, the Fall and Winter models using G1 showed 31% and 27% improvement respectively in AU-ROC compared to the global model for the same feature group. Fig. 3 shows the

ROC for nMIL of the Fall model. Seasonal models show significant improvement in AU-PRC, and high scores for precision and recall show that the model has higher precision and higher recall. This improvement in results can be explained by reduced heterogeneity of weather data conditions within seasons, where embedding and detection models were able to better capture the season-specific behaviors. Furthermore, the weather has much more effect on grid stability in the Winter months, and power grid events can be more predictable.

Table 1 and Table 2 show that adding external data sets (such as weather data) can improve models' detection performance compared to using only PMU data. Furthermore and across all models, we saw gains in performance by using multi-modal data (G1, G2, G3) compared to uni-modal data (V, *f* , W).

TABLE 2. Performance for seasonal model.

Season	Features	AU-ROC	AU-PRC	Precision	Recall
Winter	G1	**0.922**	**0.898**	0.609	**0.933**
	G2	0.850	0.738	0.565	0.867
	G3	0.881	0.803	**0.769**	0.667
	f	0.733	0.679	0.588	0.667
	V	0.814	0.687	0.500	0.933
	W	0.833	0.671	0.700	0.933
Spring	G1	0.870	0.929	**0.923**	0.783
	G2	**0.884**	**0.933**	0.919	0.739
	G3	0.866	0.925	0.861	**0.804**
	f	0.593	0.778	0.659	0.630
	V	0.791	0.865	0.906	0.630
	W	0.853	0.922	0.857	0.782
Summer	G1	0.705	0.735	0.813	0.406
	G2	**0.818**	**0.888**	0.821	**0.718**
	G3	0.793	0.845	0.815	0.688
	f	0.635	0.695	0.615	0.250
	V	0.780	0.857	**0.909**	0.313
	W	0.779	0.772	0.789	0.468
Fall	G1	**0.944**	**0.985**	0.947	0.750
	G2	0.847	0.957	0.933	0.583
	G3	0.840	0.960	0.947	0.750
	f	0.674	0.901	0.833	0.625
	V	0.743	0.940	0.826	**0.792**
	W	0.930	0.984	**1.000**	0.291

Bold values represent the best for each season. Bold and underlined values represent the best overall.

1) HOW EARLY CAN WE PREDICT?

All prediction results reported in the previous subsections are obtained with a leadtime (hours ahead of the event) of 1 hour, where we considered 5 hours before the event to detect an event in the next hour. Fig. 4 shows the area under the AU-ROC for different leadtimes using nMIL on G1 for the Fall model. As expected, the closer we get to the actual event, the accuracy of event detection gets higher. We can see that even at 3-4 hours ahead, we can still obtain useful predictions out of the developed model. This is potentially deployable since early detection of events can be useful for grid operators if the lead-time is sufficient to allow taking early proactive actions before events cause wide disruptions to the grid.

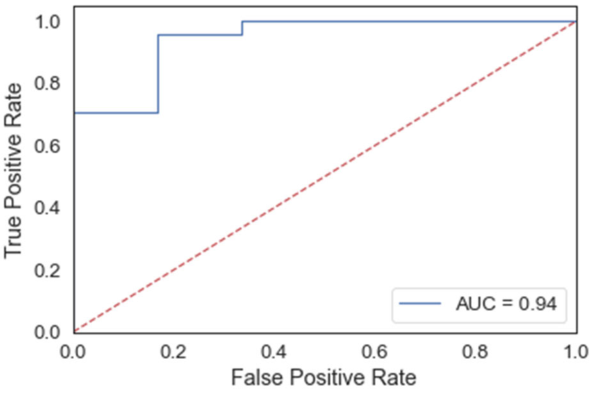

FIGURE 3. Receiver Operator Curves for nMIL.

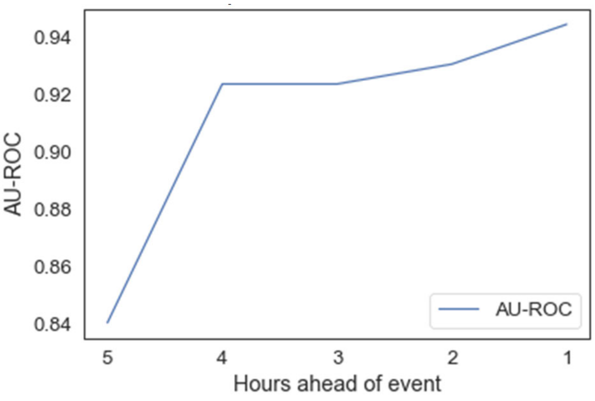

FIGURE 4. Receiver Operator Curves for nMIL.

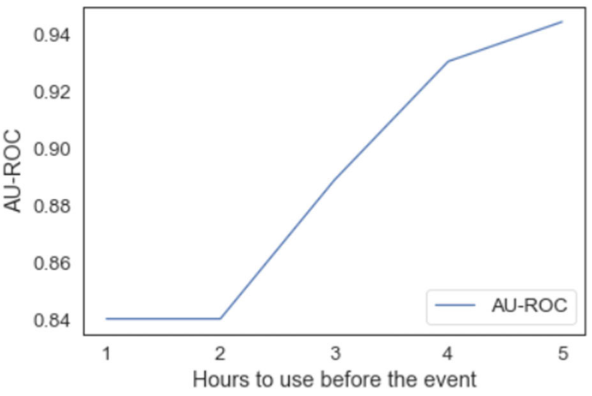

FIGURE 5. Area Under the Receiver Operator Curve for multiple *k*.

2) HOW MUCH DATA IS NEEDED?

In the experimental setup reported in the previous section, we have considered *k* to be 5 hours. In this section, we discuss the effects on the event performance if different *k* were used. Fig. 5 shows the performance of nMIL on G1 for the Fall model using multiple *k* values. Fig. 5 shows that increasing *k* from 1 to 5 hours contributes to a much more stable prediction. As we further increased *k*, AU-ROC increased.

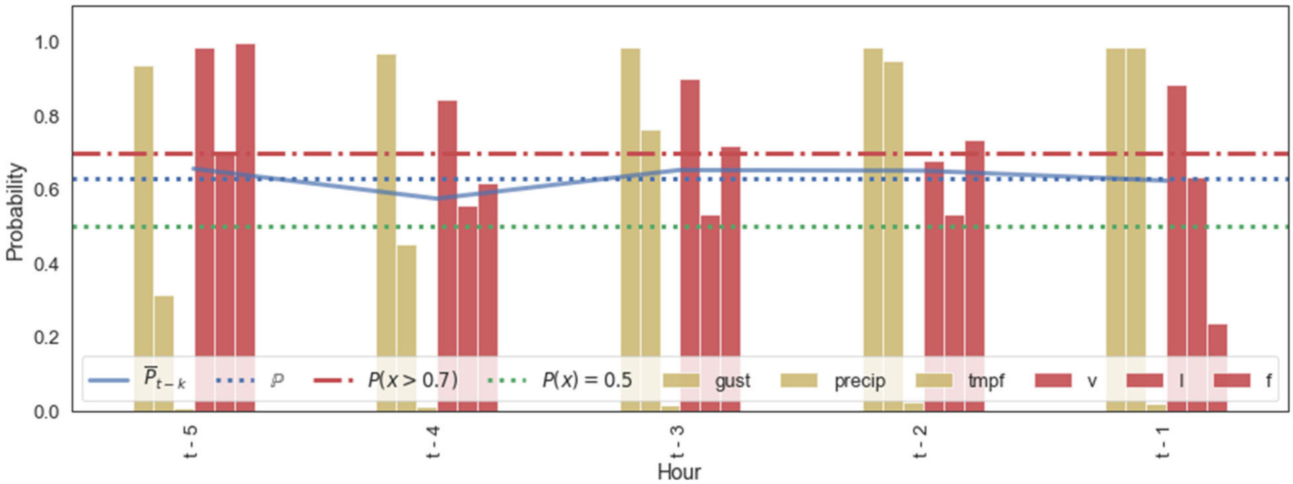

FIGURE 6. Probabilities of detection for an actual event using nMIL with G1 (weather parameters are shown in green and PMU data are shown in red). The figure shows how precursors importance's are increasing when the time approaches the event.

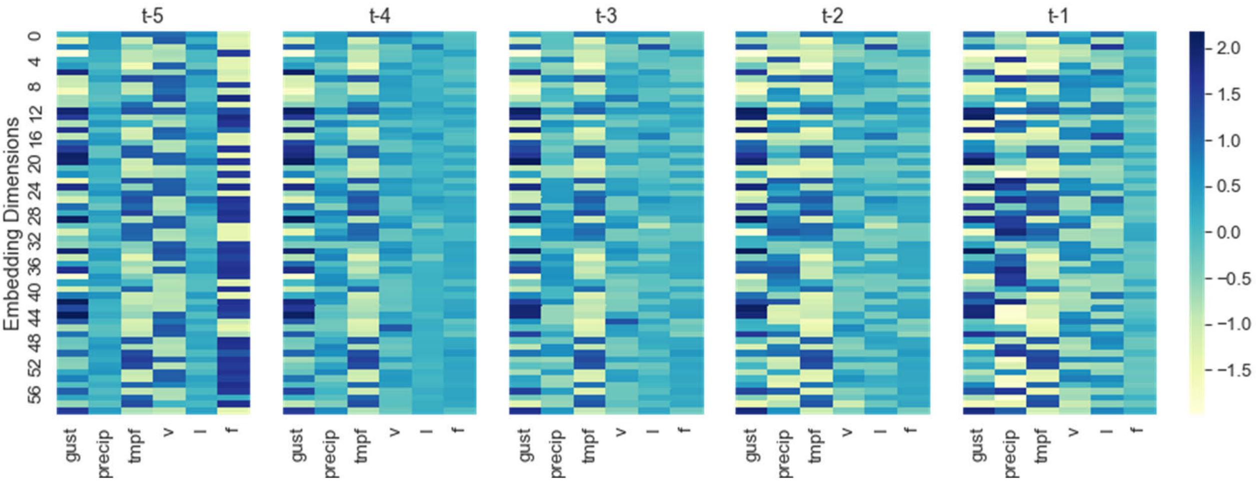

FIGURE 7. Embedded G1 data for the event predicted and shown in Fig. 7. We can see that the embedding absolute value change as time moves towards the event time (*t*). Visually predicting events from embedded data is not a trivial task.

This can be explained since larger k allows the model to capture temporal differences and trends. We can see that there is not much difference between 1 and 2 hours, but performance improves as we increase k. The benefit of longer prediction horizons (larger k) is that it allows for earlier event detection. This enables power system operators to prepare earlier for high-risk events.

3) PRECURSOR DISCOVERY

The precursor discovery uses a threshold η to determine the significance of an individual instance x_{ij} towards the predicted label $\hat{\mathbb{Y}}_n$. For precursor discovery, we use the probabilities produced by nMIL only. This choice is based on the way the nMIL cost function is constructed. In GICF, there are no temporal (hourly) groupings, and the temporal aspect is disregarded. When discussing precursors, we are interested in identifying the precursors in a temporal setting. To demonstrate how precursors are used for a specific event, Fig 6. shows the probabilities obtained for a correctly detected event using nMIL from the feature group G1 for the Fall model. In Fig. 6, weather parameters are shown in green and PMU data are shown in red. Each group of bars represents an hour. We selected $\eta = 0.7$ (which is represented as the red dashed line). As the event is predicted using the model, in Fig. 6, we can see that as we approach the event time (t), wind gust and precipitation start contributing to the detection. Since voltage probabilities increased significantly two hours before the event, we can indicate that PMU voltage was an indicator ahead of time. This progression of probabilities can be used by power system operators as an early warning of the event. In this way we are not just predicting events, but also providing an explainable prediction, which can be used to plan outage mitigation ahead of time.

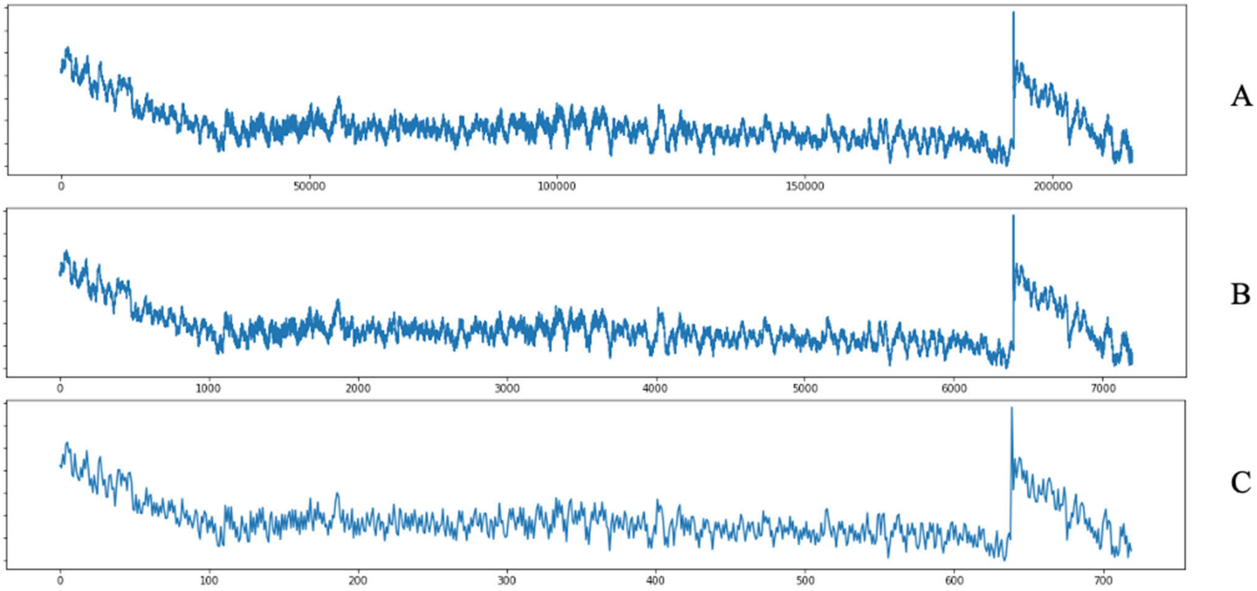

FIGURE 8. Examples of downsampling using LTTB. Graph A is original data. Graphs B and C are 60samples/min and 1sample/min respectively. We can see that graph C shows all the main characteristics using a fraction of original data size. All graphs' y-axises have similar data ranges, actual y-axis values are not shown for data privacy reasons.

The provided methodology introduces direct and explainable links between the prediction and the used data. In comparison, using the raw data or the learned representations can be challenging. The representation learning described in section III.C mapped complex and high-dimensional data to a latent space. As discussed, Fig. 6 shows how the final probabilities of the model can be interpreted. In Fig. 7, we show what the embedded data for the same event looks like. Fig. 7 shows the data used as input to the model. If we compare the two figures, we can see the changes in data (color changes as the time approaches t) that correlates with the probabilities. A one-to-one mapping between probabilities and embedded data does not necessarily exist due to the complexity of the detection model. However, a user trying to interpret the predicated event can examine the probabilities and the embedded data simultaneously and gain a better understanding of the model's behavior.

V. CONCLUSION

In this study, we:

- presented a methodology to use sparse multi-modal data for event prediction and precursor discovery designed to aid power system operators in efficiently using large amounts of multi-modal data.
- show how to use streamed and sparse PMU data and external data sources to predict power system events without relying on triggered data or tracking thousands of granular nodes in the power system.
- introduce a paradigm to use high-dimensional multi-modal data, where data is aggregated, and automatically preprocessed, a new latent space is learned, and data is mapped to it.
- introduce two settings of building and training detection models, where one uses a full year and the other uses seasons for training and testing.
- demonstrate how precursor discovery can be achieved at the same time as event detection, the models are able to predict events accurately with the best AU-ROC of 0.94 in a 1-year out of sample data.
- detect events ahead of time with acceptable performance, which can provide the power system operators with interpretable outputs in the form of probabilities for the precursors and trackable latent space data.

VI. FUTURE WORK

One of the main limiting factors of this study is the lack of power system topology, PMU locations, and any other information about the power system status. We believe that by utilizing more information such as locations of PMUs, and prediction accuracy, and the richness of precursors and provide localized info about the predicted failures. We also suggest a study of longer time horizons before the events, which requires models that can distinguish between short- and long-term spatiotemporal relations to the event.

APPENDIX I: U.S. CLIMATOLOGICAL REGIONS AGGREGATION

As discussed in section III.B.3, the weather data is aggregated for the Western Interconnection region, where many Automated Surface Observing Systems (ASOS) weather stations exist. Using information obtained at all weather stations can introduce data redundancy in sparse settings, especially if the weather stations are measuring areas of similar climates. To mitigate this, weather stations are aggregated based on their locations in climatological divisions. This aggregation aims to capture the weather in a smaller spatial region without mixing areas of different climates. To achieve this, we relied on the climatological division discussed earlier. The aggregation process is performed as follows:

A. WEATHER STATION SELECTION

To select the weather stations of interest, we overlaid the Western Interconnection region maps with the locations of each weather station. This is possible since weather stations' have their latitude-longitude known. Stations that don't rely on the Western Interconnection service area were excluded using the overlaid maps. The same methodology was used to select weather divisions of interest.

B. AGGREGATING WEATHER DATA BY CLIMATE DIVISIONS

One of the ways the National Oceanic and Atmospheric Administration (NOAA) reports the climate divisions is through a list of zip codes [15]. In this list, each climate division has the list of county Federal Information Processing Standard Publication (FIPS) codes associated with it. Using this dataset, we mapped each location (latitude-longitude) of the weather station to FIPS codes using the U.S. government Census Geocoding Services [19]. Then the intersection of weather divisions and weather stations is determined through common FIPS codes. The last step is to aggregate weather measurements from different stations. This is achieved by averaging all-weather measurements except for precipitation, which was summed. This approach followed standard practices for NOAA datasets.

APPENDIX II: LTTB DOWNSAMPLING EXAMPLES

As discussed in section III.C.1, PMU data is downsampled to reduce data noise and processing times. Largest Triangle Three Buckets (LTTB) method is used to downsample PMU data. This appendix shows an example of original data compared to downsampled data. Graph A in Fig. 8 shows the original data. Original data is two hours long for one signal for one PMU, which at 30Hz will have 216,000 data points (30Hz $\times$ 3600seconds/hour $\times$ 2 hours $=$ 216,000). Graphs B and C in Fig. 8 show the 60 samples/min and 1 sample/min, respectively. Graphs B and C still have a strong resemblance to the original data, but Fig. 8 graph C has 0.3% of the original size (720 data points), and Fig. 8 graph B has 3.3% of the original size (7200 data points). LTTB captured all data's main characteristics and values changes but with a fraction of size.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

[1] North American Synchro Phasor Initiative, "Data mining techniques and tools for synchrophasor data," NASPI, Houston, TX, USA, Tech. Rep. NASPI-2018-TT-007, Jan. 2019.

[2] M. Biswal, S. M. Brahma, and H. Cao, "Supervisory protection and automated event diagnosis using PMU data," *IEEE Trans. Power Del.*, vol. 31, no. 4, pp. 1855–1863, Jan. 2016.

[3] L. Xie, Y. Chen, and P. R. Kumar, "Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis," *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2784–2794, Nov. 2014.

[4] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, and Y.-J. Shin, "Wavelet-based event detection method using PMU data," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1154–1162, May 2017.

[5] M. Alqudah, M. Pavlovski, T. Dokic, M. Kezunovic, Y. Hu, and Z. Obradovic, "Fault detection utilizing convolution neural network on timeseries synchrophasor data from phasor measurement units," *IEEE Trans. Power Syst.*, vol. 37, no. 5, pp. 3434–3442, Sep. 2022, doi: 10.1109/TPWRS.2021.3135336.

[6] M. Pavlovski, M. Alqudah, T. Dokic, A. A. Hai, M. Kezunovic, and Z. Obradovic, "Hierarchical convolutional neural networks for event classification on PMU measurements," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art no. 2514813, doi: 10.1109/TIM.2021.3115583.

[7] L. Zhao, "Event prediction in the big data era: A systematic survey," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–37, 2021.

[8] M. Barati, "Faster than real-time prediction of disruptions in power grids using PMU: Gated recurrent unit approach," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Feb. 2019, pp. 1–5, doi: 10.1109/ISGT.2019.8791625.

[9] T. Dokic and M. Pavlovski, "Spatially aware ensemble-based learning to predict weather-related outages in transmission," in *Proc. Hawaii Int. Conf. Syst. Sci. (HICSS)*, Maui, Hawaii, Jan. 2019, pp. 3484–3493.

[10] D. Kotzias, M. Denil, N. de Freitas, and P. Smyth, "From group to individual labels using deep features," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 597–606.

[11] Y. Ning, S. Muthiah, H. Rangwala, and N. Ramakrishnan, "Modeling precursors for event forecasting via nested multi-instance learning," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1095–1104.

[12] *Iowa Environmental Mesonet*, ASOS Network, Iowa State University, Ames, IA, USA, May 2022. [Online]. Available: https://mesonet.agron.iastate.edu/ASOS/

[13] Western Interconnection, *Western Electricity Coordinating Council*. Accessed: May 31, 2022. [Online]. Available: https://www.wecc.org/epubs/StateOfTheInterconnection/Pages/Western-Interconnection.aspx

[14] N. B. Guttman and R. G. Quayle, "A historical perspective of U.S. climate divisions," *Bull. Amer. Meteorological Soc.*, vol. 77, no. 2, pp. 293–304, 1996.

[15] The National Oceanic and Atmospheric Administration (NOAA). *U.S. Climate Division Dataset Maps*. Accessed: May 31, 2022. [Online]. Available: https://psl.noaa.gov/data/usclimdivs/

[16] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," 2021, *arXiv:2106.00750*.

[17] A. A. Hai, T. Dokic, M. Pavlovski, T. Mohamed, D. Saranovic, M. Alqudah, M. Kezunovic, and Z. Obradovic, "Transfer learning for event detection from PMU measurements with scarce labels," *IEEE Access*, vol. 9, pp. 127420–127432, 2021, doi: 10.1109/ACCESS.2021.3111727.

[18] S. Steinarsson, "Downsampling time series for visual representation," M.S. thesis, Dept. Ind. Eng., Mech. Eng. Comput. Sci., School Eng. Natural Sci., Univ. Iceland, Reykjavik, Iceland, 2013.

[19] U.S. Federal Communications Commission. *Enterprise Area API*. Accessed: May 31, 2022. [Online]. Available: https://geo.fcc.gov/api/census/

MOHAMMAD ALQUDAH (Student Member, IEEE) received the B.Sc. degree from the Jordan University of Science and Technology, Irbid, Jordan, in 2012, and the M.S. degree from Binghamton University, Binghamton, NY, USA, in 2015. He is currently pursuing the Ph.D. degree in computer and information sciences with the Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, PA, USA. His research interests include machine learning for spatiotemporal and graph data and the application of machine learning in smart grids.

MLADEN KEZUNOVIC (Life Fellow, IEEE) has been with Texas A&M University, College Station, TX, USA, since 1986, where he is currently a Regents Professor, a Eugene E. Webb Professor, and the Site Director of "Power Engineering Research Center" Consortium. For over 30 years, he has been the Principal Consultant of XpertPower Associates, a consulting firm specializing in power systems data analytics. He has authored over 600 papers, given over 120 seminars, invited lectures, short courses, and consulted for over 50 companies worldwide. His research interests include protective relaying, automated power system disturbance analysis, computational intelligence, data analytics, and smart grids. He is a CIGRE Fellow, an Honorary, and a Distinguished Member. He is also a member of the National Academy of Engineering. He is a Registered Professional Engineer in Texas.

ZORAN OBRADOVIC (Senior Member, IEEE) is currently a Distinguished Professor and the Center Director of Temple University, an Academician at the Academia Europaea (the Academy of Europe), and a Foreign Academician at the Serbian Academy of Sciences and Arts. He mentored 45 postdoctoral fellows and Ph.D. students, many of whom have independent research careers at academic institutions and industrial research laboratories. His research results were published at about 400 data science and complex networks articles addressing challenges related to big, heterogeneous, spatial and temporal data analytics motivated by applications in healthcare management, power systems, earth, and social sciences. He was the general chair, a program chair, and a track chair of 11 international conferences. He is the Editor-in-Chief of the *Big Data* journal and the Steering Committee Chair of the SIAM Data Mining Conference. He is also an editorial board member at 13 journals.