IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 31, 2021, accepted September 8, 2021, date of publication September 10, 2021,
date of current version September 21, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3111727*

# Transfer Learning for Event Detection From PMU Measurements With Scarce Labels

**AMEEN ABDEL HAI[1], (Student Member, IEEE), TATJANA DOKIC[2], (Member, IEEE), MARTIN PAVLOVSKI[1], (Member, IEEE), TAIF MOHAMED[2], (Graduate Student Member, IEEE), DANIEL SARANOVIC[1], 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** Event detection in electrical grids is a challenging problem for machine learning methods due to spatiotemporally nonstationary systems and the inability to automate event labeling in high-volume data such as PMU measurements. As a result, the existing historical event logs created manually do not correlate well with the corresponding PMU measurements due to scarce and temporally imprecise labels. Trying to overcome this problem by extending event logs to a complete set of labeled events is very costly and often infeasible. We focused on utilizing a transfer learning model to reduce the need for additional data labeling by leveraging some labeled data instances available from a small number of well-defined event detection task. To demonstrate the feasibility, we tested our approach on a large dataset collected by 38 PMUs from the Western Interconnection of the U.S.A. over two years. The model evaluation performed based on varying percentages of labeled source data corresponding to ∼20-700 characteristic events on different sizes of time windows ranging from 2-seconds to 1-minute demonstrates that the developed method can significantly improve automated event detection based on PMU measurements when extensive labeling is costly or impossible to obtain. When compared to the state-of-the-art machine learning algorithms (unsupervised, semi-supervised, and supervised), the results show that the transfer learning method has significantly better performances when detecting events by learning from as low as 20 representative labeled data instances.

**INDEX TERMS** Big data applications, event detection, machine learning, phasor measurement units, power system faults, signal sampling, smart grids, time series analysis.

## I. INTRODUCTION

### A. PROBLEM DEFINITION

The stored data collected by the Phasor Measurement Units (PMUs) at the electric utilities in the USA has increased to hundreds of terabytes in the last few years [1]. In the past decade, PMU data have been used extensively for post-mortem analysis in case of system-wide disturbances. In recent years, utilities have been interested in investigating ways to increase the value of the stored PMU data through novel applications of the machine learning models for improved situational awareness and predictive decision-making capabilities [2].

Event detection is an essential task that involves detecting instances in a dataset that significantly deviate from the norm [3]. The increase in the volume of PMU data is making it more challenging to quickly analyze a large number of historical recordings.

Event detection can be deemed as an unsupervised learning task [4]. Usually, *unsupervised* approaches utilize the underlying assumption that events occur infrequently, meaning they fall in low-density regions of the instance space, or they are distant from normal events to identify them. However, PMU data regularly violate this assumption, affecting the performance of unsupervised approaches (e.g., maintenance events can occur infrequently and irregularly, but are considered normal). Labeled data allow detectors to correct the errors made by unsupervised approaches. Unfortunately, a fully *supervised* learning approach to event detection relies

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

heavily on labeled data, which when done manually may be labor-intensive and hence prohibitively expensive.

### B. CONTRIBUTION

To avoid the expense of extensive manual labeling, semi-supervised approaches to event detection are often used in conjunction with active learning to efficiently collect labels [5]. However, when dealing with a large amount of PMU data, utilizing active learning to assign labels for each individual instance might be infeasible. To reduce the required labeling effort, we employ *transfer learning* to leverage a small number of well-labeled instances from one task to another without additional labeling effort. We demonstrate that a *transfer learning* method is applicable for PMU data and can detect events without having to rely on an extensive number of labels or event logs of PMU data. Our approach outperforms state-of-the-art machine learning algorithms from varying learning types (unsupervised, semi-supervised, and supervised) on a large benchmark when developing the model from a large dataset that requires intensive event labeling effort. Experiments conducted show that the employed transfer learning method is capable of detecting events with as low as ∼20 representative labeled data instances.

### C. PAPER ORGANIZATION

The remainder of this paper is organized as follows. Section II provides and describes the related work. Section III describes and provides preliminaries on transfer learning for event detection. Section IV describes the methodology used to conduct experiments, evaluate and elucidate the proposed transfer learning method, and provides insights into the comparison with a variety of learning types of algorithms used in the literature. Section V describes the data preprocessing techniques used in the experiments. The experimental setup is outlined in Section VI. Section VII presents the experimental results and discussion. Finally, Section VIII concludes the paper. Section IX discusses future work. References are provided at the end.

## II. LITERATURE REVIEW

A variety of studies have investigated ways to reduce the size of the PMU dataset by different means of dimensionality reduction and feature engineering to address the increase in the volume of PMU data. The dimensionality reduction method based on Principal Component Analysis was used in [6] for early online event detection, and in [7] to detect and analyze complex cascading events. The feature engineering method based on the Minimum Volume Enclosing Ellipsoid was reported in [8]. Several studies have used signal transform methods, such as the fast variant of Discrete S-Transform [9]–[11], or wavelet analysis [10]–[13]. Fast event detection based on Detrended Fluctuation Analysis on Big PMU Data was developed in [14]. Domain-specific shapelets were investigated for event detection and classification in [10], [11]. In [15] the Dynamic Programming based Swinging Door Trending was used. Signal Energy Transform was used to detect and classify faults in [16]. Several machine learning models were tested in these studies: Agglomerative Hierarchical Clustering [8], Extreme Learning Machine classifier [9], K-Nearest Neighbor [10], [11], [17], Support Vector Machine (SVM) [10], [11], [17], Decision Tree [17], Convolutional Neural Network [13]. Transfer learning has been applied to several power systems applications in recent years, such as transient stability prediction in [18], detection of oscillation events in [19], and detection of high-impedance faults in distribution systems in [20]. Studies [18]–[20] demonstrate the applicability of transfer learning to a variety of power system problems. Our study extends the benefits of using transfer learning to solve the problem of transmission system event detection from an exceedingly small number of labeled events based on PMU data.

## III. TRANSFER LEARNING FOR EVENT DETECTION FROM A SMALL NUMBER OF LABELS

While event detection tasks would benefit from labeled data, it is often done using an unsupervised approach since assigning labels across all the events manually can be time-consuming and hence costly. The downside is that the unsupervised detectors do not benefit from labeled data that provide the possibility of correcting errors made by the unsupervised detectors. On the other hand, supervised learning algorithms rely on a sufficient number of labeled data. Thus, supervised, and unsupervised learning algorithms are infeasible for event detection tasks when labels are scarce and temporally imprecise. Transfer learning can be utilized to leverage a small number of related labeled data instances from a related task to the target task. Related instances can aid semi-supervised learning algorithms to detect events based on minimal labeled data, since it only selects and transfers tasks that are similar to instances in the target set.

Often, transfer learning is used in conjunction with semi-supervised learning algorithms, since semi-supervised algorithms assume only a limited amount of labeled data instances for training are available. Hence, semi-supervised learning algorithms are employed when labeled data instances are scarce and difficult to obtain. Semi-supervised learning algorithms aim to train a classifier from both the labeled and unlabeled data samples in order to achieve better performance than supervised learning algorithms trained on labeled data only.

The aim of transfer learning is to learn a model for the unlabeled dataset of the target domain given labeled data from a related dataset of the source domain [21]. Since this study concerns event detection, the task is to compute and assign an anomaly score to each time window (data instance) in the target dataset that quantifies how anomalous the time window is based on similarity measures; assigning an anomaly score to a time window can be compared with a predefined threshold to classify whether an anomalous event exists within a given time window [21]. We use $D_s$ to denote the source dataset, which contains labeled time windows, and $D_t$ to denote the target dataset, which contains unlabeled time windows of events to be classified as either normal or anomalous events. We use $x_s$ to refer to a time window from the source dataset, and $x_t$ to refer to a time window from the target dataset.

There are three important assumptions for transfer learning techniques to be considered when applied to event detection tasks [22]. First, the source and target datasets were obtained from the same $m$-dimensional feature space. Second, the marginal distributions of the source and target datasets differ (covariate shift assumption). A covariate shift assumption occurs when dissimilar behaviors are observed in either domain. Third, the conditional distributions can differ due to changes in context, meaning the same behavior might have a different meaning in the two domains (concept shift assumption). Assumptions two and three complicate the transfer task.

## IV. METHODOLOGY

To demonstrate the performance of the utilized transfer learning method, a comparative analysis with a multitude of event detection algorithms with varying learning types used as a baseline was performed. Additionally, different datasets were used for experiments with varying splits of the data and window dimensions.

### A. UNSUPERVISED LEARNING

Unsupervised learning algorithms aim to identify hidden patterns without using any labeled data samples. Thus, unsupervised learning algorithms are capable of learning without an error signal to assess and evaluate the performance of the model. Since unsupervised learning algorithms do not require any labels during learning and identifying hidden patterns, event detection tasks using this method can be beneficial when labels are not available [23]. However, unsupervised learning algorithms utilize the fundamental assumption that events occur infrequently, and PMU data often violate this assumption [5]. The lack of labeled data instances that provide the option to correct the errors made by unsupervised detectors degrades the performance of the algorithms.

As a part of the comparison study, an event detection experiment was performed using two unsupervised learning algorithms, namely: 1) the k-nearest neighbor outlier (kNNO) detection algorithm that computes for each data point the anomaly score as the distance to its k-nearest neighbors in the dataset [4], and 2) the isolation nearest neighbor ensembles (iNNE) algorithm that computes for each data point the anomaly score roughly based on how isolated the point is from the rest of the data [24]. They learn a structure on the training data without incorporating any labels into the models. Event detection is performed on the test dataset to classify data samples as anomalous or normal events.

In order to assess the performance of the algorithms, the predicted labels were compared to the ground truth (actual) labels obtained by visual inspection by a domain expert.

### B. SUPERVISED LEARNING

Supervised learning is based on training a model using previously observed labeled data samples and assuming that the marginal distribution of the source training data and the target test data are identical (no covariate shift assumption). Supervised learning algorithms tend to rely heavily on learning data samples and require a sufficient amount of training data before performing classification, which can be infeasible in event detection tasks [23]. The more complex the problem and the models are the more training data is required.

We employed state-of-the-art and most common conventional supervised learning algorithms to compare with other learning types. We used scikit-learn library for Machine Learning in Python [25]. A variety of classification algorithms from this library were utilized, including Multilayer Perceptron (MLP), Logistic Regression (LR), K-Nearest Neighbor (KNN), Support Vector Machine (SVM).

### C. SEMI-SUPERVISED LEARNING

The semi-supervised learning concept is in between unsupervised and supervised learning. Semi-supervised classification algorithms aim to train a classifier from both the labeled and unlabeled data samples, such that they achieve better performance than the supervised or unsupervised learning algorithms. There are many practical benefits in using semi-supervised learning, especially, when labeled data instances are scarce and difficult to obtain, since such algorithms assume only a limited amount of labeled data instances for training are available. Semi-supervised learning algorithms might perform as well as supervised learning algorithms, but with much fewer time-labeled data instances, which is beneficial in event detection tasks to reduce annotation effort resulting in reduced implementation costs [26].

Two semi-supervised learning algorithms that do not rely entirely on labels obtained from event logs or by visual inspection to classify data samples as normal or anomalous events were utilized: 1) the semi-supervised k-nearest neighbor anomaly (SSKNNO) detection algorithm, which is a combination of the well-known kNN (i.e., unsupervised learning) classifier and the kNNO (k-nearest neighbor outlier detection) (i.e., supervised learning) method [5]. Since SSKNNO is a distance-based method that relies on Euclidean similarity measure, the number of labeled instances does not affect the learning process. Having as minimum as one labeled data instance from each pattern of signals or type of event should be sufficient for the algorithm to detect events. The algorithm uses an unsupervised setting when a similar labeled data instance is not available in the training data. 2) the semi-supervised detection of outliers (SSDO) algorithm, which computes an unsupervised prior anomaly score, and then, corrects this score with the known label information. It is based on constrained k-means clustering [27]. These algorithms take a partially labeled dataset that consists of three labels: unknown (0), event (1), normal (−1), and assigns a binary label (−1, 1) to each unknown instance in the dataset. The performance of the algorithm was assessed by comparing the predicted labels to the ground truth labels. Small proportions of labeled data samples combined with unlabeled data samples were used during training.

### D. TRANSFER LEARNING + SEMI-SUPERVISED LEARNING

We formulate the event detection task using transfer learning technique as:

**Input:** $D_s$ and $D_t$ from the same feature space. Where $D_s$ denotes a source dataset containing labeled time windows and $D_t$ denotes a target dataset containing unlabeled time windows $D_{t,}$;

**Do:** Compute an anomaly score for every time window in $D_t$ based on $D_t$ and a subset of the related time windows in $D_s$;

**Output:** $y$ labels (predictions) indicating whether a time window in $D_t$ contains normal or anomalous behavior.

A two-step transfer learning approach used in our study is based on a recently introduced LocIT algorithm [5], that was not yet applied on PMU data. First, the algorithm takes as an input a labeled source dataset $D_s$. Then, it selects a subset from the labeled source time windows to transfer to the unlabeled target dataset $D_t$. If the local data distribution of a certain time window is similar in both the source and target datasets, the algorithm transfers the time window from the source to the target domain. LocIT utilizes unsupervised learning techniques since labels for time windows in the target dataset are not available and the labeled time windows in the source dataset should not influence the transfer decision. Second, the algorithm computes an anomaly score using a semi-supervised learning algorithm based upon nearest-neighbor techniques that consider both the related time windows that were selected and transferred from the source $D_s$ and the unlabeled target time windows [5].

LocIT selects and transfers similar time windows from $D_s$ to $D_{trans}$, where $D_{trans}$ is a subset that contains the selected labeled time windows for transfer [5]. Let $D^* = D_t \cup D_{trans}$, where $D^*$ is a dataset containing the transferred time windows combined with the target unlabeled time windows. $D^*$ is a partially labeled dataset, where time windows from $D_{trans}$ are labeled as an event (1) or normal (-1), and $D_t$ time windows are labeled as unknown (0). Then, a semi-supervised SSKNNO algorithm takes $D^*$ as input and classifies each unknown time window as an anomalous event or normal, indicating whether a given event occurred in a given time window or healthy signal respectively. This process is further illustrated in the flowchart in Fig. 1.

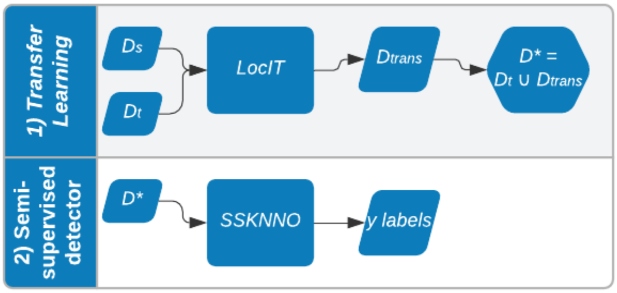

**FIGURE 1.** Flowchart that illustrates the two-step process of event detection using transfer learning + semi-supervised detector.

*Local Structure of Time Windows:* LocIT algorithm defines the localized source distribution for a given source time window $x_s$ using the subset $N_\psi\ (x_s, D_s)$ of the nearest neighbor $\psi$ of $x_s$ in $D_s$; and defines the localized target distribution based upon the subset $N_\psi\ (x_s, D_t)$ of $x_s$'s $\psi$ nearest neighbor in $D_t$. Where $\psi$ controls the strictness of the transfer. The higher the value of $\psi$ is (i.e., 1.0), the stricter the transfer is. If $\psi$ is 0, the algorithm ignores the differences of local distribution and considers the complete global structure of $D_s$ and $D_t$ to determine the transfer.

The algorithm transfers a time window from the source subset to the target subset if the distributions of both subsets are sufficiently identical. The similarity measure (i.e., *location distance*) used to compare the first and second order statistics of $N_\psi\ (x_s, D_s)$ and $N_\psi\ (x_s, D_t)$ is defined as:

$$d_1\,(N_1, N_2) = \left\| \frac{1}{k}\left(\sum_{x_i \in N_1} x_i - \sum_{x_j \in N_2} x_j\right)\right\|_2. \qquad (1)$$

The *location distance* used in equation (1) is the l2-norm of the difference of the arithmetic mean (i.e., centroids) between two neighborhood subsets $N_1$ and $N_2$. Large values of $d_1$ reduce the chance of meaningful transfer.

The distance between the covariance matrices of two neighborhood subsets (i.e., *correlation distance*) is defined as:

$$d_2\,(N_1, N_2) = \frac{\|C_{N_1} - C_{N_2}\|_F}{\|C_{N_1}\|_F} \qquad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm and $C$ is the covariance matrix. The Frobenius norm was considered since $N_1$ and $N_2$ are matrices. Large values of $d_2$ indicate that the localized distributions of the source and target subsets are different, which decreases the chance of a meaningful transfer.

*Learning the Transfer Function:* In order to transfer a time window from the source $D_s$ to target subset $D_t$, the transfer function decides whether to transfer the time window based upon combining the values of $d_1$ and $d_2$. LocIT utilizes an SVM classifier that learns on the target distribution using the target data only to serve as the transfer function. SVM predicts whether a time window in the source instance fits in the target domain by leveraging the smoothness assumption, having the meaning that neighboring target time windows have similar localized distributions while the farthest time windows have dissimilar localized distributions. Hence, the negative training instances are generated by computing for every time window in the target subset a feature vector consisting of the distances between the neighborhood subsets of $x_t$ and its farthest neighbor. The one positive training instance is generated for each instance $x_t$ by finding its nearest neighbor in the target subset and computing $d_1$ and $d_2$ on the target subset. Finally, once the SVM classifier is trained on the target subset using both the negative and positive training instances, each instance from the source subset can be predicted to check whether it belongs to the target. If it belongs, the algorithm transfers it and adds it to $D_{trans}$.

## V. DATA PROCESSING

### A. PMU DATA

The PMU dataset used for testing was provided in the Apache Parquet database. The original dataset contains measurements from 38 PMUs from the Western Interconnection of the U.S.A. captured over a period of two years (2016-2017). The dataset was anonymized by the provider. Geographical locations of the PMUs and the network topology information are not made available. The data are collected with two frame reporting rates per second (fps), 30 fps, and 60 fps, and contain measurements from PMUs located at several voltage levels in the transmission network. This dataset corresponds to a variety of event types, including line fault transformer outages, and frequency events. Some data quality issues, such as missing data, data duplicates, and outliers were observed but did not have a significant impact on our method.

### B. EVENT LOG

The event log received from the data provider contains manually created labels with only an approximation of the event start time with a precision of 1 minute. We referred to these labels as temporally imprecise. The time labels were not created based on the PMU time reference; thus, some events were mislabeled and did not occur at the location of the PMUs used in this study. Using such limited labels makes it challenging to temporally extract more precise PMU labels from the event log.

To extract temporally more precise labels, we considered a set of labels created based upon visual inspection of the PMU-recorded signals by a domain expert on our team. The domain expert on our team relabeled the data to ensure that the labels were accurate and precise, since the initial labels (event log) received were inaccurate. The different sets of labels (1-minute, 30-seconds, 10-seconds, 5-seconds, 2-seconds) for event and normal operations identified through this study are presented in Table 1.

### C. FEATURE EXTRACTION

We defined the *Rectangle Area* (*RA*) features extracted per PMU for each time window. No data cleansing was performed on the PMU dataset from the chosen 38 PMUs prior to the feature extraction. The *RA* feature, created using the frequency and positive-sequence voltage magnitude measurements, is defined as:

$$RA_{PMU,TW} = (f_{max} - f_{min}) * (V_{max} - V_{min}) \tag{3}$$

where $f_{\max}$ and $f_{\min}$ are the maximum and minimum frequency values, and $V_{\max}$ and $V_{\min}$ are the maximum and minimum positive sequence voltage magnitude recorded by the selected *PMU* device, inside the selected time window *TW*.

After feature extraction, only minor cleansing of outliers was performed by removing *RA* values that were too large to be possible. Only 11 *RA* values were discarded. They were replaced with zeros. The impact of missing data is negligible. If at least two data points were present inside a time window, the RA was calculated. For example, in the case of the 1-minute window on a 30 fps PMU, we only need 2 out of 1800 ($30fps * 60sec$) points to be able to calculate the *RA*. In case there is only one data point within a time window, *RA* is set to zero. Data duplicates do not have any impact on this method since the minimum and maximum values of voltage and frequency are not affected by the duplicates.

The RA feature is sufficient to capture whether an event has occurred within a time window. The RA feature is limited to detecting events and is not suitable for classifying event types. The RA feature was used since it yielded the best performances among multiple data processing techniques that were tested. Furthermore, aggregated RA features allow the utilization of simple and efficient similarity measures to compute distances between time windows to find the nearest and farthest neighbors.

Data processed based on the rectangle area were standardized using *StandardScaler,* which subtracts the mean, and then scales each feature to unit variance.

**TABLE 1.** Number of labels per category and window selection method.

| Event Log | # Event Labels | # Normal Labels | Event Start Time | Event End Time |
|---|---|---|---|---|
| **1-min Labels** | 1033 | 923 | $ST_{\mathrm{VI}} - 5$ sec | $ST_{\mathrm{VI}} + 55$ sec |
| **30-sec Labels** | 1038 | 1846 | $ST_{\mathrm{VI}} - 2$ sec | $ST_{\mathrm{VI}} + 28$ sec |
| **10-sec Labels** | 1038 | 1846 | $ST_{\mathrm{VI}} - 1$ sec | $ST_{\mathrm{VI}} + 9$ sec |
| **5-sec Labels** | 1038 | 1846 | $ST_{\mathrm{VI}}$ | $ST_{\mathrm{VI}} + 5$ sec |
| **2-sec Labels** | 1038 | 1846 | $ST_{\mathrm{VI}}$ | $ST_{\mathrm{VI}} + 2$ sec |

$ST_{\mathrm{VI}}$ - *start time of the event based on visual inspection.*

### D. TEMPORAL SPLIT

A set of 38 PMUs that contain time windows collected over a span of two years, 2016 and 2017 was split into two subsets, where the first subset was used as a source dataset for transfer learning, $D_s$, and the second subset is the target for transfer learning, $D_t$. The split between two the subsets was based on the temporal split between the years 2016 and 2017. Knowledge was leveraged and transferred from the year of 2016, $D_s$, to the target subset $D_t$, which contains time windows collected from the year of 2017. $D_t$ is a fixed dataset that contains all windows from 2017 in all the experiments conducted. Proportions of labeled time windows were randomly selected from $D_s$, and combined with target time windows, $D_t$, in a dataset $D^*$, which is a partially labeled dataset that contains the transferred related labeled windows from $D_s$ and windows to be classified as anomalous or normal event, $D_t$.

**TABLE 2.** Split into two subsets of PMUs for transfer learning based on calculated *rectangle area* during events.

| | Event 1 | Event 2 | Event 3 | Event 4 | Comment |
|---|---|---|---|---|---|
| **Top 1** | **$RA_{PMU1}$=56** | $RA_{PMU5}$=32 | **$RA_{PMU2}$=48** | $RA_{PMU4}$=17 | Only the top 3 PMUs with largest RA for an event are considered as candidates for the Source Subset |
| **Top 2** | $RA_{PMU3}$=54 | **$RA_{PMU2}$=31** | $RA_{PMU7}$=32 | $RA_{PMU6}$=16 | |
| **Top 3** | $RA_{PMU7}$=44 | $RA_{PMU4}$=28 | $RA_{PMU3}$=27 | **$RA_{PMU1}$=12** | |
| **Top 4** | $RA_{PMU2}$=42 | $RA_{PMU1}$=27 | $RA_{PMU1}$=24 | $RA_{PMU7}$=8 | The rest of the PMUs with lower RA are not considered as candidates for the Source Subset |
| … | … | … | … | … | |
| **Final split with minimum elements in *PMU Source Subset*** | | | | | |
| *PMU Source Subset, $D_s$* = {PMU1, PMU2} Each event has at least one of these two PMUs in the Top 3 based on the *RA* | | | | | |
| *PMU Target Subset, $D_t$* = {PMU3, PMU4, PMU5, PMU6, PMU7} | | | | | |

While proportions of labeled time windows were randomly selected, it was ensured that the selected time windows result in a balanced subset containing both anomalous and normal events.

For the unsupervised, semi-supervised, and supervised classifiers, the set of 38 PMUs was also split temporally, hence, training data containing data time windows from the year of 2016, and test data containing data time windows from the year of 2017. Since these classifiers do not transfer related time windows, classifiers were trained on entire time windows from 2016, and tested/classified time windows from the future, hence, time windows collected from the entire 2017.

Features for a certain time window were combined into a feature vector that contains 38 RA features, one feature for each observed PMU. Labels $y$ are created for each time window as ('1' – in case of an event reported, '$-1$' – in case of a normal operation) for transfer learning and semi-supervised learning classifiers. Whereas, unsupervised and supervised learning classifiers, windows are labeled as ('1' – in case of an event reported, '0' – in case of a normal operation). When performance measures were applied to assess the performance of the transfer learning and semi-supervised classifiers, predicted labels '$-1$' were transformed to '0' to match with ground truth labels.

### *E. PMUs SPLIT*

Similarly, a set of 38 PMUs was split into two subsets, the source subsets, $D_s$, and the target subset, $D_t$. The split between two subsets was made using *RA* feature based on the following procedure. First, a set of 35 events was selected randomly. For each of the 35 events, the *RA* feature was extracted on each PMU. For each of the 35 events, top 3 PMUs with the greatest *RA* were selected. Different subsets of PMUs were iterated until the smallest subset was found that had at least one of top three PMUs in each of the 35 events. This resulted in 12 chosen PMUs that combined have a representative in the top three *RA* in all 35 events. The procedure is outlined in Table 2 using a simplified example with 7 PMUs and 4 events.

Additional 7 PMUs were selected randomly from the remaining set of PMUs, totaling the final 19 PMUs in the *PMU Source Subset*. The remaining 19 PMUs were placed in the *PMU* target subset, $D_t$. A proportion of labeled time windows from $D_s$ were randomly selected; selected time windows from $D_s$ were leveraged and knowledge was transferred to $D_t$. Then, related time windows selected for transfer from $D_s$ were combined with $D_t$ in a dataset $D^*$.

Similarly, for the unsupervised, semi-supervised, and supervised classifiers, $D_s$ was used as the training subset and $D_t$ was used as the test subset. Since the aforementioned classifiers do not transfer related time windows to the target domain, all windows from the set of PMUs in $D_s$ were used for the prediction task.

Features for a certain time window are combined into a feature vector that contains 19 RA features. The process of creating labels $y$ is identical to the process of the *Temporal Split* experiment.

## VI. EXPERIMENTAL SETUP

Extensive experiments conducted in our study are described in this section. Using limited proportions of labeled data incorporated into the models we assessed and compared the capabilities of our method to alternative models (unsupervised, semi-supervised, and supervised) to detect events based on a limited proportion of labels, or without any labels used. Experiments conducted included 2%, 5%, 10%, 25%, 40%, 55%, and 70% of available labeled data, corresponding to 20, 51, 103, 259, 415, 570, and 726 available labeled data instances respectively. Available data instances were randomly selected from the source dataset $D_s$, whereas target dataset $D_t$ was fixed among all experiments. This does not apply to unsupervised learning algorithms since they do not incorporate any labels during learning. The performance of the classifiers was evaluated using the area under the receiver operating characteristic (AUROC) since this metric is the standard in event detection tasks [28]. Other relevant performance measures including Precision, Recall, F-1 score, and Matthews Correlation Coefficient (MCC) (also known as phi coefficient) were also reported. The formal definitions of these metrics are very common and can be easily found [29, 30, 31].

The different uses of leveraging knowledge from source to target domain are illustrated in Section V-D and Section V-E. A variety of experiments were conducted to address the following comparative questions:

- How do window sizes (time intervals) over which features were computed affect the performance of the algorithms? Different window sizes varying from 2 seconds to 1 minute were experimented to determine the best choice for the event detection task.
- How does the percentage of labeled time windows in the source data affect the performance of the models? Varying percentages of labeled source data ranging from 2% to 70% were experimented to analyze the performance of the models and analyze what percentage of labeled source data is sufficient for the models to detect events.

### A. HYPERPARAMETER TUNING

Hyperparameter tuning using cross-validation is infeasible since labels of time windows in the target domain are not available, and the distributions of the source data $D_s$ and target data $D_t$ are dissimilar [31]. Instead, the baseline and recommended hyperparameters in comparative studies were used. LocIT has three significant hyperparameters that need to be set. We used a transfer threshold $\psi$ of 0.7, which indicates how closely related time windows to be transferred are and scaling that determines whether to scale the source and target domain before transfer using StandardScaler. In the final classifier, SSKNNO, the three significant hyperparameters were set as contamination of 0.34, k of 1, and strict supervision. The contamination is the threshold of anomaly score, k is the number of nearest neighbors, and supervision indicates whether to use all time windows in the set of nearest neighbors (loose) or use only windows that count the window among their neighbors. Hyperparameters that were set for all classifiers are listed in Table 3, categorized by a learning type.

**TABLE 3. Selected hyperparameters for the binary classifiers categorized by learning type.**

| | |
|---|---|
| Unsupervised | **kNNO** (weighted=True, k=10, contamination=0.34) |
| | **iNNE** (n_members=1000, sample_size=16, contamination=0.25) |
| Supervised | **MLPClassifier** (alpha=0.3) |
| | **LogisticRegression** (C=0.9) |
| | **KNeighborsClassifier** (n_neighbors=10, weights='distance') |
| | **NuSVC** (kernel='rbf') |
| Semi-supervised | **SSkNNO** (metric='euclidean', k=1, supervision='strict', weighted=True, contamination=0.34) |
| | **SSDO** (metric='euclidean', k=10, contamination=0.39, alpha=0.2) |
| Transfer Learning | **LocIT** (transfer_threshold=0.7, scaling='none', metric='euclidean') |

## VII. EXPERIMENTAL RESULTS AND DISCUSSION

### A. DISTRIBUTIONAL DIFFERENCE BETWEEN SOURCE AND TARGET DATASETS

To demonstrate the applicability of utilizing transfer learning techniques on PMU measurements data for event detection, the three assumptions explained in Section III had to be validated. Transfer learning is typically applied on datasets where traditional machine learning modeling assumptions are violated since the marginal distributions of the source and target subsets are dissimilar (covariate shift assumption), or the conditional distributions are different owing changes in context, in which the meaning of the same behavior might be different in both the source and target domains (concept shift assumption). Therefore, in the initial experiment of our study Kolmogorov-Smirnov (KS) test for comparing the similarity between two continuous distribution functions $G$ and $F$, was used to check whether the source and target distributions are identical by comparing the underlying distributions $F(x)$ and $G(x)$ of two independent samples [32], where $x$ denotes to the RA features for a certain PMU. The null hypothesis was $F = G$, indicating that the distributions of the source and target are identical.

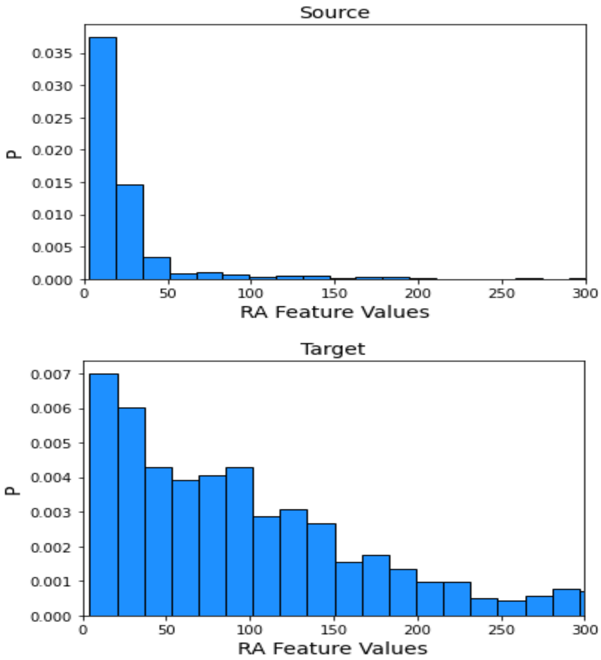

**FIGURE 2. An example to illustrate the distributional difference between the source (2016) and target (2017). X-axis is the time window RA values for a certain PMU, and y-axis, *P* denotes to the probability that a certain RA feature will belong to a certain pin. Both top and bottom figures show the distribution of the same PMU over two years, where source contain RA features collected over 2016 and target contain RA features collected over 2017.**

We applied the KS test metric on the source and target subsets, where the source is a 1-dimensional array containing the RA features collected over the year 2016 for a single PMU, and the target was a 1-dimensional array containing the RA features collected over 2017 for the same PMU. This process

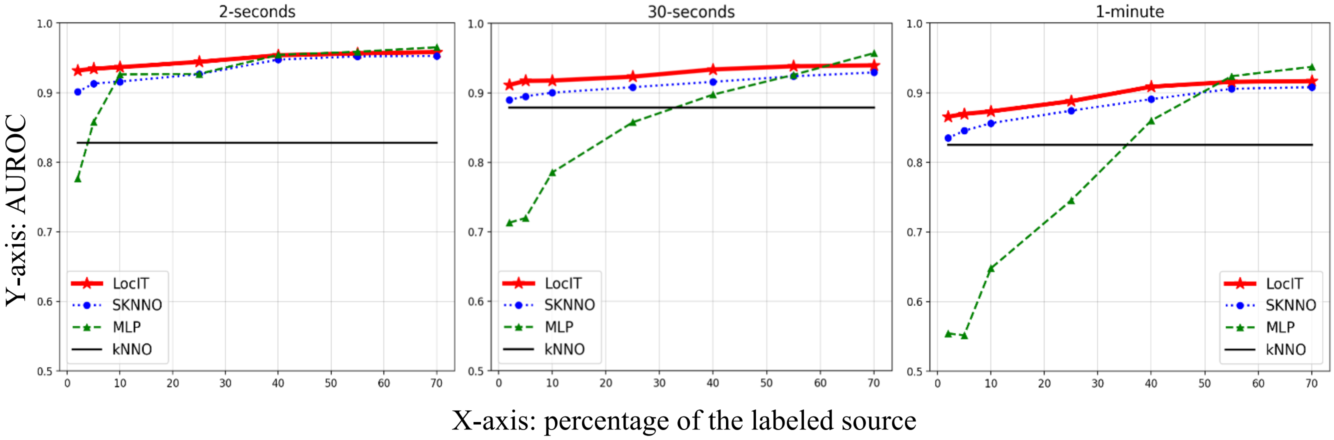

**FIGURE 3.** **Comparing performances of the proposed transfer learning algorithm LocIT based on varying percentages of labeled source data on different window dimensions to three alternative learning types SKNNO, MLP, and kNNO, based on AUROC metric. Performances improve with more labeled data added to the source. Results are consistent and show that LocIT, always performs better with limited labels used. This experiment was conducted based on a temporal split.**

was repeated for each PMU. Furthermore, we confirmed the results using the Empirical Distribution Function (EDF) by modeling and sampling the cumulative probabilities for a data sample that does not fit standard probability distribution.

We obtained the p-values from the KS test metric for all PMUs. The maximum p-value was $3.9e^{-15}$, hence, due to a very small p-value (i.e., $< 0.01$) we can safely reject the null hypothesis, indicating the source and target distributions are different. Fig. 2 shows an example for one PMU to illustrate the distributional difference between the source and target subsets. The top and bottom figures show the distribution of the same PMU over two consecutive years.

Many reasons could lead to a distributional difference of PMU data collected from the future. That might occur since power systems experience randomness of occurrence of events depending on the circumstances, including but not limited to, weather, equipment failures, wear and tear, and the fact that operating conditions differ every year. Thus, this experiment suggests that transfer learning could be more applicable than supervised learning alternatives that assume the same distribution.

### B. THE EFFECT OF VARYING PERCENTAGES OF LABELED DATA

In order to study the effect of the amount of labeled source data on the performance of our model versus other models, a variety of percentages of labeled time windows were analyzed. Often, it is non-trivial to acquire labeled data or event logs for event detection tasks since label extraction can be expensive and sometimes impossible to obtain. Thus, this experiment is relevant and provides insights on what percentage of labeled data is sufficient for the models to detect anomalous events. We randomly sampled 2%, 5%, 10%, 25%, 40%, 55%, and 70%, corresponding to 20, 51, 103, 259, 415, 570, and 726 of labeled source data $D_s$ and only considered these labeled time windows when performing a transfer. Experiments were repeated five times and the results were averaged. This experiment was conducted based on *Temporal Split* of the data described in Section V-D. The best performing methods from three alternative learning types (i.e., unsupervised, semi-supervised, fully supervised) were chosen and compared to the transfer learning method.

Fig. 3 shows that the AUROC improves with more labeled data added to the source subset on three datasets with different window dimensions, listed in Table 1. The utilized transfer learning method LocIT, outperforms unsupervised kNNO, semi-supervised SSKNNO, fully supervised MLP, learning algorithms with limited or no labels used in the source data. With only 2% of labeled source data used, corresponding to ∼20 characteristic events the transfer learning algorithm performed generally well, while the fully supervised algorithm performed poorly. The gap widened between supervised learning and transfer learning algorithms as the labeled source data decreased. However, with $>$60% of labels, supervised learning outperformed transfer learning with a slight increase in AUROC. There were considerable discrepancies between supervised and transfer learning algorithms with $<$10% of labels used in all experiments conducted. The unsupervised kNNOs curves are straight lines because it is trained without using any labels and it only considers the target data. This was included to visualize and compare with other algorithms. The unsupervised algorithm was trained without any labels, outperformed supervised learning algorithm with $<$5% of labels in 2-seconds time windows. With 1-minute and 30-seconds time windows, unsupervised outperformed supervised learning with approximately $<$30% of labels used. Unsupervised learning performed poorly compared to transfer learning and semi-supervised algorithms with a small percentage of labeled data used, since labeled data can assist with correcting the errors made by unsupervised detectors. The semi-supervised algorithm's performance was adjacent to transfer learning's performance with >10% of labeled data. Transfer learning's performance was greater than semi-supervised learning with <10% of labeled data since only related time windows were used to guide with the event detection task. On average transfer learning yields an average increase in AUROC of approximately 13% compared to supervised learning, and 5% compared to unsupervised learning. This provides an evidence that the proposed transfer learning approach can help with PMU event detection tasks when labels are not available or are expensive to obtain. Additionally, this shows that supervised learning algorithms rely heavily on labels and are infeasible for detecting events with limited labeled data.

**TABLE 4. Performance of various models trained using only 20 labeled events based on *temporal and pmus split*.**

| Experiment | Model | AUROC | Precision | Recall | F1-score | MCC |
|---|---|---|---|---|---|---|
| 2-sec. Temporal Split | LocIT | **0.93** | 0.93 | 0.93 | 0.93 | 0.86 |
| | SSKNNO | 0.90 | 0.92 | 0.92 | 0.92 | 0.83 |
| | SSDO | 0.82 | 0.84 | 0.84 | 0.84 | 0.67 |
| | MLP | 0.77 | 0.89 | 0.77 | 0.79 | 0.64 |
| | LR | 0.68 | 0.86 | 0.68 | 0.69 | 0.52 |
| | KNN | 0.67 | 0.85 | 0.67 | 0.68 | 0.50 |
| | SVM | 0.66 | 0.85 | 0.66 | 0.66 | 0.48 |
| | kNNO | 0.82 | 0.84 | 0.84 | 0.84 | 0.67 |
| | iNNE | 0.81 | 0.86 | 0.85 | 0.84 | 0.69 |
| 2 sec. PMUs Split | LocIT | **0.94** | 0.95 | 0.95 | 0.95 | 0.89 |
| | SSKNNO | 0.90 | 0.93 | 0.92 | 0.92 | 0.84 |
| | SSDO | 0.80 | 0.85 | 0.84 | 0.84 | 0.68 |
| | MLP | 0.74 | 0.84 | 0.81 | 0.79 | 0.59 |
| | LR | 0.71 | 0.87 | 0.71 | 0.72 | 0.55 |
| | KNN | 0.76 | 0.85 | 0.82 | 0.81 | 0.61 |
| | SVM | 0.72 | 0.87 | 0.72 | 0.74 | 0.57 |
| | kNNO | 0.76 | 0.83 | 0.82 | 0.80 | 0.62 |
| | iNNE | 0.86 | 0.90 | 0.89 | 0.89 | 0.84 |

Transfer Learning: LocIT; Semi-supervised: SKNNO, SSDO; Supervised: MLP, LR, KNN, SVM; Unsupervised: kNNO, iNNE

## C. TRANSFER LEARNING vs. BASELINE ANOMALY DETECTORS

Table 4 compares the proposed transfer learning algorithm, LocIT to baseline algorithms with different learning types based upon the best performing length of time windows (i.e., 2-seconds) with only 2% of labeled data used to challenge the event detection task based on *Temporal Split* and *PMUs Split* of the data. In both experiments, transfer learning outperformed unsupervised, semi-supervised, and fully supervised algorithms. Results were consistent among all experiments conducted on all datasets. With sufficient amounts of labeled data available, supervised algorithms perform well. However, when limited or no labels are available, unsupervised algorithms, semi-supervised, and transfer learning with semi-supervised, outperform supervised learning algorithms. Additionally, Table 4 shows consistency of results where LocIT outperforms other models with limited labeled data and shows significant improvement over supervised algorithms (MLP, LR, KNN, SVM). Experiments conducted provide evidence that Transfer Learning and Semi-supervised algorithms are more feasible than supervised algorithms for event detection tasks when labels are scarce.

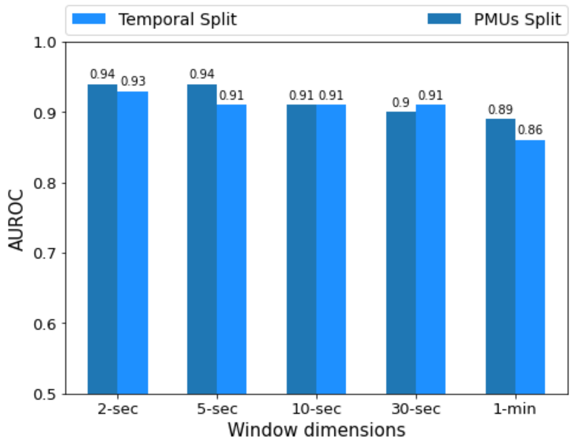

**FIGURE 4. LocIT's performance based on AUROC by varying window sizes based on temporal and PMUs data split.**

## D. THE EFFECT OF VARYING TIME WINDOW SIZES

In order to explore how the window size affects the performance of the models, a variety of window sizes were analyzed, including 2-seconds, 5-seconds, 10-seconds, 30-seconds, and 1-minute window sizes. Fig. 4 shows that the AUROC improves with shorter window sizes among both experiments *Temporal Split* and *PMUs Split*. Applying transfer learning algorithm on 2-seconds window size yields an approximately 7% increase compared with 1-minute windows based on *Temporal Split* of the data, and 5% increase based on *PMUs Split* of the data. The increase in AUROC is due to the nature of the distance-based classifier. Each time window was manually inspected to make sure that the start of the anomalous event fell within of the selected time window. Anomalous events result in fluctuations (i.e., abnormal behavior) in the signal. As such, when detecting events, distance metrics in shorter time windows highlight the deviation from normal operation more than when longer time windows are used, since the anomalous event corresponds to a shorter timeframe, whereas the rest of the signal corresponds to normal operation. Hence, having a longer time window can dilute the event effect in the window. Fluctuations impact the RA feature values owing to the difference between the minimum and maximum values of positive sequence voltage magnitude and frequency. There was no significant increase in AUROC between the 2-second and 5-second windows. There was a significant improvement in AUROC when detecting events based on 2-second windows compared to 1-minute windows. The increase in AUROC observed when comparing 1-minute

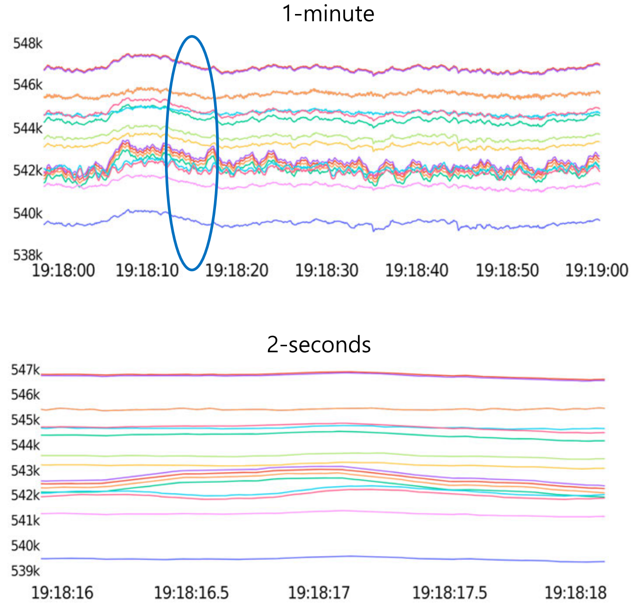

**FIGURE 5.** The top figure shows 1-minute time window of normal operation, and the bottom figure shows 2-seconds time window of normal operation. Each colored line corresponds to a certain PMU. Shorter time windows produce less fluctuations, resulting in better performance in detecting events.

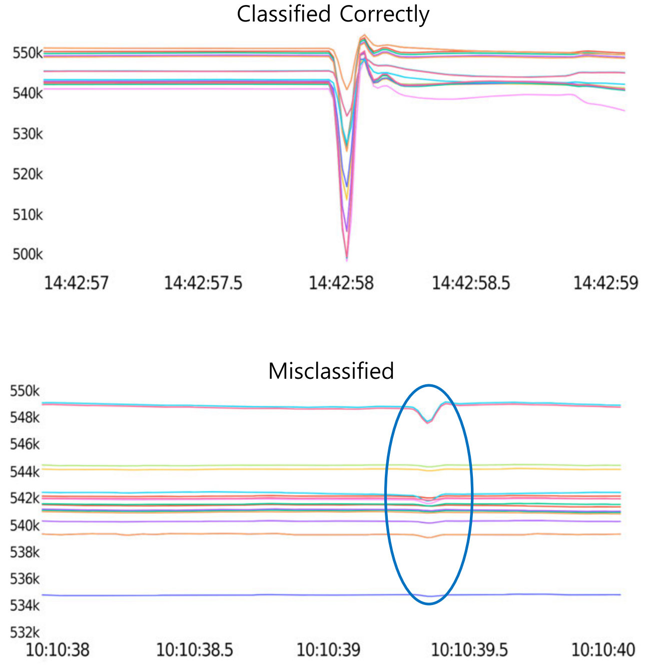

**FIGURE 6.** Both the top and bottom figures show 2-second time windows that contain events. The top figure shows an obvious event that was observed by most PMUs and was classified correctly as anomalous event. The bottom figure shows a very minor dip in voltage that did not affect most of the PMUs; hence, it was not classified correctly. The bottom time window was classified as normal event.

labels to 2-second labels can be explained by the smaller fluctuation of normal operation within a shorter time window. Experiments conducted show that shorter time windows result in a higher AUROC. Thus, the size of the time windows was determined based upon the size that exhibited the best performances formed on the results obtained from the conducted experiments.

Fig. 5 demonstrates the fluctuations caused by longer time windows based upon two normal operation instances captured in a 2-second and 1-minute time windows. 1-minute time window shows more fluctuations occurred during the normal operation, and the 2-second time window showed slight fluctuations occurred during the normal operation. Thus, the use of a shorter time window exhibits less fluctuation resulting in a better performance.

### E. LEVERAGING KNOWLEDGE ON TEMPORAL AND PMU SPLITS

Two experiments were conducted based on different ways of leveraging the data from source to target to test and ensure the model's robustness. Different data splits are introduced in Section V-D and Section V-E. Fig. 4 shows the performance of transfer learning algorithm, LocIT, based on *Temporal Split* and *PMUs Split* of the data. As can be seen, leveraging limited knowledge temporally, from the year of 2016 and transfer to 2017 results in a slight decrease in AUROC, which can be explained by the randomness of occurrence of events depending on the circumstances that might differ from a year to another. Leveraging knowledge from a set of PMUs to another set of PMUs shows an increase in AUROC, but the difference is not significant. Splitting the source and target datasets temporally yields a decrease of approximately 1.5% compared with the PMUs split of the data. Thus, this shows that it is feasible to transfer knowledge from historical data and apply it on time windows from the future, and/or time windows from a specific set of PMUs to another set of PMUs.

### F. MISCLASSIFIED TIME WINDOWS

We examined the time windows that were misclassified by the transfer learning approach to event detection to develop a better understanding of the nature of events that led to errors in detecting events. Both false positives (FPs) and false negatives (FNs) were observed. FPs are events that were misclassified as anomalous operation, but in fact, they were normal operations. FNs are the events that were misclassified as normal operation, but in fact, they were anomalous events. A pattern was observed based upon visually inspecting the misclassified events. These events were local events, meaning that they were not observed by most of the PMUs in the interconnect. Moreover, their impact on local PMUs in terms of prominent changes in voltage or in frequency is weak as compared to major events that might precede them. Recall that the input to the algorithm is a vector of RA features from all the PMUs. The difference between the maximum and minimum voltage and frequency in these instances was not as substantial, resulting in a smaller value of RA. Thus, since most PMUs did not observe these changes, false classification (i.e., errors) occurred. Additionally, since the employed semi-supervised detector is distance-based, the weak changes in voltage or in frequency affected the distance metric due to fluctuations in the time window, hence, the detector misclassified these instances.

Furthermore, upon visual inspection of the misclassified events, we observed unrealistic values of frequency in the order of thousands of Hz or sudden drops to zero in the value of voltage and frequency, since no data cleansing was performed prior to the extraction of the RA features. Hence, this led to false classifications as well.

Fig. 6 provides two 2-second time windows that contain events. The top figure shows an apparent event with a significant drop in voltage and was observed by most PMUs, hence, it was classified correctly as anomalous event. The bottom figure shows a misclassified time window since there was a very minor drop in voltage and did not affect most of the PMUs (i.e., local event).

### G. STATISTICAL SIGNIFICANCE ANALYSIS

Statistical analysis was performed to assess the significance and stability of the proposed method's performance. To address how frequently we can expect the proposed algorithm LocIT to obtain the same performance measured based upon the AUROC metric under different conditions, we randomly selected 19 PMUs for the source data and the remaining 19 PMUs were the target data. For consistency with experiments conducted in this study, we leveraged only 20 labeled time windows from the source data to predict the target domain. We repeated the random selection of PMUs 10 times and obtained results from all algorithms used in this study. Then, we employed a t-test with a significance level of 0.1 to obtain confidence intervals with 90% confidence level for the average AUROC for each algorithm individually. Table 5 summarizes the average AUROCs and their corresponding two-sided confidence intervals. In general, confidence intervals obtained for the algorithms are very small ($< 0.05$), hence, it is possible to rely on these algorithms to obtain a similar AUROC with 90% confidence level. LocIT obtained an average AUROC of 0.94 with a confidence interval width of 0.0032, meaning that the average may vary up to $\pm 0.0032$ outperforming baselines with high confidence.

Moreover, an additional analysis was performed to assess how statistically more significant the performance of LocIT is, compared to the other algorithms. We calculated the differences between LocIT's AUROC and baseline methods and employed a t-test. The p-values obtained were very small ($< 0.05$). The p-value obtained by comparing LocIT to the second best-performing method SSKNNO was $1.4e^{-8}$, indicating that LocIT's performance is significantly better than other baselines.

## VIII. CONCLUSION

Since obtaining extensively labeled data can be labor intensive, and requires domain knowledge, it may be too costly, especially, when working with big datasets. The results of our study show several benefits of the transfer learning approach utilized for event detection tasks:

- It yields an average increase in AUROC of approximately 13% compared to the best performing supervised learning algorithm, and 5% compared to the best performing unsupervised learning algorithm. The significant accuracy improvements were evident when relying on only 2% of labeled data corresponding to 20 characteristic events.
- The performance is less affected by the decrease in the number of available labels, and algorithm provides high performance even with only 20 representative labeled events used. In comparison, the supervised learning algorithms are infeasible for event detection in this domain when labels are very limited.
- The proposed approach is robust to temporal and locational options for splitting the PMU data. Consequently, it is feasible to leverage and transfer knowledge from historical PMU data to improve learning on future unlabeled instances, and to transfer selected labeled events from a specific set of PMUs to another set of PMUs. The reported results provide evidence that identifying a variety of event types, including line faults, transformer outage, and frequency events by a model that can be deployed to detect events in future PMU data while avoiding challenges faced by online learning.

**TABLE 5. Summarizes the average AUROC and their corresponding two-sided confidence interval, calculated at 90% confidence level.**

| Learning Type | Model | Average AUROC and Confidence Interval |
|---|---|---|
| Transfer Learning | LocIT | **0.94 ± 0.0032** |
| Semi-supervised | SSKNNO | 0.90 ± 0.0036 |
| | SSDO | 0.81 ± 0.0078 |
| Supervised | MLP | 0.74 ± 0.0058 |
| | LR | 0.72 ± 0.0074 |
| | KNN | 0.76 ± 0.0042 |
| | SVM | 0.72 ± 0.0199 |
| Unsupervised | kNNO | 0.75 ± 0.0037 |
| | iNNE | 0.85 ± 0.0113 |

## FUTURE WORK

The experiments conducted show that the proposed transfer learning approach is capable of detecting events by leveraging minimal labeled time windows from a related task within PMU data of the Western Interconnection of the U.S.A. Upon promising results reported in this paper, the proposed technique could be extended to leverage labeled time windows from the Western Interconnection of the U.S.A. and transfer learning to the Eastern Interconnection of the U.S.A. for event classification task. The challenge of this task revolves around different number of PMUs at two interconnections, resulting in different dimensions of feature vectors. Moreover, the distance metric used in the semi-supervised detector (SSKNNO) might be less effective for the Eastern Interconnection where a much larger number of PMUs were observed. Hence, implementation of an appropriate distance measure that is suitable for high dimensional feature vectors might be required to enhance the event detection task.

## 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 authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## REFERENCES

[1] NASPI Data Network Management Task Team, "NASPI 2020 survey of industry best practices for archiving synchronized measurements," North Amer. Synchrophasor Initiative, Tech. Rep. NASPI-2020-TR-024, Nov. 2020.

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

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–72, 2009.

[4] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, Jun. 2000, doi: 10.1145/335191.335437.

[5] V. Vercruyssen, W. Meert, and J. Davis, "Transfer learning for anomaly detection through localized and unsupervised instance selection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6054–6061.

[6] 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.

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

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

[9] 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, Aug. 2016.

[10] 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 *Proc. Power Syst. Comput. Conf. (PSCC)*, Jun. 2016, pp. 1–7.

[11] S. Brahma, R. Kavasseri, H. Cao, N. R. 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 Trans. Power Del.*, vol. 32, no. 1, pp. 294–301, Feb. 2017.

[12] 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.

[13] S. Wang, P. Dehghanian, and L. Li, "Power grid online surveillance through PMU-embedded convolutional neural networks," *IEEE Trans. Ind. Appl.*, vol. 56, no. 2, pp. 1146–1155, Mar. 2020.

[14] M. Khan, P. M. Ashton, M. Li, G. A. Taylor, I. Pisica, and J. Liu, "Parallel detrended fluctuation analysis for fast event detection on massive PMU data," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 360–368, Jan. 2015.

[15] M. Cui, J. Wang, J. Tan, A. R. Florita, and Y. Zhang, "A novel event detection method using PMU data with high precision," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 454–466, Jan. 2019.

[16] R. Yadav, A. K. Pradhan, and I. Kamwa, "Real-time multiple event detection and classification in power system using signal energy transformations," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1521–1531, Mar. 2019.

[17] A. Shahsavari, M. Farajollahi, E. M. Stewart, E. Cortez, and H. Mohsenian-Rad, "Situational awareness in distribution grid using micro-PMU data: A machine learning approach," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6167–6177, Nov. 2019.

[18] S. Jafarzadeh, N. Moarref, Y. Yaslan, and V. M. I. Genc, "A CNN-based post-contingency transient stability prediction using transfer learning," in *Proc. 11th Int. Conf. Electr. Electron. Eng. (ELECO)*, Bursa, Turkey, Nov. 2019, pp. 156–160.

[19] Z. E. Mrabet, D. F. Selvaraj, and P. Ranganathan, "Adaptive hoeffding tree with transfer learning for streaming synchrophasor data sets," in *Proc. IEEE Int. Conf. Big Data*, Los Angeles, CA, USA, Dec. 2019, pp. 5697–5704.

[20] Y. Zhang, X. Wang, Y. Luo, Y. Xu, J. He, and G. Wu, "A CNN based transfer learning method for high impedance fault detection," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Montreal, QC, Canada, Aug. 2020, pp. 1–5.

[21] J. Van Haaren, A. Kolobov, and J. Davis, "TODTLER: Two-order-deep transfer learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3007–3015.

[22] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," 2018, *arXiv:1812.11806*. [Online]. Available: http://arxiv.org/abs/1812.11806

[23] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 2, pp. 34–38, 2013, doi: 10.14569/IJARAI.2013.020206.

[24] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells, "Isolation-based anomaly detection using nearest-neighbor ensembles," *Comput. Intell.*, vol. 34, no. 4, pp. 968–998, Nov. 2018, doi: 10.1111/coin.12156.

[25] *Scikit-Learn—Machine Learning in Python*. Accessed: Jun. 12, 2021. [Online]. Available: https://scikit-learn.org/stable/

[26] X. Zhu and A. Goldberg, "A concise introduction to multiagent systems and distributed artificial intelligence," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.

[27] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Baumer, and J. Davis, "Semi-supervised anomaly detection with an application to water analytics," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, p527–536

[28] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "Systematic construction of anomaly detection benchmarks from real data," in *Proc. ACM SIGKDD Workshop Outlier Detection Description*, 2013, pp. 16–21.

[29] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, pp. 1–27, Oct. 2008.

[30] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, May 2000.

[31] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[32] J. L. Hodges, "The significance probability of the Smirnov two-sample test," *Arkiv Matematik*, vol. 3, no. 5, pp. 469–486, Jan. 1958.

**AMEEN ABDEL HAI** (Student Member, IEEE) received the B.Sc. degree in software engineering from the University of Northampton, U.K., in 2015, and the M.Sc. degree in computer science from Saint Joseph's University, PA, USA, in 2018. Currently, he is pursuing the Ph.D. degree in computer and information sciences with Temple University, PA, under the supervision of Dr. Zoran Obradovic. His main research interests include machine learning, data mining, and big data analytics.

**TATJANA DOKIC** (Member, IEEE) received the M.Sc. degree in electrical and computer engineering from the University of Novi Sad, Municipio de Novi Sad, Serbia, in 2012, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA, in 2019. Currently, she is an Assistant Research Engineer with Texas A&M University. Her main research interests include big data for power system applications, power system asset and outage management, weather impacts on power systems, time series analysis of PMU measurements, and smart grids.

**MARTIN PAVLOVSKI** (Member, IEEE) received the B.Sc. degree in electrical engineering and information technologies from Saints Cyril and Methodius University of Skopje, Skopje, North Macedonia, in 2015, and the Ph.D. degree in computer and information science from Temple University, Philadelphia, PA, USA, in 2021, under the supervision of Dr. Zoran Obradovic. Currently, he is a Research Scientist at Yahoo!. His research interest includes machine learning from structured data.

**TAIF MOHAMED** (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Texas A&M University at Qatar, in 2018. Currently, she is pursuing the Ph.D. degree in electrical engineering with Texas A&M University, College Station, TX, USA, under the supervision of Dr. Mladen Kezunovic. Her main research interests include power system stability, applications of machine learning in power systems, and smart grids.

**DANIEL SARANOVIC** received the B.Sc. and M.Sc. degrees in mathematics from the Faculty of Mathematics, University of Belgrade, Belgrade, Serbia, in 2016 and in 2018, respectively. Currently, he is pursuing the Ph.D. degree in computer and information sciences with Temple University, PA, USA. His main research interests include machine learning, learning from spatio–temporal data, and data mining.

**MOHAMMAD ALQUDAH** (Student Member, IEEE) received the B.Sc. degree in computer engineering from Jordan University of Science and Technology, Irbid, Jordan, in 2012, and the M.S. degree in industrial and systems engineering from Binghamton University, NY, USA, in 2015. Currently, he is pursuing the Ph.D. degree in computer and information sciences with Temple University, PA, USA. His main research interests include machine learning, data mining, and big data analytics.

**MLADEN KEZUNOVIC** (Life Fellow, IEEE) received the Diploma Ing. degree from the University of Sarajevo, Sarajevo, Bosnia and Herzegovina, and the M.Sc. and Ph.D. degrees in electrical engineering from The University of Kansas, Lawrence, KS, USA, in 1974, 1977, and 1980, respectively. He 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 the ''Power Engineering Research Center'' Consortium. For over 25 years, he has been the Principal Consultant of XpertPower Associates, a consulting firm specializing in power systems data analytics. His expertise is in protective relaying, automated power system disturbance analysis, computational intelligence, data analytics, and smart grids. He has authored over 600 articles, given over 120 seminars, invited lectures, and short courses, and consulted for over 50 companies worldwide. He is a CIGRE Fellow and Honorary and Distinguished Member. He is a Registered Professional Engineer in Texas City.

**ZORAN OBRADOVIC** (Senior Member, IEEE) is a Distinguished Professor, the Center Director at Temple University, an Academician with the Academia Europaea (the Academy of Europe), and a Foreign Academician with 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–temporal data analytics motivated by applications in healthcare management, power systems, earth, and social sciences. He is also an Editorial Board Member at 13 journals and was the general chair, the program chair, or the track chair of 11 international conferences. He is the Steering Committee Chair of the SIAM Data Mining Conference. He is the Editor-in-Chief of the *Journal of Big Data*.