Classification of Disturbances and Cyber-Attacks in Power Systems Using Heterogeneous Time-Synchronized Data

Shengyi Pan, *Member, IEEE*, Thomas Morris, *Senior Member, IEEE*, and Uttam Adhikari, *Student Member, IEEE*

***Abstract*—Visualization and situational awareness are of vital importance for power systems, as the earlier a power-system event such as a transmission line fault or cyber-attack is identified, the quicker operators can react to avoid unnecessary loss. Accurate time-synchronized data, such as system measurements and device status, provide benefits for system state monitoring. However, the time-domain analysis of such heterogeneous data to extract patterns is difficult due to the existence of transient phenomena in the analyzed measurement waveforms. This paper proposes a sequential pattern mining approach to accurately extract patterns of power-system disturbances and cyber-attacks from heterogeneous time-synchronized data, including synchrophasor measurements, relay logs, and network event monitor logs. The term common path is introduced. A common path is a sequence of critical system states in temporal order that represent individual types of disturbances and cyber-attacks. Common paths are unique signatures for each observed event type. They can be compared to observed system states for classification. In this paper, the process of automatically discovering common paths from labeled data logs is introduced. An included case study uses the common path-mining algorithm to learn common paths from a fusion of heterogeneous synchrophasor data and system logs for three types of disturbances (in terms of faults) and three types of cyber-attacks, which are similar to or mimic faults. The case study demonstrates the algorithm's effectiveness at identifying unique paths for each type of event and the accompanying classifier's ability to accurately discern each type of event.**

***Index Terms*—Common paths, cyber-attack detection, disturbances, symmetric and unsymmetrical faults, synchrophasor data and device log mining.**

I. Introduction

SITUATIONAL awareness technologies have been studied and continuously improved for decades. The need to continue situational awareness improvements is motivated by recent power disturbances, which have led to large-scale blackouts [1]. A power-system disturbance, such as a transmission line fault, can initiate a chain of reactions, which lead to a cascading blackout if timely actions from operators are not taken. Poor visibility across the power system may also cause the significance of an event to be misunderstood and lead to incorrect control actions by operators in control centers. Additionally, as power systems increasingly depend on communication infrastructures to provide the wide-area monitoring and control, power systems are exposed to the threat of cyber-attacks. Cyber-attacks are another form of power-system contingency. Attacks that target power systems can exploit vulnerabilities in control devices and communication links to corrupt the control and measurement signals [2], [3], and interrupt monitoring algorithms [4]. Cyber-attacks that corrupt control and measurement signals can be disguised as power-system disturbances or control actions. Situational awareness technologies are needed, which distinguish between actual power-system disturbances related to natural events and cyber-attacks. The emphasis of this work is not on classifying disturbance types as quite a number of methods have been proposed to do so in the power system, but on distinguishing between disturbances and cyber-attacks. First, in the case that a cyber-attack impersonates a disturbance or control action, proper classification will lead to proper response. Classifying a cyber-attack as a disturbance or control action can lead to improper response and cause an outage or other negative impacts on the power system. Conversely, incorrectly classifying a disturbance or control action as a cyber-attack can lead to improper response within the information and communications technology (ICT) system. Second, a single classifier, which identifies all types of power-system contingences, is needed as an input to automated event response algorithms such as autonomic management frameworks, system integrity protection schemes (SIPS) [5], wide-area protection systems (WAPS) [6], and autonomic control frameworks [7]. This paper presents a methodology to mine the patterns for disturbances and cyber-attacks using a two-dimensional (2-D) graph from logged heterogeneous system data, to use the common paths in the graph as signatures of each type of modeled scenario, and finally, to classify specific disturbances and cyber-attacks. For proof of concept, in the paper, we consider disturbances as different types of line-to-ground and line-to-line faults.

Wide-area measurement systems (WAMS) couple time-synchronized voltage, current, and frequency measurements with high-speed networks to allow improved power-system situational awareness [8]. Compared with the traditional supervisory control and data acquisition (SCADA) systems that poll

Manuscript received July 01, 2014; revised September 18, 2014, December 18, 2014 and February 19, 2015; accepted March 22, 2015. Date of publication April 08, 2015; date of current version June 02, 2015. Paper no. TII-14-0692.

The authors are with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762 USA (e-mail: sp821@msstate.edu; morris@ece.msstate.edu; ua31@msstate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2015.2420951

field sensors once per several seconds, synchrophasor systems allow measurement of up to 120 samples/s. Synchrophasor data were used in this work for two reasons. First, the common path-mining algorithm uses a set of observed system states in temporal order as a signature for each observed event type. Synchrophasor measurements enable identification of fast-moving power-system events. Some power-system events involve fast-changing behaviors and may last only a few milliseconds [9]. For example, zone 1 faults are typically set to be cleared instantly. The presence of a fault and the system response of opening the breaker to clear the fault take just a few cycles. These events can be missed by slower speed measurement systems. Second, synchrophasor systems provide more accurate system-state visibility due to the use of time-synchronized measurements. The common path-mining algorithm can leverage this improved visibility to track events related to a single event from multiple sensors. The relatively high measurement frequency and time-synchronized characteristic offered by WAMS create very large volumes of data and enable various applications including wide-area protection schemes (WAPS), and SIPS [5], [6], [10]. The common path-mining algorithm is not dependent on synchrophasor systems. Common path mining requires the ability to observe sequences of events. Other devices such as fault data recorders or meters may potentially be substituted to detect events of interest. Using synchrophasor data alone is not enough to detect cyber-attacks. For example, a cyber-attack can mimic a real fault by first injecting false measurements, then tripping the relay. Such mimicry cannot be detected with synchrophasor data alone. The status of other power-system components such as relays and breakers is also available as time-synchronized data via synchrophasor systems [10]. Combining synchrophasor data with other system logs such as relay status log and network event monitor logs can extend the situational awareness capabilities provided by a synchrophasor system to detect cyber-attacks. However, this creates the challenge of how heterogeneous data sources can be merged to train and use such a classifier. This paper provides a solution to this problem by proposing a data-mining approach that leverages the timestamped data to extract temporal patterns, which can be used to describe system behavior related to disturbances and cyber-attacks. Henceforth, disturbances and cyber-attacks are collectively referred to as scenarios.

In this work, a pattern for a scenario is presented as a common path that consists of a sequence of system states in temporal order. A system state in a common path is made up of multiple instantaneous readings from available sensors from the system. One advantage of the common path is that it overcomes the difficulty in analyzing time-domain waveforms by discovering the critical system states across very short time intervals (in milliseconds). The automatic process of discovering common paths is introduced by using a case study in a simulated three-bus two-line transmission system. For this work, a case study is provided, which considers disturbances including symmetric and asymmetric faults and different cyber-attacks that mimic the single-line-to-ground (1LG) fault to confuse operators in the control center. The cyber-attacks studied in this work belong to masquerading and/or man-in-the-middle (MITM) attacks that target physical devices such as phasor measurement units (PMU) and relays. These attacks may originate from a compromised node in control center, sending control commands or measurement packets covered by legitimate source IP addresses and legal packet formats. As such, it is assumed that the masquerading packets cannot be detected by traditional network intrusion detection systems. Validation of the common path-mining algorithm is based on simulated data because actual synchrophasor data are not available for researchers due to the proprietary nature of data, confidentiality issues, and lack of proper sharing mechanism among researchers and institutes. Additionally, datasets captured from utilities contain a limited number of scenarios. This limits diversity in the dataset. Some power-system scenarios are rare, especially cyber-attacks. Hardware-in-the-loop (HIL) simulation allows targeted dataset creation with realistic scenarios captured from the same commercial devices found in utilities. The same datasets used in this work have also been used in [11] for synchrophasor data-mining research.

This work has three primary contributions that distinguish it from existing methods. First, this work demonstrates a new classifier capable of distinguishing power-system disturbances and cyber security attacks that interrupt power-system control actions and mimic real disturbances. Compared to a similar work in [11], the method described in this paper provides precise classifications of fault types and the types of cyber-attacks with similar accuracy. Second, this work uses the common path-mining algorithm to mine fused heterogeneous data and create common paths for each known event type. The common path-mining algorithm uses less memory when compared to traditional data mining methods that require data to be mapped into memory before mining. The smaller memory requirement is achieved via a preprocessing step, which compresses the massive time-synchronized data into a sequence of system states, aka paths, which require considerably less memory than storing all time-synchoronized measurements associated with an event. Third, power systems are dynamic in nature, which leads to minor variations in system state for known scenarios. The classifier presented in this paper learns by parsing datasets marked with scenario type. The training process results in an ordered sequence of system states, i.e., a path, representing each unique instance of a scenario found in the dataset. To avoid overfitting, the common path-mining algorithm was developed to discover critical states shared by similar paths representing the same scenario. The result of the common path algorithm is a merged set of paths representing all scenarios in the dataset. The classifier matches monitored state-transition patterns to common paths of known scenarios to provide a specific classification of the observed behavior.

The remainder of this paper is organized as follows. Section II presents related works including an overview of other data-mining approaches used for classification of power-system disturbances or cyber-attacks. Section III discusses the methodology, the process of common path mining, and the classifier training and validation phases. Section IV introduces the case study test bed, test data, and test data preprocessing

procedure. Section V presents the classification results of three experiments. Section VI concludes this work and proposes future work.

II. Related Works

Current research on applying data mining to synchrophasor data for power-system fault and disturbance classification can be found in [12] and [13]. The K-nearest neighbor algorithm was used to classify three phase faults (3LG), voltage oscillation, and voltage sag scenarios in [11]. The algorithm accuracy is not provided in [12]. Hoeffding Tree-based stream data mining is used in [13]. This approach was able to classify 3LG and 1LG faults grouped for binary classification with greater than 90% accuracy. Both [12] and [13] used simulated power-system data. Both [12] and [13] propose methods to mine synchrophasor data. However, both are designed for power-system measurement data only and do not incorporate any other types of system information. By only considering measurement data, it is impossible to detect cyber-attacks such as fault replay or command injection attacks in which valid measurements or control commands are replayed. The work described in this paper fuses synchrophasor data and control system log information to allow precise classification of power-system faults and cyber-attacks.

Multiple traditional data-mining algorithms were used to classify power-system faults and cyber-attacks in [11]. The authors of [11] used the same dataset for algorithm validation as that used for this paper. The traditional data-mining algorithms were able to differentiate between power-system disturbances and cyber-attacks. However, the traditional data-mining algorithms were not able to classify specific fault and cyber-attack types within each large category.

Many other data-mining approaches have been developed to extract signatures and classify power-system disturbances, but they have no ability to detect cyber-attacks. Many such approaches classify power-system disturbances in the time domain. Decision trees were used to classify power-system disturbances in [14] and [15]. Statistical characteristics of power-system frequency were used in [16] to represent the signatures of power-system disturbances. Many works have applied neural networks to classify faults. In [17] with the help of wavelet transforms, current phase is decomposed and fed into a particle swarm optimization-based neural network for fault classification. A Chebyshev neural network is examined in [18] on current signals to evaluate the fault classification performance. In [19], the neural network is integrated with a wavelet transform multiresolution analysis technique to extract patterns for faults in shipboard power systems using energy variation of fault signals. In [20], the authors used a neural network with current waveforms and data from digital fault recorders to classify faults, normal maintenance operations, and power-quality disturbances. The works above all propose batch processing data-mining approaches to learn patterns for power-system events. These methods are not suitable for synchrophasor measurement data because batch processing requires all data to be read into memory to learn patterns. A single PMU can generate two million daily samples of data and multiple PMU can quickly exhaust available memory resources. The method proposed in this paper distinguishes itself from batch processing data-mining approaches by compressing fused synchrophasor and system log information into a set of system-state transitions, which minimize memory requirements during the training step. Furthermore, the same compression scheme is used during the classification step allowing the use of pattern matching to support real-time classification.

The work presented in this paper uses a sequential data-mining approach to classify patterns from sequences of events. Sequential data mining is better suited for high-velocity and high-volume synchrophasor data streams because synchrophasor data are discrete data but continuous in time. Additionally, the common path-mining algorithm presented in this paper can learn to classify traditional power-system contingencies, such as faults, and cyber-attacks against power systems which masquerade as traditional contingencies.

Machine-learning approaches have also been applied to detect cyber-attacks against power systems, but they do not consider power-system fault detection. In [21], detection rules were derived by manually specifying allowable ranges for different system measurements using domain expert knowledge. Such specification-based methods have been shown to have high detection accuracy; however, the manual effort required to develop such a decision tree is too great to apply to a problem on the scale of power-system protection. Other works have been found, which provide intrusion detection for synchrophasor systems, but they still do not provide power-system fault detection. An intrusion detection system (IDS) was proposed, which uses white lists to detect invalid network behaviors based on a synchrophasor network protocol specification [22]. A second proposed IDS uses timing and data-volume information to identify data-integrity attacks against synchrophasor systems [23]. However, by looking only at protocol format, timing, and data-volume information, these methods are not able to detect insider attacks, e.g., the command injection from a valid machine where the network packets have legitimate format, valid timing, and data-volume information. In [24], the authors manually created rules using the industrial state modeling language (ISML) to track SCADA system states. Nader *et al*. used a kernel machine-learning method to model SCADA system normal behavior, in order to detect machine failures and intrusions [25]. Due to a lack of attack data, only system normal behavior was learnt, and therefore, the authors were not able to test detection of attacks.

This paper presents a data-mining technique to develop signatures of multiple types of power-system faults and cyber-attacks. The resulting signatures provide a hybrid specification, which specifies both normal reactions to faults and symptoms of cyber-attacks. The data-mining algorithm presented in this paper has the distinct advantage requiring far less system expertise to create signatures.

The data-mining technique used in this paper uses the mining sequential patterns' technique which discovers patterns of activity from time-ordered data. The mining sequential patterns' concept was first presented in [26] as a method to perform market basket analysis. Mining sequential patterns was used to discover patterns in clinical client-care management process

data that consist of patient records and log data over a period of treatment time in [27]. This technique was extended in [28] by employing a 2-D Bayesian network to graphically represent patterns in Hemodialysis processes, which consists of a sequence of medical activities over time. In order to discover patterns, a patient's physiological "state" is defined using clinical log data and patient records (e.g., body temperature, weight, mood, etc.). The pattern is therefore represented as contiguous transitions of states in a 2-D graph. Classification was made using the learnt patterns.

For this work, the frequent pattern (FP)-growth algorithm as used to mine for frequent sequential patterns. FP-growth reduces the cost of searching for frequent sequences by adopting a divide-and-conquer strategy [29]. As demonstrated in [30], FP-growth algorithm outperforms several popular frequent pattern-mining algorithms in run time, and therefore, it was chosen for this work. Frequent pattern mining is traditionally used for market basket analysis, a method to build associations between commonly purchased items at a store. In this paper, frequent pattern mining is used to identify associative relationships between observed power-system states related to a particular event type or scenario.

Compared with peer works, this work is unique in that we propose a data-mining algorithm that can learn patterns for both power-system disturbances and cyber-attacks from heterogeneous data including synchrophasor measurements and device logs from multiple locations in the power system. Learnt patterns are translated into common paths. Common paths are used as signatures for pattern recognition. This approach enables a fast low-memory process for detecting power-system contingencies and cyber-attacks. It is possible to use separate classifiers for power-system event detection and cyber intrusion detection. However, for attacks which mimic power-system events, a supervisor process (a human or another algorithm) will be required to analyze outputs from the two separate algorithms to resolve conflicts. Combining power-system event detection and cyber-intrusion detection resolves this issue. Furthermore, this work is unique because it provides a mechanism for precise classification of power-system disturbances and cyber-attacks which attempt to mimic the same disturbances. Such precise classification enables automated response algorithms which will lead to a more reliable power system.

III. Common Path Mining

A. Sequential Events for a Power-System Scenario

Power-system scenarios can be described as an ordered sequence of measureable events. For example, Fig. 1 depicts phase a current magnitude during a 1LG fault on a transmission line. The current magnitude can be quantized into three ranges: high, normal, and low which are represented by dark gray, white, and light gray rectangles shading Fig. 1. When the system is in a normal state, the current stays in the normal range, marked as node A in Fig. 1. When the 1LG fault occurs, current increases to the high range via node B. The protection scheme will operate two relays, $R1$ and $R2$, at both ends of the transmission line to open breakers and isolate the fault. Current

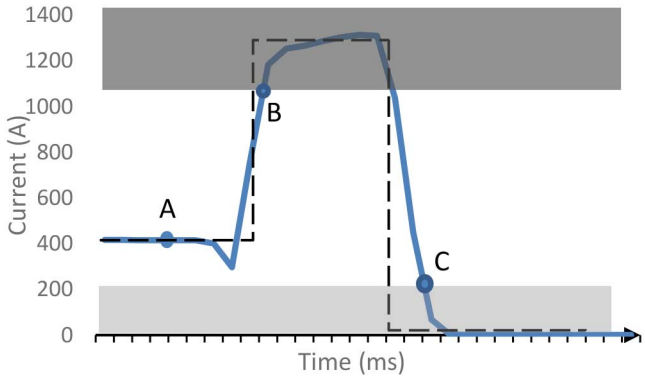

Fig. 1. Ideal versus actual 1LG fault and protection system response.

magnitude then drops through node C to zero. If following six notations are used to denote six events: "$I_{R1} = H$" as node "B," meaning "Current measured by $R1$ increases to High;" "$I_{R2} = H$" for "Current measured by $R2$ increases to High;" "$R1 = \text{Trip}$" for "Relay $R1$ trips;" "$R2 = \text{Trip}$" for "Relay $R2$ trips;" "$I_{R1} = 0$" as node "C" for "Current measured by $R1$ drops to Zero;" "$I_{R1} = 0$" for "Current measured by $R2$ drops to Zero." The timestamps of 1LG fault and resulting protection scheme operation can be represented by expression (1) where $t(\cdot)$ stands for the timestamp of corresponding events

$$\begin{aligned} t_{(I_{R1}=H)} &= t_{(I_{R2}=H)} < t_{(R1=\text{Trip})} \\ &= t_{(R2=\text{Trip})} < t_{(I_{R1}=0)} = t_{(R2=0)}. \end{aligned} \tag{1}$$

Expression (1) assumes a fault which appears at both relays at the same time and assumes that both relays operate at the same time. In fact, the fault may occur at different locations along the line leading to variations in the time each relay observes the fault and variations in relay operation time. Power systems are dynamic. In Fig. 1, the dashed line shows an ideal waveform of current magnitude during a fault and the solid line graphs a waveform captured from real-time digital simulator (RTDS) simulation of a 1LG fault. The actual waveform includes multiple variations from the ideal waveform. A power system's response to load variation, fault location variation, and transient behaviors results in irregular waveforms. Such variations are reflected as dispersions in the timestamps of node B and node C for different instances of the same scenario. The dispersion in timestamps can be seen not only in the events related to the current magnitude but also in the events related to other features. Fig. 2 shows box plots of timestamps of six events for three fault scenarios and one scenario where relays $R1$ and $R2$ are tripped by attackers. Fig. 2 (X-axis) is the set of observed events. The box plots represent 40 instances of each scenario. To provide an ordered sequence, the timestamp of the first event in a sequence was subtracted from timestamps of all later events in the sequence. The box plots and the interconnecting edges of a scenario are depicted using the same color. As shown in Fig. 2, events take place in temporal order. Event timestamps vary due to system dynamics. For each scenario, a track can be drawn by connecting box plot medians. The tracks shown in Fig. 2 generally agree with expression 1. Expert knowledge can be used to create similar expressions for all known system behaviors.

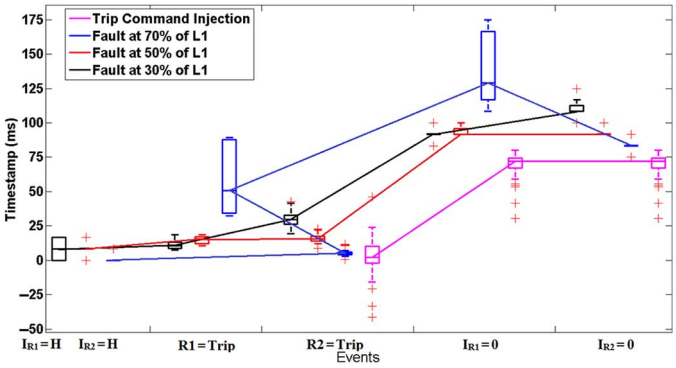

Fig. 2. Distribution of timestamps for events.

However, time variation prevents these from serving as signatures for classification. This leads to the need for a graph to describe an ordered set of events describing a scenario while comprehending the variation in timestamps.

Tracks are an ordered list of events with measurements where each vertex is an event measured at a single sensor. The classifier presented in this paper uses paths which are an ordered list of system states where a state is snapshot of measurements from all available sensors at a given time instant. The steps taken to convert heterogeneous data collected during a scenario into a path will be introduced in the next section. Path vertices are states and path edges are transitions between states. Paths are a means for providing stateful monitoring of the system. The training process performed to create paths is subject to overfitting due to the time variations seen in Fig. 2. In the overfitting case, different instances of the same scenario may have different paths. A technique for common path mining is provided below to identify shared critical states between a set of paths for a scenario leaving a common path that comprehends the variation in timestamps.

B. Common Path Mining

The mining common path algorithm is used to derive common paths for each scenario of interest. Common paths are maximal frequent sequences found in the set of paths observed for a given scenario. Common paths can be used as a signature for a scenario and pattern matching can be used to classify system events by scenario type.

The *Common Path-Mining Algorithm* is described below. The algorithm must be run once for each scenario of interest.

Algorithm 1. Common Path Mining

Input: Raw data from power system for the scenario of interest

Output: A common path

Step 1) *Collect raw data*. Raw data consist of measurements and timestamps. Expressions 2–4 show three measurements and timestamps from two example sensors, $s1$ and $s2$. Each sensor may measure a single item or multiple items and each sensor provides a timestamp. For example, $s1_1$ denotes the measurements from sensor $s1$ at timestamp 1; $s2_{a_{1.5}}$ is a measurement from sensor $s2$ for item a at timestamp 1.5. Many instances of raw data are needed for each scenario. All sensors must have a measurement at time 0

$$s1_1 = (s1_{a_1}, s1_{b_1}, \ldots, t_{s1_1}) \tag{2}$$

$$s2_{1.5} = (s2_{a_{1.5}}, s2_{b_{1.5}}, \ldots, t_{s2_{1.5}}) \tag{3}$$

$$s1_2 = (s1_{a_2}, s1_{b_2}, \ldots, t_{s1_2}). \tag{4}$$

TABLE I
MERGED RAW DATA

Sample	Sample	Sample	Sample	Timestamp
$\boldsymbol{s1_{a_0}}$	$s1_{b_0}$	$s2_{a_0}$	$s2_{a_0}$	t_{s1_0}
$\boldsymbol{s1_{a_1}}$	$s1_{b_1}$	$s2_{a_0}$	$s2_{a_0}$	t_{s1_1}
$\boldsymbol{s1_{a_2}}$	$s1_{b_2}$	$s2_{a_{1.5}}$	$s2_{a_{1.5}}$	t_{s1_1}

Step 2) Merge raw data. The various sensor data must be merged into a single database. Since each sensor may take measurements at different times, the merged data must be time aligned. The highest frequency sensor is used as a baseline. Slower rate sensor data are merged into the baseline sensor's log file. Measurements from slower sensors, which are between timestamps of the baseline sensor, are delayed to the next baseline sensor timestamp. Table I shows an example of merged raw data based on the input data from expressions 2–4.

Step 3) *Quantize data*. Data from sensors can take many forms: real numbers, integers, Boolean values, etc. Data must be quantized to reduce state space. For sensors with real and integer values, data can be quantized into numbered ranges. For example, voltage and current can be quantized into low (0), medium (1), and high (2) ranges according to two thresholds r_1 and r_2. The choice of r_1 and r_2 requires expert knowledge. Expression 5 provides an example quantization mapping for measurement s

$$q(s_i) = \begin{cases} 0, & \text{if } s_i \le r_1 \\ 1, & \text{if } r_1 \le s_i < r_2 \\ 2, & \text{if } s_i \ge r_2. \end{cases} \tag{5}$$

Step 4) *Map to states*. A state is a set of merged and quantized sensor measurements and a timestamp. Expression 6 shows an example state

$$S_j = (q(s1_i), q(s2_i), \ldots, t_i). \tag{6}$$

States are stored in a state database. Only unique states are stored in the database and the state index j is incremented for each unique state. The state database is common for all instances of all scenarios.

After mapping to states, an instance of a scenario can be represented as an uncompressed path. Expression 7 shows an uncompressed path representing the kth instance of scenario U

$$U_k = (S_0, S_0, S_1, S_2, \ldots). \tag{7}$$

Step 5) *Compress data into paths*. The uncompressed paths are compressed by removing sequences of states that do not change leaving just one instance of that state. This step provides a compression, which reduces memory usage and results in a tuple that represents all state transitions for the system. The state transitions correspond to events. The result of compression is a path that represents a single instance of a scenario. A path $\mathbf{P}_i$ is a list of observed system states arranged in temporal order according to their timestamps ordered by increasing time

$$\mathbf{P}_i = (S_1, S_2, \ldots, S_n). \tag{8}$$

Dynamic systems will have many paths for the same scenario due to minor variations in sampled data resulting from measurement inaccuracies and changes in the larger system. For example, power systems are large interconnected systems. Changes outside the monitored portion of the power system may lead to variability in observed measurements, the same scenario in the monitored portion of the power system.

Step 6) *Mine common paths*. The common path-mining process uses the mining frequent patterns' algorithm FP-growth [24] to mine for common sequences of states from $\mathbf{P}$. Among these frequent sequences, the maximal sequences are used as common paths. Note that there could be more than one common path for a scenario.

A sequence α is a subset of a path, i.e., $\alpha \subseteq \mathbf{P}$. Sequence α is denoted by $\{\mathbf{S_{i+1}}, \mathbf{S_{i+2}}, \ldots, \mathbf{S_{i+m}}\}$. A path $\mathbf{P}$ contains sequence α if all of the elements in α appear in $\mathbf{P}$ in the same order. In a set of sequences, a sequence α is *maximal* if α is not contained in any other sequences.

Let $\mathbf{G}$ be the set of all observed paths for a scenario Q, so that $\mathbf{G} = \{\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P_n}\}$ where n is the number of observed paths for Q. A path supports sequence α if the sequence is contained in the path. The number of paths that contain the sequence α is defined as *support count*. Given the support count for the sequence α and the total number of paths in $\mathbf{G}$, the *support* for the sequence α can be defined as the support count divided by the total number of paths in $\mathbf{G}$.

A sequence whose support is greater than a minimum support threshold is called a *frequent sequence*. A common path for scenario Q is a frequent sequence whose support is greater than a minimum support threshold and is maximal. There may be multiple common paths for a single scenario. Common paths reflect the states that occur most frequently for a scenario.

Table II provides examples of different paths for one scenario. Each path is mined from a measured event database. T represents the timestamps for states. P_1 represents the ideal case for a path representing a scenario. P_2 matches P_1, except that a subset of states is delayed. This may occur due to timestamp variation in events or due to system dynamics. P_3 contains an extra state. Dynamics may occur when a feature oscillates during a state transition. P_4 represents the case when a path is similar, but a state is different from the ideal case. This could happen when an event in a state (i.e., S_2) does not occur due to the variation in the timestamp, which results in a different state (i.e., S_{11}). P_5 represents an error path. In the error path, no sequences match the ultimate common path.

TABLE II
EXAMPLE PATHS FOR A SCENARIO

	T_1	T_2	T_3	T_4	T_5	T_6	
P_1	S_1	S_2	S_3	S_4	S_5		Ideal case
P_2	S_1		S_2	S_3	S_4	S_5	Delayed states
P_3	S_1	S_{10}	S_2	S_3	S_4	S_5	Extra states
P_4	S_1	S_{11}	S_3	S_4	S_5		Modified states
P_5	S_{21}	S_{22}	S_{23}	S_{24}	S_{25}		Error path

For this example, $\mathbf{G} = \{P_1,\ P_2,\ P_3,\ P_4,\ P_5\}$. If the minimum support threshold is set to 60%, the output of FP-growth algorithm for $\mathbf{G}$ is a set of frequent sequences which meet the minimum support threshold including $\{S_1,\ S_2,\ S_3,\ S_4,\ S_5\}$ and $\{S_1,\ S_3,\ S_4,\ S_5\}$. For this example, $\{S_1,\ S_2,\ S_3,\ S_4,\ S_5\}$ is maximal and is therefore the common path. The sequence $\{S_1,\ S_3,\ S_4,\ S_5\}$ is not maximal because it is contained in $\{S_1,\ S_2,\ S_3,\ S_4,\ S_5\}$. If the minimum support threshold is changed to 70%, the maximal frequent sequence will be $\{S_1,\ S_3,\ S_4,\ S_5\}$. Since only one sequence meets the threshold, it is maximal.

Algorithm 2. Classification Using Common Paths, cp

Input: PUT (path under test)
Output: C (Class)
1: For each common path, cp_i, in cp:
2: **If** $\mathrm{cp}_i \subseteq$ PUT:
3: Add cp_i to CCP (list of candidate common paths)
4: Filter CCP for cp_i with maximal length
5: If $\mathrm{size(CCP)} == 1$
6: Return class = look-up class of CCP_0
7: Else return class = unknown.

The common path is used as a signature during classification. Changing the minimum support threshold changes the number of states in a common path and can affect classification accuracy. It is not necessary to find a common path which matches the ideal path, rather the goal is to find a common path which is unique for a scenario and which leads to maximum classification accuracy. For a noisy system, a shorter common path may yield better classification results.

Common paths are signatures which can be compared to compressed paths for classification. Algorithm 2 shows the process for classifying a single PUT. Algorithm 2 can be used for real-time classification as shown in Algorithm 3. The while loop in Algorithm 3 executes at the frequency of the sensor with the highest sample rate. The merge raw data step in Algorithm 1 is not needed in for real-time processing since the value of all sensors can be read in each loop iteration. The steps {collect raw data, quantize data, and map to state} are the same as the steps of the same name in Algorithm 1. The function call class (PUT) in Algorithm 3 refers to calling Algorithm 2.

Algorithm 3. Real-time Classification Using Common Paths

Input: Real-time raw data
Output: Classified scenario type

```
1: While(true)
2:   While (state! = steady_state)
3:     Collect raw data sample
4:     Quantize data
5:     Map to state
6:     If (state_i! = state_{i-1})
7:       Add state_i to path PUT
8:   class = class(PUT)
```

Algorithm 3 can be implemented in a daemon to monitor a power system in real time. The definition of steady state will vary by system and can be measured by a lack of state change over a user-defined period of time.

For the proof of concept described later in this paper, the common path-mining algorithm was verified by collecting raw data in advance and using Algorithm 2 to classify paths.

The rest of this paper presents a case study that applies the mining common path algorithm to a three-bus two-line transmission system for classifying four types of power-system symmetric and unsymmetrical faults and three cyber-attacks scenarios.

IV. Power-System Test Bed

A real-world power system is dynamic and consists of thousands of buses, loads, transmission lines, and other components. The power-system operation goes through various states and is a continuous process. The three-bus two-line transmission system used in this work is modified from the IEEE nine-bus three-generator system [31] according to our simulation requirements. Although this system is relatively small, it captures the essence of the larger power system and is small enough to be comprehensible in every detail. Multiple instances of the classifier proposed in this work would be deployed to monitor sections of a power system. The case study system uses commercial PMU and relays from two major vendors. The test bed and datasets exhibit behaviors of a real power system, yet fit into the resources available in the lab in terms of hardware and software limitations. Because the three-bus two-line transmission system is capable of varying generation from two sources, varying load, simulating faults on two transmission lines at locations with 1% increments, simulating loss of a transmission line due to control action or fault, and of multiple cyber-attacks, it is adequate for proof of concept of this work. The transmission system used for HIL simulation for this work is shown in Fig. 3.

A. Simulated Scenarios

The power-system disturbances and three types of attacks simulated for this work are described as follows.

1) Power-System Faults: In this work, we consider symmetric and unsymmetrical faults in a power system as the examples of disturbances. A power-system fault is a condition

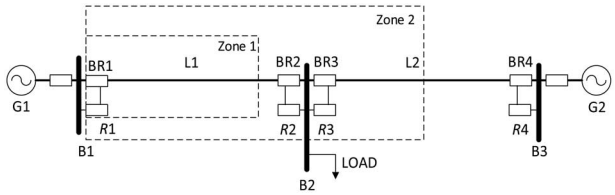

Fig. 3. Three-bus two-line transmission system for case study.

where the system voltage, current, and frequency are abnormal. Typically, 1LG faults, double-line-to-ground (2LG) faults, three-line-to-ground (3LG) faults, and line-to-line (LL) faults represent greater than 95% of faults in a power system [31]. In this work, for proof of concept, we simulated phase-a-to-ground fault for 1LG faults, phase-a-b-to-ground faults for 2LG faults, phase a-b-c-to-ground fault for (3LG) faults, and phase-a-to-b LL fault for LL faults.

2) Trip Command Injection Attack: Trip command injection attacks create contingencies by remotely sending unexpected relay trip commands from an attacker's computer to relays at the ends of a transmission line. The trip command injection attack used for this work closely mimics the 1LG fault. The attack was implemented against relay $R1$ and $R2$ by replaying relay trip commands captured from Modbus over Transmission Control Protocol (TCP) network traffic. However, we assume that these commands are sent from a compromised legitimate computer, such that these commands cannot be detected by a network event monitor as attacks since they are from a valid source and have valid formats. The two relay trip commands open the breakers at the ends of transmission line L1. This attack stresses the system by forcing L2 to carry more power flow, which may cause cascading failures in a power system. However, for this work, cascading failures were not simulated. The trip command injection attack instances were created under random load conditions in the same range used for faults.

3) Aurora Attack: The Aurora vulnerability refers to potential harm caused to a generator by intentionally opening and closing a breaker near the generator in rapid succession [33]. In this work, an aurora cyber-attack was simulated, which periodically sends opening–closing commands to relays that cause the breaker on the transmission line to open and close at a very fast pace.

4) 1LG Fault Replay Attack: The 1LG fault replay attack attempts to emulate a valid fault by altering system measurements to mimic a 1LG fault followed by sending an illicit trip command from a compromised computer to relays at the ends of the transmission line. This attack may lead to confusion and potentially cause an operator to take invalid control actions. A Python script is used to initiate an MITM attack between the hardware PDC and the historian. The attack replays synchrophasor measurements from a valid 1LG fault and then replays commands to trip the relays on the affected line.

B. Test Bed Architecture

The HIL test bed shown in Fig. 4 was used to simulate the distance protection scheme on the three-bus two-line

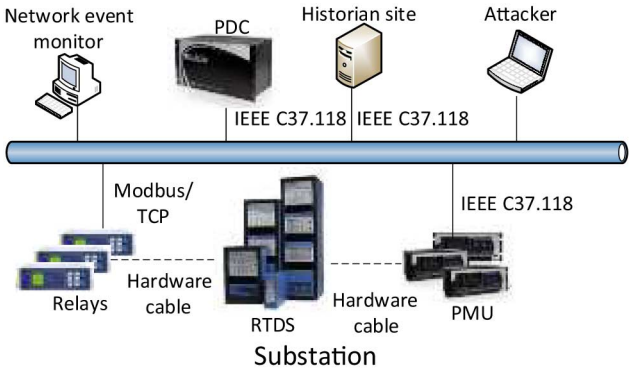

Fig. 4. Hardware in the loop test bed.

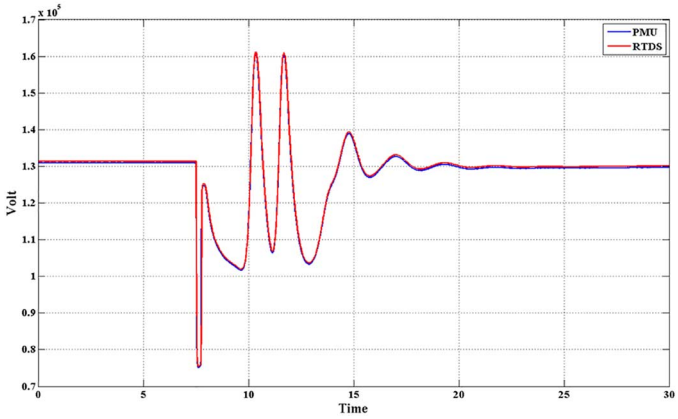

Fig. 5. Comparison of voltage measured by PMU and RTDS.

transmission system and implement the faults and cyber-attacks scenarios. The RTDS was used to simulate transmission lines, breakers, generators, and load. Four physical relays were wired to the RTDS in a HIL configuration. The relays implemented a two-zone distance protection scheme. The relays trip and open the breakers once a fault occurs on a transmission line. Fault logics for different types of faults were created in RSCAD and then the faults were implemented in the RTDS. Prior to each implementation of a fault, the system load was randomized in the range of 200–399 MW. Each fault instance was implemented at a random location in 1% increments from 10% to 90% of line L1.

The relays used in this work are the GE-D60 and SEL-421. Both are digital relays with integrated PMU functionality. However, PMUs and relays were drawn separately in Fig. 4. The PMUs stream real-time synchrophasor measurement data, using the IEEE C37.118 protocol at a rate of 120 samples/s, to the PDC. Then, aggregated synchrophasor data are forwarded to the OpenPDC software. The electrical parameters from RTDS simulation and PMU measured values were compared. The current transformer (CT) and potential transformer (PT) ratios of the simulated power-system model and of the actual hardware PMUs and the scaling factors of I/O components were adjusted to make the output from the RTDS simulation and PMU measurements close to identical. Validation of the HIL configuration required two steps. First, the power-system model described in [34] was implemented as a baseline. The RTDS simulation, PMU, and PDC voltage, current, and frequency were compared for dynamic and steady-state conditions described in [34]. Simulated and measured voltage, current, and frequency results matched with values noted in [34]. The baseline system was modified to create the three-bus two-line transmission system without altering the external hardware configuration. After altering the power-system model RTDS, PMU, and PDC voltage, current, and frequency continue to match. Fig. 5 shows overlapping voltage magnitude from the RTDS simulation and PMU. The voltages seen in simulation and at the PMU are the same throughout the simulated events. Current and frequency plots, as well as PDC measurements, also match but are not shown in the figure to save space.

A python script processes the synchrophasor measurement data received by OpenPDC into a comma-separated values format (CSV) file for each instance of a scenario. A row in the CSV file includes readings of frequency, current phasors, voltage phasors, and sequence components from the four PMUs, and a timestamp. Each CSV file is labeled with the instance number, scenario name, as well as load ranges and/or fault location at the moment the instance of the scenario occurs. The label is useful for grouping instances as will be discussed in Section V. The label is also used for training and classifier testing. The four relays were sources of timestamped relay state changes. There is also a network event monitor that logs any trip command packets sent to relays. All logs and synchrophasor measurement CSV files were stored by a historian.

For this work, simulation of all scenarios starts from a stable state and ends at a stable state. Faults last for 1 s and the relay closes the breaker 2 s after opening. Also, the distance protection scheme was simplified by disabling reverse time-delay backup and limiting the number of protection zones for each relay to 2. Each relay provides primary protection up to 80% of the line (Zone 1 protection) and backup protection (Zone 2 protection) up to 150% of the line. The trip time for Zone 1 protection is set to instantaneous, while the trip time for the Zone 2 protection is set to 20 cycles.

C. Test Data and Data Preprocessing

In total, 1023 instances of 1LG faults, 274 instances of 2LG faults, 584 instances of 3LG faults, 272 instances of LL faults, 274 instances of command injection attacks, 225 instances of aurora attack, and 703 instances of 1LG fault replay attack were simulated. Test data consist of the synchrophasor measurement CSV files, the four relay logs, and network event-monitor logs collected during all of these scenarios. The relay log that contains timestamp and corresponding event information (trip or nontrip) was extracted from the relays. The network event-monitor log contains timestamp and corresponding network events (trip command seen or not seen). Each CSV file contains tuples with 52 synchrophasor measurements as each PMU provides 13 measurements including voltage and current phasor magnitude (V_a, V_b, V_c and I_a, I_b, I_c), zero, positive and negative sequence voltage and current phasor magnitude (V_0, V_1, V_2 and I_0, I_1, I_2), and apparent line impedance (Z). A single CSV file has approximately 2000 tuples for an instance of a single scenario. Since the PMU streams at 120 samples/s,

2000 tuples correspond to 17 s of simulated system time per scenario. The test data were separated into training and testing datasets, each of which was the input to common path mining and classifier algorithms described in the previous section.

Rather than using all recorded input features from the dataset, only a portion of measurements was retained as selected features. In this work, the selected features contain relay status and the three-phase current magnitudes ($I_a,\ I_b,\ I_c$). Relay status was used as a feature because all cyber-attacks studied in this work maliciously trip relays via the network. The network event-monitor log was selected as one of the features for the same reason. The three-phase current magnitudes were selected because the current magnitudes of the three phases were the most significant measurements during symmetric and unsymmetrical faults. Other unselected measurements were discarded from the input data.

The measurement data from the PMU and relay log were merged into a single file for one instance of a scenario. The PMU current magnitude measurements were measured at 120 samples/s, while relay status occurs asynchronously. To merge the features, phase current was chosen as a reference and the relay status was up-sampled prior to merging.

Each feature was quantized into finite ranges. The phase currents were quantized into low, normal, and high ranges. The low range was 0–99 Amperes (A). The normal range was 100–1199 A. The high range was greater than 1200 A. The relay status was quantized into two values: 1) tripped; and 2) nontripped.

The aggregated features describe the system state at a given timestamp. A system state thus is a vector of a timestamp and features with quantized measurements. An example of state that describes relay $R1$ and $R2$ tripping due to high current magnitude can be represented as a vector $\text{Timestamp},\ I_{R1} = \text{High},\ I_{R2} = \text{High},\ R1 = \text{Trip},\ R2 = \text{Trip},\ \ldots$, where "$I_{R1} = \text{High}$" and "$I_{R2} = \text{High}$" in the vector represent high-current magnitudes measured by PMUs in $R1$ and $R2$. "$R1 = \text{Trip}$" and "$R2 = \text{Trip}$" in the vector represent relay trip status of the two relays. The time difference between two states is same as that between two rows, which is the reciprocal of the synchrophasor measurement rate; $1/120\ \text{samples/s} = 8.3\ \text{milliseconds (ms)}$. The timestamps of rows in the file were normalized by subtracting the time of the first row from all other rows. This causes all files for all scenario instances to start from time 0.

V. Evaluation

Three experiments were performed to validate the common path-mining algorithm. Experiment 1 classifies two classes, 1LG fault and command injection attack. This was an initial proof of concept to show that the algorithm can distinguish a fault from a single attack intended to mimic the fault. Experiment 2 repeated Experiment 1 with the fault labels preprocessed into groups by fault location and system load. This was done to show that the common path-mining algorithm can learn unique paths for sequences of events with very small differences. In Experiment 3, four types of short-circuit faults and three types of cyber-attacks were simulated. This experiment demonstrates that the common path-mining algorithm can learn paths for larger sets of symmetric and asymmetric faults and multiple different cyber-attacks. To test for overfitting, captured data from experiment 3 were used to test classifier accuracy using 10-round cross-validation. Experiment 3 was also repeated with varying PMU sample rates to show the effect of sample rate on classifier accuracy. For each experiment, the training phase that computes a set of common paths is described in Algorithm 1 in the previous section; the testing phase that classified testing paths is described in Algorithm 2.

TABLE III
Confusion Matrix for Experiment 1

	Fault	C. inj.
Fault	491	0
C. inj.	0	123
Unknown	28	4

A. Experiment 1

For the first experiment, approximately half of the test data for 1LG fault and command injection attack was randomly chosen as a training dataset, while the rest was used as a testing dataset. This resulted in 519 instances of 1LG fault and 127 instances of the command injection attack, which were used for training. Table III is a confusion matrix from Experiment 1.

For this work, accuracy, misclassification, and unknown rates were defined as follows. The accuracy rate is the percentage of instances correctly classified. Misclassification rate is the percentage of the instances of a class which were misclassified as another scenario. The unknown rate is the percentage of the instances of a scenario which were not classified as any scenario. Unknown instances either match no common paths or match more than one common path from more than one class.

For the first experiment, the overall classification accuracy was 95%. No instances were misclassified. A total of 5% of tested scenario instances were unknown. All unknown instances matched at least one fault and at least one command injection common path.

There were a total of 221 common paths found for the two scenarios: 203 for 1LG fault scenario and 18 for the command injection scenario. This high number of paths results from the dynamic nature of the power system. Fig. 6 is a plot of the fault location, from the perspective of relay $R1$, versus relay trip times for relays $R1$ and $R2$. Fig. 6 clearly shows zone 1 and zone 2 trip boundaries for both relays. Additionally, Fig. 6 shows that the relay trip times vary with fault location especially in the fault location region from 24% to 79% of the transmission line. The large number of common paths for the 1LG fault injection scenario is primarily due to this variation. System behavior also varies as the system load changes.

B. Experiment 2

Ideally, faults between 0% and 20% of the transmission line should have instant trip time for relay $R1$ and trip after 20 cycles for relay $R2$. Faults between 80% and 100% of the transmission line should trip after 20 cycles for relay $R1$ and instantly for relay $R2$. In the 21%–79% range, both relays

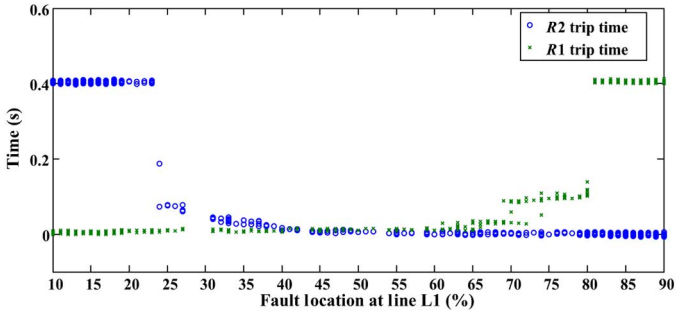

Fig. 6. Relay trip time versus fault location for relays $R1$ and $R2$.

should ideally trip instantly. Observed trip times match the ideal case for the 0%–20% and 80%–100% ranges. Note, the apparent impedance setting for zone 2 for relay $R2$ causes the zone 1-to-zone 2 transition to occur at approximately 23% of the line (77% of the line from relay $R2$'s perspective) instead of at the expected 20% of the line (80% of the line from relay $R2$'s perspective).

The trip times from 24% to 80% of the line are always instantaneous. Observed trip times tended to increase as the fault approached the zone 1 to zone 2 boundary points. To compensate for this observed behavior, the 1LG fault paths were grouped by fault location per the following groups: 10%–23%, 24%–29%, 30%–35%, 36%–40%, 41%–60%, 61%–65%, 66%–70%, 71%–80%, 81%–90%. Additionally, it was observed that trip times partially correlated with the system load. As a result, the 1LG fault class used in Experiment 1 was divided into multiple classes by fault location and load. Four load ranges were used: (200–249, 250–399, 300–349, 350–399 MW). This subdivided the 1LG fault class into $9 * 4 = 36$ subclasses.

The command injection attack class in Experiment 1 was also divided using four load ranges, which results in four command injection attack classes.

The extra step of subdividing the 1LG fault class and command injection attack results in a total of 40 classes. The training dataset and testing dataset in this experiment is the same as that used in Experiment 1.

Table IV is a confusion matrix for all scenarios for Experiment 2. As previously mentioned, the 1LG fault classes were divided by fault location and system load. To save space, the groups in the confusion matrix were combined to just show the fault location classes and one command injection class. An extra row (marked Unk. for unknown) was added to the confusion matrix to show instances of scenarios, which were not classified.

Experiment 2 classification accuracy, misclassification, and unknown rates can be viewed from multiple perspectives. The overall accuracy rate for the groups shown in the confusion matrix was 87.6%. Misclassification and unknown rates for the same groups were 9.1% and 3.3%, respectively. From the confusion matrix, the majority of misclassification occurred when 1LG fault groups were classified as members of a neighboring or nearby fault group. The unknown cases are separated into unknown instances, which resulted from an instance matching multiple fault common paths ("Unk. fault" in Table III) and

TABLE IV
CONFUSION MATRIX FOR EXPERIMENT 2

	10%–23%	23%–29%	30%–35%	36%–40%	41%–60%	61%–64%	65%–70%	71%–80%	81%–90%	C. inj.
10%–23%	191	3	0	0	0	0	0	0	0	0
23%–29%	0	4	0	0	0	0	0	0	0	0
30%–35%	0	0	4	0	0	0	0	0	0	0
36%–40%	0	7	0	2	6	0	0	0	0	0
41%–60%	0	0	0	2	41	2	0	0	0	0
61%–65%	0	0	0	0	5	10	0	0	0	0
65%–70%	0	0	0	0	8	3	14	4	0	0
71%–80%	0	0	0	0	0	0	1	38	18	0
81%–90%	0	0	0	0	0	0	0	0	135	0
C. inj.	0	0	0	0	0	0	0	0	0	127
Unk. fault	0	0	9	7	0	0	0	0	0	0
Unknown	4	0	0	0	1	0	0	0	0	0

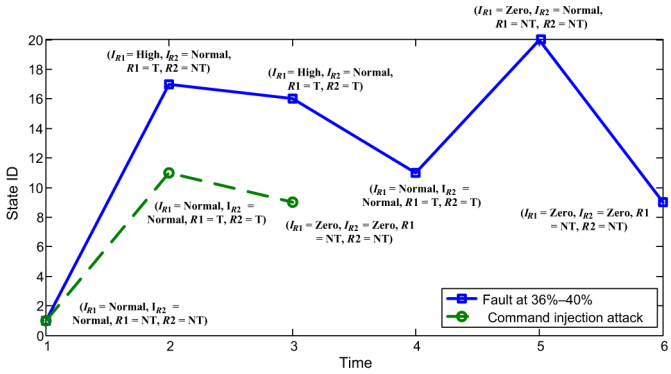

Fig. 7. 2-D coordinates documenting 1LG fault versus command injection attack common paths.

unknown instances which matched no common path. The 16 cases of faults, which matched common paths from more than one group, all occurred because both the (30%–35%) and the (36%–40%) shared a common path.

The intent of subdividing the 1LG fault class was not to classify 1LG faults by a specific fault location. Correctly classifying a fault as a fault is sufficient as many algorithms are available to provide fault location information. The accuracy rate when the fault location classes were combined into a single class is 96.7%. The misclassification rate was 0% and the unknown rate was 3.3%.

Common paths can be mapped into 2-D coordinates with the Y-axis indicating the state identification code (state ID) and the X-axis indicating normalized timestamps. An edge between two vertices represents the temporal transition between two states. Each vertex is marked with state information. Fig. 7 shows common paths for two scenarios, a 1LG fault in the 36%–40% fault location group and a command injection attack. Both the fault and command injection common paths start at the system normal state. These paths differ immediately because, for faults, the PMU will measure high current when a fault is present. This makes the second state of the fault common path high current detected at relay $R1$. The command injection attack occurs when there is no fault present. As such, the second state for the command injection attack has normal current at both relays, while both relays' status indicates a trip.

Fig. 8 shows common paths for two different 1LG fault locations. Note that not all features are displayed in the vertex

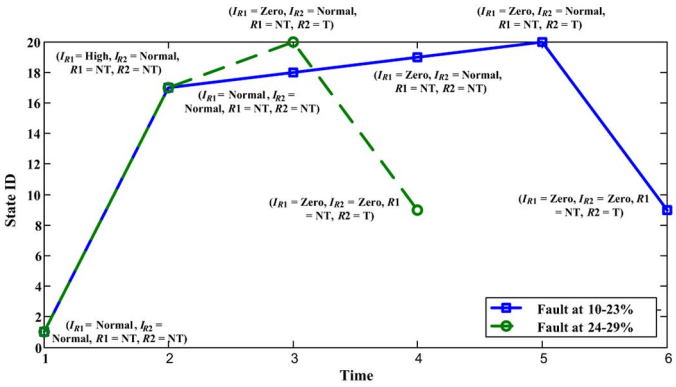

Fig. 8. 2-D coordinates comparing two common paths for 1LG faults of different locations.

labels. The 10%–23% fault is in relay $R2$ zone 2 and the 24%–29% fault is in relay $R2$ zone 1. This difference is the primary reason for different paths for the two fault subgroups.

Figs. 7 and 8 demonstrate that common paths contain the critical states for different scenarios. The primary contribution of the common path-mining algorithm is the ability to automatically learn unique paths for each scenario type from data.

Training and testing processing time and memory usage were measured using an Ubuntu Linux Virtual Machine with 3.5 GHz CPU and 2 GB memory. For Experiment 1, training required 202 s and 25.3-MB memory with approximately 2.5-GB time-synchronized data. Experiment 1 testing required 0.85 s per scenario instance. For Experiment 2, training required 205 s and 25.2-MB memory for the same amount of training data. Experiment 2 testing required 0.83 s per scenario instance. The difference in classification time between the two experiments is likely due to host computer load varying between experiments. Classification is a pattern-matching exercise similar to other pattern-matching technologies such as virus scanners or rule-based network intrusion detection systems. The classification testing for this work was not optimized for actual use in a real system. To build a real-time classifier, a program would be required to collect raw data samples, quantize data, and map data to states (steps 3–5 of Algorithm 3). The instances used for testing in this work each required approximately 17 s of wall clock time to occur in the system. For a synchrophasor system with 120 samples/s, there are 8.3 ms between samples. This time could be utilized to process samples and perform the comparison, and a decision tree architecture could be used to facilitate fast pattern matching.

C. Experiment 3

A third experiment was conducted for classifying four types of symmetric and unsymmetrical faults and three types of cyber-attacks. The training phase used the same methodology as Experiments 1 and 2. Validation in this experiment used 10-round cross-validation. In each round, half of the test data was randomly chosen as a training dataset and the remaining data were used as the testing dataset. Table IV is a combined confusion matrix for 10 rounds of validation for the 1LG, 2LG, 3LG,

TABLE V
CONFUSION MATRIX FOR FOUR TYPES OF FAULTS AND THREE CYBER-ATTACKS

	1LG flt.	2LG flt.	3LG flt.	LL flt.	Cmd. inj.	Aurora	Flt. replay
1LG flt.	5009	0	0	3	0	0	109
2LG flt.	6	1248	0	0	0	0	0
3LG flt.	86	11	2905	24	0	0	17
LL flt.	0	14	0	1089	0	0	0
Cmd. inj.	0	0	0	6	1380	0	124
Aurora	0	0	0	0	0	971	0
Flt. replay	0	0	0	0	0	0	3138
Unknown	177	58	15	238	0	159	97

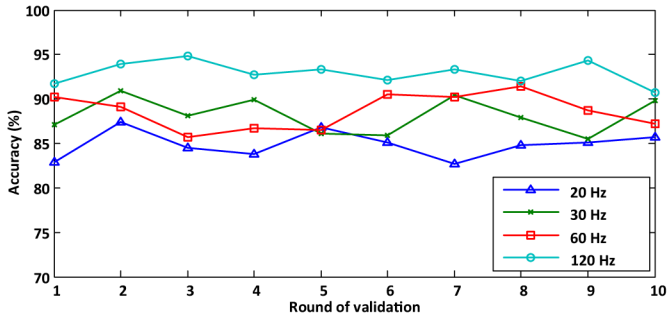

Fig. 9. Accuracy rates for 10-round cross-validation with different PMU streaming rates.

LL faults, command injection, Aurora, and fault replay attacks. Each entry in the table sums up numbers for 10 rounds in the corresponding location.

The total number of classifications made in Table V is 16 885, of which 15 740 instances are correctly classified. The average accuracy for the seven classes shown in Table V is 93.21%. Only 488 instances of faults (177 of 1LG fault, 58 of 2LG fault, and 15 of 3LG fault and 238 for LL) were classified as unknown, and only six instances of faults are misclassified as cyber-attacks. The lowest accuracy for an individual class or scenario type was for fault replay attacks. Fault replay attack classification accuracy was 90%. Fault replay attacks were misclassified as a fault for 3.6% of the tested instances and misclassified as a command injection attack for 3.5% of tested instances. The fault replay attack is intended to mimic a 1LG fault and as such is sometimes able to confuse the classifier. The fault replay includes elements from the command injection attack. This leads to similarities which cause occasional misclassification as a command injection attack. Table IV demonstrates that the classifier is able to distinguish faults and cyber-attacks.

The accuracy rate for 10-round validation when the PMU is sample rate at 20, 30, 60, and 120 Hertz (Hz) is plotted in Fig. 9. Classification accuracy is higher when the PMU is streaming at 120 Hz and lowest at 20 Hz. However, event at 20-Hz accuracy exceeds 80%. This is reasonable as higher PMU samples rates gives better visibility of the system states when fast-moving events, such as faults, are considered.

Table VI shows a comparison of classifier results from the Random Forest, JRip, Adaboost + JRip, and common path-mining algorithms. The values for Random Forest, JRip, and Adaboost + JRip are from the work described in [11] which

TABLE VI
COMPARISON OF COMMON PATH MINING TO OTHER ALGORITHMS

	Random forest	JRip	Adaboost + JRip	Mining common path algorithm
# of classes	3	3	3	7
Accuracy (%)	80	90	95	93
Precision	0.65	0.89	0.99	0.98
Recall	0.20	0.85	0.93	0.95
F measure	0.28	0.90	0.95	0.96

used datasets derived from the same test bed as this common path-mining algorithm evaluation for training and testing. Among the four algorithms, $\mathrm{Adaboost} + \mathrm{JRip}$ has the highest value in four metrics: accuracy, precision, recall, and F measure. Accuracy, precision, recall, and F measure for the common path-mining algorithm were computed from the results for Experiment 3 of this work. Note that the mining common path classifier uses seven classes and has similar performance to $\mathrm{Adaboost} + \mathrm{JRip}$ classifier with only three classes. The Random Forest, JRip, $\mathrm{Adaboost} + \mathrm{JRip}$ classifiers used the following classes: normal behavior, attack events, and natural events. The mining common path classifier classes were three separate attacks and four types of faults (natural events). The ability to make a precise classification versus a broad category while maintaining high accuracy makes the common path-mining algorithm promising. Such precise classification is necessary to quickly understand the root cause of events to enable automated response.

VI. CONCLUSION AND FUTURE WORK

The common path-mining algorithm creates common paths from heterogenerous data in the power system. A common path represents a set of critical states in which a system will step through in temporal order for a scenario such as a disturbance or a cyber-attack. Common paths can be used as signatures to classify power-system behaviors with high specificity. Such a classifier is a useful tool for use with automated system integrity protection systems and wide-area control systems, which include responses for both natural, equipment failure, and cyber-attack-related contingiencies.

Simple paths can be derived from monitored instances of sceanrios applied to a test bed. However, the transients present in time-domain measurement data lead to different paths for different instances of the same scenario. The common path-mining algorithm uses a sequential pattern-mining approach to overcome this challenge and common paths for the scenario.

To validate the correctness of the algorithm, a case study was performed, which applied the common path-mining algorithm and classifier to detect disturbances and cyber-attacks. The classifier provides a capability to accurately distinguish between different types of power-system faults and cyber-attacks including command injection, aurora attacks, and fault replay attacks. Three separate experiments were performed. The first experiment applied the common path-mining algorithm to data with two classes: 1LG fault and command injection. The second experiment adds an extra step prior to the training phase where the 1LG fault class is divided into a number of subclasses by taking advanatage of power-system domain expertise. The extra step of subdividing classes in training produces slightly better accuracy, misclassification, and unknown classification. Both experiments required similar training time, testing time, and memory usage. A third experiment was conducted using the same training as Experiment 2. Ten-round cross-validation was performed with varying PMU sample rates. The 10-round validation shows that the classifier has not overfit the data. Comparison of varying PMU sample rates shows that the highest accuracy is achieved with PMU sampled in 120 Hz. This is expected since faults are fast-moving events and the 120-Hz sample rate provides the most visibility of system-state changes.

This paper demonstrates a methodology to leverage synchrophasor measurements for power-system disturbance and cyber-attack detecion and highlights the promise of the minining common paths aglorithm. Future work includes applying the algorithm to larger systems with more known types of disturbances, control actions, and cyber-attacks.

The common path-mining algorithm was evaluated using a three-bus two-line transmission system. It is possible to scale up for larger systems by sampling system state from larger portions of a power system. Training and classification time will increase linearly as the number of tuples in a sample increases based on the property FP-growth algorithm [30]. This leads to an effective limit on the number of measured tuples used for one instance of the classifier. When this limit is reached, different portions of a power system can be monitored by separate instances of the classifier. Using multiple instances of the classifier leads to two potential future works. First, classifiers will have overlapping visibility. As such, a method will be needed to rationalize results from overlapping classifiers. Second, a partitioning scheme is needed to determine classifier boundaries.

REFERENCES

[1] H. Polzin and B. McMillan. (2012, Apr.). *Arizona-Southern California Outages on September 8, 2011*. Causes and Recommendations, Federal Energy Regulatory Commission and the North American Electric Reliability Corporation, Washington, DC, USA [Online]. Available: http://www.nerc.com/fileUploads/File/News/AZOutage_Report_01MAY12.pdf

[2] N. Falliere, L. O'Murchu, and E. Chien, "W32. Stuxnet Dossier, V 1.4," Symantec Corp., Mountain View, CA, USA, Tech. Rep. MS10-046, 2011 [Online]. Available: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[3] S. Ntalampiras, "Detection of integrity attacks in cyber-physical critical infrastructures using ensemble modeling," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 104–111, Nov. 2014.

[4] H. Lin, Y. Deng, S. Shukla, J. Thorp, and L. Mili, "Cyber security impacts on all-PMU state estimator—A case study on co-simulation platform GECO," in *Proc. IEEE 3rd Int. Conf. Smart Grid Commun.*, Nov. 2012, pp. 587–592.

[5] V. Madani *et al.*, "IEEE PSRC report on global industry experiences with system integrity protection schemes (SIPS)," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2143–2155, Oct. 2010.

[6] V. Terzija *et al.*, "Wide-area monitoring, protection, and control of future electric power networks," *Proc. IEEE*, vol. 99, no. 1, pp. 80–93, Jan. 2011.

[7] Y. Deng *et al.*, "Communication network modeling and simulation for Wide Area Measurement applications," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT)*, Washington, DC, USA, Jan. 16–20, 2012, pp. 1–6.

[8] R. Amgai, J. Shi, and S. Abdelwahed, "An integrated lookahead control-based adaptive supervisory framework for autonomic power system applications," *Int. J. Elect. Power Energy Syst.*, vol. 63, pp. 824–835, 2014.
[9] A. Bose, "Smart transmission grid applications and their supporting infrastructure," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 11–19, Jun. 2010.
[10] D. Bakken *et al.*, "Smart generation and transmission with coherent, real-time data," *Proc. IEEE*, vol. 99, no. 6, pp. 928–951, Jun. 2011.
[11] R. Borges *et al.*, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2014, pp. 1–8.
[12] M. Al Karim, M. Chenine, K. Zhu, and L. Nordstrom, "Synchrophasor-based data mining for power system fault analysis," in *Proc. 3rd IEEE PES Int. Conf. Exhib. Innov. Smart Grid Technol. (ISGT Europe)*, Oct. 2012, pp. 1–8.
[13] N. Dahal, "Synchrophasor data mining for situational awareness in power systems," Ph.D. dissertation, Dept. Elect. Comput. Eng., Mississippi State Univ., Starkville, MS, USA, 2012.
[14] P. K. Ray, S. R. Mohanty, N. Kishor, and J. P. S. Catalao, "Optimal feature and decision tree-based classification of power quality disturbances in distributed generation systems," *IEEE Trans. Sustain. Energy*, vol. 5, no. 1, pp. 200–208, Jan. 2014.
[15] A. Rodriguez *et al.*, "A Decision Tree and S-transform based approach for power quality disturbances classification," in *Proc. 4th Int. Conf. Power Eng. Energy Elect. Drives (POWERENG)*, May 2013, pp. 1093–1097.
[16] W. Gao and J. Ning, "Wavelet-based disturbance analysis for power system wide-area monitoring," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 121–130, Mar. 2011.
[17] J. Upendar, C. P. Gupta, G. K. Singh, and G. Ramakrishna, "PSO and ANN-based fault classification for protective relaying," *IET Gener. Transmiss. Distrib.*, vol. 4, no. 10, pp. 1197–1212, Oct. 2010.
[18] B. Y. Vyas, B. Das, and R. P. Maheshwari, "Improved fault classification in series compensated transmission line: Comparative evaluation of chebyshev neural network training algorithms," *IEEE Trans. Neural Netw. Learn. Syst.*, Oct. 2014 [Online]. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6920088&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6920088
[19] W. Li, A. Monti, and F. Ponci, "Fault detection and classification in medium voltage DC shipboard power systems with wavelets and artificial neural networks," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 11, pp. 2651–2665, Nov. 2014.
[20] K. Silva, B. Souza, and N. Brito, "Fault detection and classification in transmission lines based on wavelet transform and ANN," *IEEE Trans. Power Del.*, vol. 21, no. 4, pp. 2058–2063, Oct. 2006.
[21] R. Mitchell and I.-R. Chen, "Behavior-rule based intrusion detection systems for safety critical smart grid applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1254–1263, Sep. 2013.
[22] Y. Yang *et al.*, "Intrusion detection system for network security in synchrophasor systems," in *Proc. IET Int. Conf. Inf. Commun. Technol.*, 2013, pp. 246–252.
[23] B. Sikdar and J. Chow, "Defending synchrophasor data networks against traffic analysis attacks," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 819–826, Dec. 2011.
[24] A. Carcano *et al.*, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 179–186, May 2011.
[25] P. Nader, P. Honeine, and P. Beauseroy, "{l_p}-norms in one-class classification for intrusion detection in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2308–2317, Nov. 2014.
[26] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, Mar. 1995, pp. 3–14.
[27] F. Lin, S. Chen, S. Pan, and Y. Chen, "Mining time dependency patterns in clinical pathways," *Int. J. Med. Informat.*, vol. 62, no. 1, pp. 11–25, 2001.
[28] F. Lin, C. Chiu, and S. Wu, "Using Bayesian networks for discovering temporal-state transition patterns in hemodialysis," in *Proc. 35th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2002, pp. 1995–2002.
[29] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2012.
[30] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," *Data Min. Knowl. Discovery*, vol. 8, no. 1, pp. 53–87, Jan. 2004.
[31] P. Anderson, *Analysis of Faulted Power Systems*. Hoboken, NJ, USA: Wiley, 1995.
[32] H. Saadat, *Power System Analysis*, 3rd ed. Alexandria, VA, USA: PSA Publishing, 2010.
[33] M. Zeller, "Myth or reality—Does the Aurora vulnerability pose a risk to my generator?," in *Proc. 64th Annu. Conf. Protective Relay Eng.*, Apr. 11–14, 2011, pp. 130–136.
[34] H. Ferrer and E. Schweitzer, *Modern Solutions for Protection, Control, and Monitoring of Electric Power Systems*. Oregon, IL, USA: Quality Books Inc., 2010, pp. 57–104.

Shengyi Pan (S'12–M'14) received the B.Eng. degree in electronic information engineering from Fuzhou University, Fuzhou, China, in 2008; the M.Sc. degree in data communications from the University of Sheffield, Sheffield, U.K., in 2009; and the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA, in 2014.

From 2010 to 2014, he was a Research Assistant with the Department of Electrical and Computer Engineering, Mississippi State University, where his research focused on smart grid cyber security and data-driven intrusion detection technologies. He is currently a Software Engineer with MaxPoint Interactive Inc., Morrisville, NC, USA, for big data application development in Internet digital advertising. His research interests include smart grid technologies, cyber security, data mining, and bid data technologies.

Thomas Morris (M'06–SM'08) received the B.S. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 1994, and the M.S. and Ph.D. degrees in computer engineering from Southern Methodist University, Dallas, TX, USA, in 2001 and 2008, respectively.

He joined Mississippi State University, Starkville, MS, USA, in 2008. He currently serves as an Associate Professor of Electrical and Computer Engineering, Associate Director of the Distributed Analytics and Security Institute (DASI), and the Director of the Critical Infrastructure Protection Center (CIPC). His research interests include cyber security for power systems and industrial control systems.

Uttam Adhikari (S'11) received the B.S. degree in electrical engineering from Tribhuvan University, Kathmandu, Nepal, in 2005, and he is currently pursuing the Ph.D. degree in electrical and computer engineering at Mississippi State University, Starkville, MS, USA.

His research interests include cyber-physical system modeling and simulation, wide-area measurement systems, data mining, and cyber security in smart grid.