Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems

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

***Abstract*—Synchrophasor systems provide an immense volume of data for wide area monitoring and control of power systems to meet the increasing demand of reliable energy. The construction of traditional intrusion detection systems (IDSs) that use manually created rules based upon expert knowledge is knowledge-intensive and is not suitable in the context of this big data problem. This paper presents a systematic and automated approach to build a hybrid IDS that learns temporal state-based specifications for power system scenarios including disturbances, normal control operations, and cyber-attacks. A data mining technique called common path mining is used to automatically and accurately learn patterns for scenarios from a fusion of synchrophasor measurement data, and power system audit logs. As a proof of concept, an IDS prototype was implemented and validated. The IDS prototype accurately classifies disturbances, normal control operations, and cyber-attacks for the distance protection scheme for a two-line three-bus power transmission system.**

***Index Terms*—Cyber-attacks, data mining, distance protection, intrusion detection system (IDS), power system, synchrophasor system.**

I. Introduction

THE NEXT generation power system, also known as the smart grid, will rely on advanced technologies such as synchrophasor systems for wide area monitoring and control in order to meet the increasing demand of reliable energy. While in the past, power system components were isolated, they are now interconnected via information infrastructure, e.g., Ethernet, and therefore are under the threat of cyber-attacks. Due to the critical role that the power system plays in our society, there is a common agreement that the electric power grid needs to be better secured to ensure continually available power for the nation [1]. There have been multiple documents from different organizations which provide recommendations and guidelines for industry to better secure their facilities [2], [3]. However, the U.S. Government Accountability Office (GAO) has concluded that current guidelines are insufficient to securely implement the smart grid and the GAO calls for research and development to improve upon current security mechanisms [4].

Manuscript received July 14, 2014; revised January 15, 2015; accepted February 17, 2015. Date of publication March 18, 2015; date of current version October 17, 2015. This work was supported by the U.S. National Science Foundation under Grant DUE-1344369 and Grant DUE-1315726. Paper no. TSG-00716-2014.

The authors are with Mississippi State University, Starkville, MS 39762 USA (e-mail: sp821@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/TSG.2015.2409775

Intrusion detection systems (IDSs) identify activities that violate the security policy of a computer system or network. IDS are a necessary complement to preventive security mechanisms such as firewalls because IDS detect attacks that exploit system design flaws or bugs and IDS provide forensic evidence to inform system administrator's reactions to cyber-attacks [5]. The increasing coupling of cyber infrastructure and physical devices of the smart grid makes a traditional host-based IDS inadequate because host-based IDS monitor host in the system individually while power system control algorithms such as the distance protection scheme usually involve multiple devices at multiple locations. Therefore, new IDS should have the ability to take multiple data sources into account and perform stateful monitoring at the system level. Manually building a stateful IDS is a knowledge intensive task which requires vulnerability analysis and manual creation of rules and patterns which describe attacks and normal behaviors. The manual development process results in limited scalability and updates are slow and expensive.

This paper documents a systematic and automated approach to building a hybrid IDS that leverages features of signature-based and specification-based IDS. The IDS classifies system behaviors over time as specific disturbances, normal control operations, or cyber-attacks. Sequences of critical states, called common paths, provide a specification or signature for each scenario. A fundamental ingredient of the IDS presented in this paper is a data mining technique that aggregates synchrophasor measurement data and audit logs from multiple system devices to learn the common paths. The automatic approach eliminates the need to manually analyze and manually code patterns and is able to handle very large amounts of data.

Common paths are signatures of events present in a training database. Common paths are also specification since they describe expected system behaviors related to normal expected system behaviors and cyber-attacks behaviors. The IDS matches a temporal set of monitored system states to common paths to make a classification. Behaviors which do not match a common path are considered unspecified events and are either zero-day attacks or unknown system behaviors.

A case study is included to demonstrate that the proposed IDS provides high detection accuracy for both known and unknown scenarios and thus is suitable for mission critical environments such as power systems.

The rest of this paper is organized as follows. Section II reviews related works. An overview of the test bed and simulated power system scenarios is presented in Section III.

Section IV introduces the procedure to construct the proposed IDS. Experiments and results are discussed in Section V. The conclusion is provided in Section VI.

II. Related Works

A. IDS for Smart Grid

In recent years, the emergence of the smart grid has motivated research into a variety of IDS techniques. People with different backgrounds have created various IDS that focus on different aspects of the smart grid. One type of IDS research focuses on intelligent electronic device (IED) security within the smart grid [6], [7]. This type of IDS is usually host-based and thus only identifies attacks against a single IED/network appliance in the system based on its intended behaviors. While host-based IDS secure individual devices in the smart grid, they do not provide stateful monitoring at the system level. More advanced IDS of this type consider behaviors of multiple devices within the system to obtain system level detection. Mitchell and Chen [8] proposed a rule-based IDS for the electric grid by considering the behaviors of three types of physical devices in the electric grid: 1) head-ends; 2) distribution access points/data aggregation points; and 3) subscriber energy meters. Readings from 22 sensors from the three types of devices were used as state components. The method quantized each of the 22 components into a limited number of ranges. Three state machines with 3456, 1728, and 3456 states were manually built for the three devices and the state machines act as specifications for the three types of devices. Manual construction of such an IDS is cost prohibitive and dies not scale for larger power systems. Additionally, changes to system behaviors require updating the specification state machines via the manual process.

Network-based IDS leverage communication traffic in the information infrastructure of the smart grid to detect cyber-attacks. IDS can leverage trust systems which monitor communications to and from a device [24] to validate communications and limit command and control actions to those approved by the trust system. Yang *et al.* [9] proposed an IDS for synchrophasor systems that detects cyber-attacks by using white lists of packets with legitimate source IP addresses, correct packet formats, and legal values for fields. The Yang IDS was evaluated for man-in-the-middle (MITM) and denial of service attacks against synchrophasor devices using the IEEE C37.118 protocol. Zhang *et al.* [10] proposed a distributed IDS that analyzes communication traffic at different network levels of the smart grid including home area networks, neighborhood area networks, and wide area networks. An intelligent module was deployed at each level to classify malicious data and possible cyber-attacks using data mining algorithms. These modules then communicate to provide a system level view of the communication network to improve the detection accuracy. Hadeli *et al.* [11] proposed an anomaly detection technique for industrial control systems that whitelists legitimate communication patterns extracted from different industrial control system protocols available in the system. The Hadeli IDS uses a system description file to provide a description of the overall expected communication patterns in the industrial control system. The IDS proposed by [9]–[11] can detect malicious changes to network traffic, but all three IDS fail to detect malicious payload that results in invalid changes to the physical system. For example, Hadeli *et al.*'s [11] method cannot detect an injected but otherwise valid command to trip a protection relay from a valid IP address which will take a transmission line out of service and cause a blackout. A specification-based IDS was developed to track sequential events in an advanced metering infrastructure (AMI) [12]. A manually constructed state machine was used to extract legitimate sequential system states from two AMI protocols and devices status. To prove the correctness of the state machine, a model checking technique was used to verify the specifications. This IDS is not applicable for use with transmission systems because transmission systems have far more control actions and disturbances than AMI. As such, manually building such a state machine would be very expensive.

Other proposed IDS for smart grid leverage power system theory. For instance, Valenzuela *et al.* [13] used optimal power flow programs to detect cyber-attacks which alter system measurement data to cause the power flow to be dispatched erroneously. Talebi *et al.* [14] proposed a mechanism for identification of bad data attacks in a power system using weighted state estimation. Although these works are all proven capable of detecting altered data, these IDS are limited to one type of attack and cannot be extended to detect other attacks against power systems.

B. Accuracy of Specification-Based IDS

The detection accuracy of specification-based IDS depends on how accurately the specifications describe system behaviors. A promising way to improve the accuracy of specifications is through the use of data mining. A data mining technique was applied to an IDS framework proposed by Lee *et al.* [15] that combined signature-based IDS and anomaly-based IDS. Data mining programs were applied to a large volume of log data to learn attack signatures and normal behavior patterns and automatically create detection rules. Lee *et al.* [15] showed that the signatures for attacks and patterns for system normal behaviors created using their data mining technique are accurate by comparing their detection results to all other participants in the Defense Advanced Research Projects Agency intrusion detection evaluation program prepared by MIT Lincoln Laboratories. Lee *et al.*'s [15] IDS was originally designed for stateless IDS therefore it cannot be directly applied to specification-based IDS. A new data mining algorithm must be developed to discover sequential events for specifications.

C. Data Mining Techniques for Learning Specifications

A specification for a scenario contains a sequence of execution events or system states. The nature of specifications requires the data mining technique applied to the proposed IDS to be able to mine sequential patterns and identify the dependent relationship between events. 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 algorithm was first mentioned in [16]. Lin *et al.* [17] applied it to discover patterns in clinical client care management process data that consists of patient records and log data over a period of treatment time. This technique was extended in [18] by employing a Bayesian network to graphically represent patterns of different hemodialysis processes which consists of a sequence of patients' physiological states that are snapshots of clinical log data and patient records, e.g., body temperature, pulse rate, etc. In Lin *et al.*'s [18] work, states were assigned with probabilities for the purpose of prediction.

For the work presented in this paper, the FP-growth [19] algorithm was used in the training process to mine frequent sequential patterns from power system data. FP-growth is an implementation of frequent item set mining. A common example of frequent item set mining is market basket analysis in which stores attempt to find associative relationships among products purchased by multiple customers, such as finding products often purchased together. Common path mining is similar to market basket analysis except common path mining finds system states which are commonly found together in a set. Common path mining also preserves temporal order of the system states.

III. Common Path Mining

This paper uses the concept of a common path to represent the patterns encoded in a fusion of time stamped sensor data. A common path consists of a sequence of critical system states in temporal order. Describing the common path mining algorithm requires definitions of the concepts of state, feature, sequence, and path.

A state is used to represent a system's instantaneous status. A state consists of a set of observed system measurements or features f as well as a normalized time stamp (TS), i.e., $\mathbf{S} = \{\text{TS}, f_1, \ldots, f_n\}$. The value of a feature is read from a sensor. The possible values for a feature are in a range called its domain. A feature that has continuous values in its domain should be discretized to finite ranges to avoid an infinite state space.

A path $\mathbf{P}$ is a list of observed system states arranged in temporal order according to their TSs, namely, $\mathbf{P}_i = \{\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_n}\}$, ordered by increasing time. A sequence s is a subset of a path, i.e., $s \subseteq \mathbf{P}$. We denote a sequence s by $\{\mathbf{S_{i+1}}, \mathbf{S_{i+2}}, \ldots, \mathbf{S_{i+m}}\}$. A path $\mathbf{P}$ contains sequence s if all of the elements in s appear in $\mathbf{P}$ in the same order. In a set of sequences, a sequence is maximal if the sequence is not contained in any other sequences.

Let $\mathbf{G}$ be the set of all observed paths for a scenario Q so $\mathbf{G} = \{\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P_n}\}$ where n is the number of observed paths for Q. A path supports sequence s if the sequence is contained in the path. Support can be defined as a metric in which the support of sequence s is the percentage of paths in $\mathbf{G}$ that contain sequence s. A common path for scenario Q is any 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 I
Example Paths for a Scenario

	T_1	T_2	T_3	T_4	T_5	T_6	
P1	S_1	S_2	S_3	S_4	S_5		Ideal Case
P2	S_1		S_2	S_3	S_4	S_5	Delayed States
P3	S_1	S_{10}	S_2	S_3	S_4	S_5	Extra States
P4	S_1	S_{11}	S_{12}	S_4	S_5		Modified States
P5	S_1	S_{22}	S_{23}	S_{24}	S_{25}		Error Path

The common path mining algorithm consists of six steps. The first five steps create paths, $\mathbf{P}$, for each instance of a scenario. First, raw data is collected from various sensors in the system. Second, raw data is fused or merged into a single database. Sensors may measure at different times and frequencies. Lower frequency sensor data is up sampled so that all high frequency measurements are maintained. Third, measurements which are continuous are quantized to minimize the total number of possible states in a database. Expert knowledge is used to design ranges for each sensor. A database is a table with columns for each sensor and rows representing the state of the system at increasing TSs. In the fourth step, the database is parsed to find all unique states. Fifth, the database is compressed by merging all rows which are the same state. In the sixth step, all known paths for a scenario, the set $\mathbf{G}$, are processed with the mining frequent patterns algorithm FP-growth [19] to mine for frequent sequences of states. The support threshold is set via trial and error or using expert knowledge. Maximal frequent sequences are common paths for the scenario.

Example: Consider the set of paths shown in Table I. For the example $\mathbf{G} = \{\text{P1, P2, P3, P4, P5}\}$. If the minimum support threshold is set to 60%, the set of frequent sequences in $\mathbf{G}$ which meet the minimum support threshold includes $\{S_1, S_2, S_3, S_4, S_5\}$, $\{S_1, S_3, S_4, S_5\}$, and $\{S_1, S_4, S_5\}$. For this example, $\{S_1, S_2, S_3, S_4, S_5\}$ is maximal and is therefore the common path. The sequences $\{S_1, S_3, S_4, S_5\}$ and $\{S_1, S_4, S_5\}$ are not maximal because they are contained in $\{S_1, S_2, S_3, S_4, S_5\}$. Alternatively, if the minimum support threshold is changed to 70%, the set of sequences in $\mathbf{G}$ which meet the minimum support threshold includes only $\{S_1, S_4, S_5\}$. Since $\{S_1, S_4, S_5\}$ meets the threshold in this case, it is maximal and is a common path.

Table I also provides examples of possible types of paths that could be found in the dataset. P1 represents the ideal case for a path representing a scenario. P2 matches P1 except a subset of states are delayed. This may occur due to a measurement error or due to power system dynamics. P3 contains an extra state. Extra states may occur when a feature oscillates during a state transition. P4 represents the case when a path is similar but a state is different from the ideal case. This can happen when an event that should have occurred at T_2 occurs at T_3 instead, which mangles states S_2 and S_3 (they change to S_{11}, S_{12}). P5 represents an error path. In the error path no sequences match the ultimate common path.

The common path is used as a specification 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.

A common path for a single line-to-ground (SLG) fault should have a sequence of critical states representing "current going high," "relay trip," and "current falling to zero." The ability to find a common path is greatly dependent on the quality of paths in **G**. For example, if there are a many error paths in **G** it will be difficult to find sequences which meet the minimum support threshold.

Classification is performed by comparing observed system states to the states of known common paths. The path under test (PUT) is compared to all common paths. If $cp_i \subseteq$ PUT then cp_i is a candidate common path. The PUT is classified as matching the scenario of the maximal candidate common path from the set of candidate common paths. If more than one candidate common path are maximal the PUT is classified as unknown.

The rest of this paper presents a case study which applies the common path mining algorithm to a three-bus two-line transmission system for classifying 25 power system scenarios.

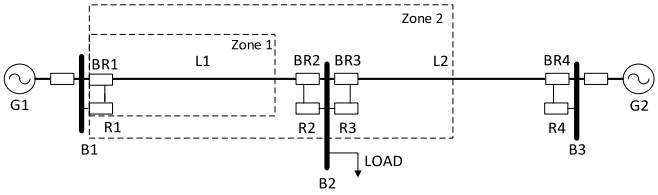

Fig. 1. Distance protection scheme in a three-bus two-line transmission system.

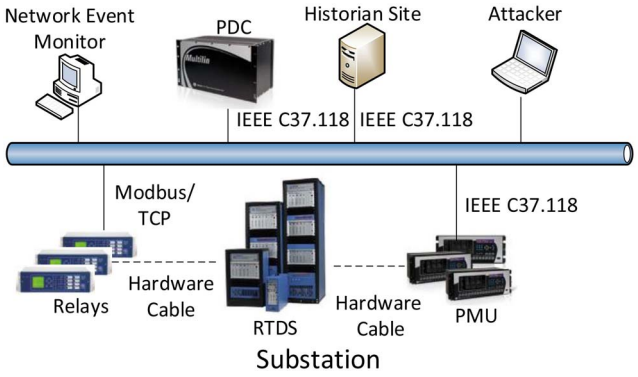

Fig. 2. Hardware in the loop test bed.

IV. Test Bed Architecture

A. Distance Protection for Transmission Lines

The distance protection scheme is the most popular scheme for protecting transmission lines. The principle of operation recognizes that the impedance of a high-voltage transmission line is approximately proportional to its length. This means the impedance "seen" by the relay during a fault is proportional to the distance between the point of fault and the relay. Distance relays are encoded with multiple protection zones. Each zone is assigned an apparent impedance threshold and a trip time. Relays have over lapping protection zones to provide system protection redundancy. One relay's zone 1 is part of another relay's zone 2 and so forth. For this case study, the distance protection scheme was simplified by disabling reverse time delay backup and limiting the number of protection zones for each relay to 2. Fig. 1 shows a three-bus two-line transmission system that is modified from IEEE four-bus three-generator system. Relay R1's zones 1 and 2 are shown as dashed line boxes. 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 in case that the primary protection fails. The trip time for zone 1 protection is configured to be instantaneous while the trip time for the zone 2 protection is time-delayed to avoid false tripping unless the primary relay fails.

B. Test Bed Architecture

A hardware-in-the-loop test bed, shown in Fig. 2, was used for power system scenario implementation and data generation. A real time digital simulator (RTDS) was used to simulate transmission lines, breakers, generators, and load. Four physical relays were wired to the RTDS in a hardware-in-the-loop configuration. The relays implemented the two zone distance protection scheme. The relays trip and open the breakers when a fault occurs on a transmission line. All relays included integrated phasor measurement unit (PMU) functionality to measure power system transmission line state, however, the PMUs were drawn separately in the graph because relays are controlled by Modbus/transmission control protocol (TCP) and PMUs stream synchrophasor measurements using the IEEE C37.118 protocol. The PMUs streamed real-time synchrophasor measurement data at a rate of 120 samples/s, to the phasor data concentrator (PDC) which aggregates network frames from multiple PMU and forwards the aggregated synchrophasor frames to the OpenPDC application. A set of scripts control the simulation by inducing random state changes, capturing measurements, labeling captured data by scenario type, and merging data from multiple sources into a single file. The synchrophasor measurement data includes of frequency, current phasors, voltage phasors, and sequence components. The four relays were sources of time stamped relay state changes. A signature-based IDS, Snort, runs on a PC to detect network activity. Snort provides alerts when it detects remote tripping command activities in the network. Snort, by itself, cannot distinguish between legitimate and illegitimate remote trip commands since they appear the same on the network. A control panel computer simulates energy management system (EMS) functionality. The EMS simulation was used to disconnect a transmission line for maintenance by remotely tripping relays via a Modbus/TCP network packet. An EMS log provides the TS of such a line maintenance event. For this paper, it is assumed that an attacker computer has successfully penetrated the utility's operational network and can launch cyber-attacks from a node on the operational network. Scenarios of power system disturbances, normal operations,

TABLE II
SIMULATES SCENARIOS

Scenario Name	Description
Q1-Q2	Single line-to-ground fault on L1 or L2 respectively.
Q3-Q4	Fault replay attack which mimics a valid fault on L1 or L2 respectively.
Q5-Q6	Remotely open both relays at both ends of transmission line (L1 or L2 respectively) for maintenance.
Q7-Q10	Command injection attack to remotely open one relay (R1, R2, R3, R4 respectively)
Q11-Q12	Command injection attack to remotely open two relays (R1 and R2, or, R3 and R4 respectively). This attack mimics the maintenance scenarios (Q5-Q6).
Q13-Q16	One relay (R1, R2, R3, R4 respectively) disabled during a fault on the line connected to that relay.
Q17-Q20	One relay (R1, R2, R3, R4 respectively) disabled during a maintenance event on a line connected to that relay.
Q21-Q22	Two relays (R1, R2, R3, R4 respectively) disabled during a fault on the line connected to those relays.
Q23-Q24	Two relays (R1, R2, R3, R4 respectively) disabled during a maintenance event on a line connected to that relay.
Q25	Normal system operation. No event occurring.

and power system cyber-attacks are applied against the simulated power system and its components. Data logs were captured from the synchrophasor system, relays, Snort, and the simulated EMS. All data logs were time stamped and with the name of the scenario being simulated.

C. Test Bed Scenarios

The power system scenarios used to train and validate the IDS presented in this paper have been grouped into three categories: 1) power system single-line-to-ground faults; 2) normal operations; and 3) cyber-attacks. Each category is described in this section with details. There are a total 25 scenarios each named with capital "Q" along with a number. The system load was randomized at the beginning of each scenario. Power system SLG faults belong to the shunt fault family and account for up to 70% of faults in a power system [20]. For this paper, only phase-*a*-to-ground faults were simulated as each phase to ground fault has similar characteristics. The phase-*a*-to-ground fault is abbreviated as "fault" in the rest of this paper. Table II provides a summary of the simulated scenarios used to validate the proposed IDS.

For the SLG fault scenarios (Q1 and Q2) the relay operates instantaneously for zone 1 and after a time delay for faults in zone 2. The auto-reclosing scheme models a high speed three-phase reclosing scheme [21] which closes the breaker after one second.

The SLG fault replay attacks (Q3 and Q4) attempt to emulate a valid fault by altering system measurements followed by sending an illicit trip command 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 was used to initiate a MITM attack between the hardware PDC and the OpenPDC application. The attacks replay synchrophasor measurements from a valid SLG fault then replay commands to trip the relays on the affected line.

The transmission line maintenance scenarios (Q5 and Q6) simulate the situation when an operator remotely trips relays to open breakers at both ends of a transmission line to take the line out of service for line maintenance. The operator initiated remote trip commands are recorded and time stamped in the control panel log.

Power system cyber-attacks may originate from insiders, amateur hackers, political activists, criminal organizations, governments, and terrorists. Cyber-attacks may appear as a nuisance or may bring the system to collapse. Attacks can be carried out from within power system substations, a control center, or in transmission and distribution infrastructures by exploiting weaknesses in physical security policies. Alternatively, attacks may take advantage of security flaws and vulnerabilities in software, devices, communication infrastructures, or communication protocols to electronically infiltrate power system operational networks. Three types of attacks are simulated: 1) relay trip command injection; 2) disabling relay function; and 3) SLG fault replay.

Relay trip command injection attacks (Q7–Q12) create contingencies by sending unexpected relay trip commands remotely from an attacker's computer to the relays at the ends of the two transmission lines. The trip command injection attack used for this paper closely mimics the line maintenance scenario. The malicious trip command originates from another node on the communications network with a spoofed legitimate IP address. Since the attack is not from the control panel computer there will be no record in the control panel log, however, the Snort network traffic monitor will detect this remote trip command.

The disabled relay attacks (Q13–Q24) mimic the effects of insiders taking illicit control actions or malware taking control of software systems to manipulate control devices. A python script accesses a relay's internal registers via Modbus/TCP commands sent from the attacker's computer which modify the relevant relay settings. The disabled relay attacks overlap fault and maintenance events. The final scenario, Q25, represents a stable system state. For this scenario, the load may change, but no other attacks, disturbances, or control actions are simulated.

Scenarios start and end with the system in a stable state. As such, all faults are cleared, transmission lines taken out of service for maintenance are returned to service, and all attacks end before the next scenario is simulated.

D. Test Data

Test data used for this paper includes data logs associated with 10 000 simulated instances of the 25 aforementioned scenarios. The data log is a comma separated file with labeled tuples that include 56 sensor measurements and a TS. The 56 data sources consist of 52 synchrophasor measurements; 13 from each relay location on Fig. 1. The synchrophasor data from a single relay consists of phase voltage and current phasor magnitude, zero, positive, and

negative sequence voltage, and current phasor and apparent line impedance. The synchrophasor data was sampled at 120 times/s. Relay status information, breaker events, Snort alerts, and control panel alerts were also logged. All logged data was merged into a single dataset.

An instance of a single scenario is represented by approximately 2000 tuples in the test data set. This corresponds to approximately 17 s of simulated system time per scenario. In total, the test data has more than two million tuples. Each tuple in the test data is labeled. Approximately, half of the test data was used to train the classifier and half was used to test classification accuracy. For this paper, 15 features were used; phase current magnitude measured at each relay, relay status for each relay, Snort alert status for each relay, and control panel remote trip status.

V. Training the IDS

This section documents the IDS construction process. First, the data formatting step converts input data logs to a measured events database (MED). Next, the specification learning steps process the MED to learn common paths, a unique set of system states in temporal order, for each labeled scenario. Finally, a graph is constructed which includes common paths for all scenarios.

A. Data Formatting

The first step of the data formatting process is feature quantization. Feature quantization requires domain expertise. Features with values which can take continuous values are mapped into finite ranges to limit state space size. Features which take discrete values are generally left unchanged unless the number of discrete values is large.

The phase current measurement is a real number and therefore should be grouped into discrete ranges. Phase current magnitude was separated into normal and high ranges. The normal range was 0–1199 A. The high range was all values greater than or equal to 1200 A. The relay status, Snort alert, and control panel remote trip status features are all binary. Possible relay status values are tripped and not tripped. Possible Snort alert status values are alert and no alert. Possible control panel remote trip status values are tripped and not tripped.

The MED is a merged compressed data set with quantized features. Data from sensors with lower sample rates is up sampled to match the sampling rate of the sensor with the highest sampling rate. The up sampling process depends upon the sensor type. Continuously sampled sensors update their value at each sample period-based upon the current measured state. The current magnitude and relay status are continuously sampled. Event-based sensors provide a single message when a state change occurs. The Snort alert and control panel remote trip status features are event-based. For each, when the sensor detects the presence of an event the sensor provides a message indicating the event occurred. In a data log, a continuously sampled sensor measurement takes a value and holds that value across multiple samples until the state changes.

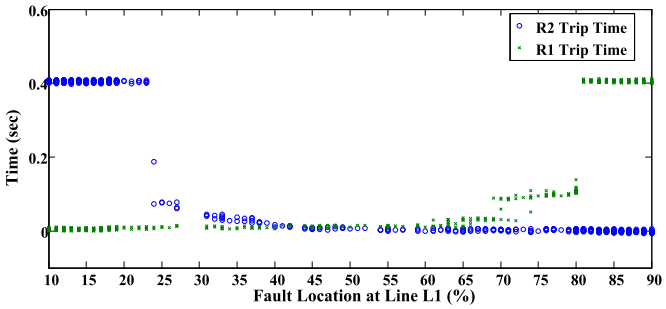

Fig. 3. Relay trip time versus fault location for relays R1 and R2.

Conversely, in the data log, event-based features are asserted for a single sample for each measured event.

When up sampling, continuously sampled sensor measurements are mapped to the nearest sample period after the measurement. All samples without a value take the value of nearest preceding sample. Event-based sensor measurements are also mapped to the nearest sample period after the measurement. All samples without a value take the nonasserted value. For this paper, the current magnitude measurements were measured at 120 samples/s which is the highest sampling rate of all features. Relay status, Snort alerts, and control panel log features were up sampled according to the aforementioned procedure.

An MED represents one instance of a scenario. The TSs of rows in the MED are normalized by subtracting the time of the first row from all other rows. This causes all MEDs to start from time 0.

B. Creating and Grouping Paths

A path is a list of observed system states arranged in temporal order. Paths are extracted by down-sampling the MED while preserving all state transitions. A state change is a change on any sensor value between two MED samples. The MED is parsed to identify all periods of consistent state. Consistent state periods are down-sampled using a user defined sample period. For this paper, the sample period was 0.5 s. Each unique state is assigned a state identifier (S_{id}) and all known states are stored in a state data base.

A path is extracted for each MED. A single scenario will have many unique paths due to the dynamic nature of power systems, variations in the order of states within a path, and due to variations in event timing. Using the raw paths derived from the extraction process for classification results in poor classification accuracy. The common path mining algorithm is used to shrink the larger group of paths into a representative set of common paths which represent normal variation and serve as a set of signatures for each scenario.

Grouping is an optional step which preprocesses input data to separate large classes into smaller sub-classes. Grouping can lead to more accurate classification when the sub-classes are sufficiently different from one another.

Fig. 3 clearly shows zones 1 and 2 trip boundaries for both relays. Additionally, Fig. 3 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 relay trip time for Fig. 3 was calculated from the MED as the time relay status is transitions from closed to open minus the initial time the line current equals is high. System behavior also varies as the system load changes.

Ideally, instances of SLG faults from a two zone distance protection scheme can be separated into three groups according to the area of the line in which the fault occurs. Group 1 includes faults from the length of the line which is protected by relay R1's zone 1 and relay R2's zone 2. From Fig. 2, group 1 includes faults which occur between 10% and 23% of the line. For group 1 faults, relay R1 should trip instantly and R2 should trip after 0.4 s. Group 2 includes faults protected by relay R1's and R2's zone 1. Both relays should trip instantly for group 2 faults. From Fig. 3, group 2 faults occur between 24% and 79% of the line. Group 3 includes faults protected by relay R1's zone 2 and relay R2's zone 1. Relay R1 should trip after 20 cycles and R2 should trip instantly for group 3 faults. From Fig. 3, group 3 faults occur between 80% and 90% of the line.

Observed trip times in group 2 tend to increase as the fault approached the zones 1 and 2 boundary points. To compensate for this observed behavior the SLG 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%, and 81%–90%. Additionally, it was observed that trip times partially correlated to the system load. As a result, the SLG fault paths were grouped by fault location and load. Four load ranges were used: 200–249, 250–399, 300–349, and 350–399 MW. This grouping subdivided the SLG fault paths into $9 * 4 = 36$ sub-groups.

C. Common Path Mining

For this experiment the set **G** consists of 5000 raw paths from 5000 instances of the 25 scenarios. The common path mining algorithm produced 477 common paths across all scenarios. The minimum and maximum number of common paths for a single scenario were 4 and 53, respectively. The 15 SLG fault scenarios had 421 common paths spread among them. The remaining ten scenarios had 56 common paths. The large number of common paths for the SLG faults is due to the large variation in relay trip times as fault location and system load varies.

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 TSs. An edge between two vertices represents the temporal transition between two states. Each vertex is marked with state information. Note that, only necessary features are displayed to save space. Fig. 4 shows common paths for two scenarios, a fault in the 36%–40% fault location of line L1 and a fault replay attack on line L1. The fault and fault replay paths both start at the system normal state. For real faults, the PMU will measure high current when a fault is present while for the fault replay attack, the attacker injects high current measurements to the PDC. This makes the second state of both common paths high current detected at relay R1, i.e., I_{R1} = high. However, these paths differ

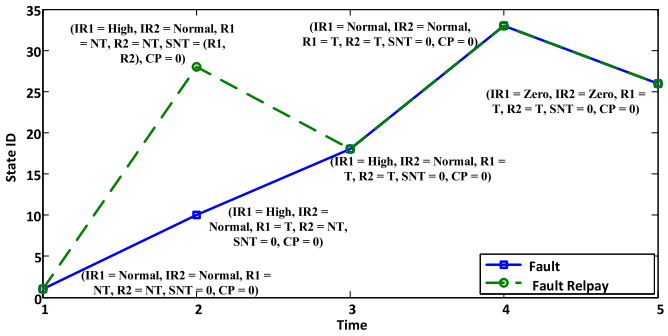

Fig. 4. 2-D coordinates documenting fault versus fault replay attack common paths.

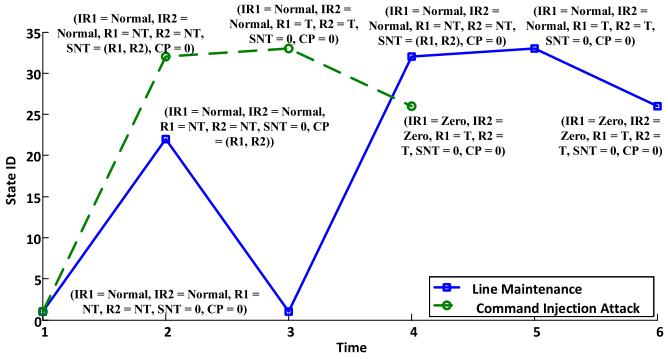

Fig. 5. 2-D coordinates documenting line maintenance versus command injection attack common paths.

immediately because for the fault replay, the attacker has to inject relay trip commands to relay R1 and R2 at the same time. As such, the second state for the fault replay attack has the trip commands to R1 and R2 detected by Snort, i.e., SNT = (R1, R2) in Fig. 4.

Fig. 5 shows common paths for line maintenance and command injection attack scenarios. The primary difference between the two scenarios is the command to open relays R1 and R2 originates from the control panel computer for the line maintenance scenario. This causes the control panel log to include a trip command message. The common path for the line maintenance scenario includes a state noting the detection of control panel log events [i.e., CP = (R1, R2)] and states showing Snort detecting remote trip command network packets [i.e., SNT = (R1, R2)]. The common path for command injection includes the Snort alert but excludes the control panel log state.

Figs. 4 and 5 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 create unique paths for each scenario type from data sets which measure behavior associated with the scenarios.

VI. Evaluation

Three approaches were used to evaluate the IDS. First, the IDS was used to classify 5000 instances of scenarios from the test data set described in Section IV of this paper. Confusion matrices are provided to show IDS accuracy. A detailed review

TABLE III
CONFUSION MATRIX FOR SCENARIOS Q1–Q13

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
Q1	505	0	31	0	0	0	0	0	0	0	0	0	0
Q2	0	502	0	34	0	0	0	0	0	0	0	0	0
Q3	10	0	301	0	0	0	0	0	0	0	0	0	0
Q4	0	6	0	321	0	1	0	0	0	0	0	31	0
Q5	0	0	0	0	130	0	0	0	0	0	0	0	0
Q6	0	0	0	0	0	108	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	67	0	0	0	0	0	0
Q8	0	0	0	0	0	0	0	54	0	0	0	0	0
Q9	0	0	0	0	0	0	0	0	99	0	0	0	0
Q10	0	0	0	0	0	0	0	0	0	57	0	0	0
Q11	0	0	6	0	1	0	0	0	0	0	127	0	0
Q12	0	0	0	0	0	0	0	0	0	0	0	104	0
Q13	0	0	0	0	0	0	0	0	0	0	0	0	179
Oth.	1	2	3	1	0	0	0	0	0	0	0	0	0
Unk	3	1	4	0	0	1	35	32	0	26	0	0	0
Unc	0	2	8	4	0	0	0	0	0	0	0	0	0

TABLE IV
CONFUSION MATRIX FOR SCENARIOS Q14–Q25

	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25
Q14	220	0	0	0	0	0	0	0	0	0	0	0
Q15	0	208	0	0	0	0	0	0	0	0	0	0
Q16	0	0	162	0	0	0	0	0	0	0	0	0
Q17	0	0	0	40	0	0	0	0	0	0	0	0
Q18	0	0	0	0	73	0	0	0	0	0	0	0
Q19	0	0	0	0	0	33	0	0	0	0	0	0
Q20	0	0	0	0	0	0	45	0	0	0	0	0
Q21	0	0	0	0	0	0	0	424	0	0	0	0
Q22	0	0	1	0	0	0	0	0	413	0	0	0
Q23	0	0	0	0	0	0	0	0	0	122	0	0
Q24	0	0	0	0	0	0	0	0	0	0	112	0
Q25	0	0	0	0	0	0	0	0	0	0	0	114
Oth	0	0	0	2	1	5	0	0	0	0	0	0
Unk	10	12	2	23	0	12	0	6	33	16	19	0
Unc	37	58	0	0	0	0	0	0	0	0	0	0

of the algorithms ability to classify SLG faults by fault location is also provided. Second, training and testing was repeated with sets of four scenarios missing from the data set. This test was used to demonstrate the IDSs ability to detect zero-day attacks and unknown scenarios. Finally, IDS cost and performance was measured by measuring the amount of processing time and memory required during training and evaluation.

Tables III and IV provide confusion matrices for the 25 tested scenarios. The confusion matrices were separated into two tables to allow them to fit in the column width of this paper. The row labeled "Oth" represents scenarios Q14–Q25 in Table III and Q1–Q13 in Table IV. The row labeled "Unk" provides the number of instances which were unclassified due to no matching common path. Finally, the row labeled "Unc" provides the number of instances with uncertain classification due to matching more than one common path from more than one scenario.

In total, 90.4% of the tested instances were correctly classified and 2.7% of the instances were misclassified. 4.7% of instances were classified as unknown and 2.2% were classified as uncertain. All of the cases of uncertain classification were related to SLG fault instances which matched a common path for more than one fault scenario.

The IDS can generate false positives, especially, in the case of scenarios which are designed to mimic a nonattack scenario or event. For this paper, false positives rates were calculated for all nonattack scenarios misclassified as attacks. Scenarios Q1 and Q2, both SLG faults, had 2.1% and 1.6% false positive rates, respectively. In both cases, the majority of false positives were classified as fault replay attacks. Replay attacks are designed to mimic SLG attacks. One out of eleven false positives was classified as a relay disable attack. Scenarios Q5 and Q6, both line maintenance events, had 0.8% and 0.9% false positive rates, respectively, which was one false positive for Q5 and Q6, respectively. For the Q5 scenario, the false positive was a command injection attack to open both relays at the end of the transmission line. For the Q6 scenario, the false positive was a fault replay attack. In both cases, the sequence of states in the common paths for the actual scenario and the misclassified scenario have overlapping sub-sequences of states. This overlap combined with variability in observed data due to power system and measurement system dynamics can lead to false positives.

Additional evaluation was performed for classifications of the sub-groups of scenario Q1, a SLG fault on line L1. The paths for Q1 were grouped into sub-groups by fault location and circuit load as previously mentioned. The SLG fault with grouping accuracy rate was 84.6% while 11.35% of the paths were misclassified. Further analysis showed that a majority of misclassification occurred when SLG fault groups were classified as members of a neighboring or nearby fault group. The grouping experiment demonstrates the common path mining algorithm's strength of finding unique paths for even similar scenarios.

Tenfold cross-validation was used to evaluate the detection accuracy of zero-day attack scenarios as shown in Table V. For each round of testing four scenarios were randomly selected to be excluded from training but present in the testing data set. The average detection accuracy for zero-day attack scenarios was 73.43%. However, there were cases where the detection rate for zero-day attack was low. For example, analysis of round three results showed that scenario Q6 (command injection to trip relays R1 and R2) was always misclassified as scenario Q3 (fault replay attack on line L1). This occurs because the expected common paths for Q6 and Q3 are similar. Therefore, when Q6 is unavailable in training, instances of Q6 are classified as instances of Q3 which leads to misclassification. In this case, both Q6 and Q3 are attacks and the zero-day attack is classified as another attack which is better than classifying as a nonattack. To improve the classification accuracy between similar scenarios additional sensors are needed to illuminate events which are different between the two scenarios. Of course, in the zero-day case it is difficult to predict which additional sensors may be required.

Training and classification processing time and memory usage were measured using an Ubuntu Linux Virtual Machine with 3.5 GHZ CPU and 2 GB memory. Training required 0.33 s per scenario instance and 34 MB memory. Classification of test cases required 0.85 s per scenario instance to complete and 26.2 MB of memory.

Multiple batch processing-based data mining algorithms were used to classify power system faults and cyber-attacks in [22] using the same data used for the work presented

TABLE V
DETECTION ACCURACY FOR FOUR RANDOM ZERO-DAY ATTACKS 10× VALIDATION

Round	Excluded Scenarios	Z.D. Acc. (%)
1	Q3, Q11, Q18, Q22	76.3
2	Q2, Q8, Q12, Q23	67.3
3	Q6, Q11, Q16, Q17	50.5
4	Q1, Q5, Q8, Q10	73.3
5	Q1, Q9, Q19, Q21	91.8
6	Q5, Q13, Q20, Q23	64.7
7	Q5, Q10, Q15, Q16	63.8
8	Q12, Q13, Q19, Q24	70.7
9	Q2, Q7, Q9, Q17	76.3
10	Q9, Q10, Q16, Q19	99.8

in this paper. The results in [22] were for classification with binary classes (attack and nonattack), three classes (attacks, nonattacks, and normal), and multiclass (all classes maintained). The common paths mining-based IDS outperformed all traditional methods in [22] for overall accuracy in the multiclass case. A combination of the JRipper_and Adaboost algorithms produced accuracy approaching 90% which is similar to the accuracy of the common paths mining-based IDS. All other test approaches had significantly lower accuracy than the IDS presented in this paper. The binary and three-class methods in [22] lead to improved accuracy at the expense of classification precision. The common paths mining-based IDS provides accurate and precise classification of each scenario type. Precise classification by scenario type is needed to speed understanding of attacks and to enable automated or manual response. Binary and three-class IDS need post processing to provide additional detail before response. The primary advantage of common paths mining-based IDS over a traditional batch processing IDS is the ability to process data as a stream rather than collecting batches of data for off line analysis. Stream processing minimizes the amount of memory required to train and classify and therefore is better suited for IDS at the scale of a power system.

VII. CONCLUSION

The common paths mining-based IDS provides stateful monitoring of an electric transmission distance protection system by leveraging a fusion of synchrophasor data and information from relay, network security logs, and EMS logs.

The IDS is trained using a common path mining algorithm. Common paths are hybrid signatures and specifications which described patterns of system behavior associated with power system events. The algorithm provides a time-domain data analysis approach to overcome transients present in the measurements. This is done by mining shared states out of a group of observed paths. Common paths are used to describe system responses to power system disturbances, control actions, and cyber-attacks.

The IDS matches monitored system state traversal to common paths to make classification decisions. Classification is specific to each trained scenario rather than simply an indication of normal or abnormal activity.

In this paper, the IDS was trained an evaluated for a three-bus two-line transmission system which implements a two zone distance protection scheme. Twenty five scenarios consisting of stocktickerSLG faults, control actions, and cyber-attacks were implemented on a hardware-in-the-loop test bed. Scenarios were run in a loop 10 000 times with randomized system parameters to create a dataset for IDS training and evaluation. The IDS correctly classified 90.4% of tested scenario instances. Evaluation also included a tenfold cross-validation to evaluate the detection accuracy of zero-day attack scenarios. The average detection accuracy for zero-day attack scenarios was 73.43%. The common paths mining-based IDS outperforms traditional machine learning algorithms and is better suited for the high volume of data present in power systems.

Currently, the common paths mining-based IDS builds common paths from captured data logs. Capturing such data logs for real systems is difficult. As such, future work is required to limit the amount the number of captured scenarios instances required to train the algorithm. The IDS was tested by offline review of test data sets. Future work is needed to update the IDS to perform real time classification from live system inputs and to incorporate the classifier with an intelligent adaptive control framework [23] to achieve increased automation in of power systems.

REFERENCES

[1] *A Systems View of the Modern Grid*, Nat. Energy Technol. Lab. (NETL), Morgantown, WV, USA, 2007. [Online]. Available: https://www.smartgrid.gov/sites/default/files/pdfs/a_systems_view_of_the_modern_grid.pdf

[2] *NERC Standards Critical Infrastructure Protection CIP-002-3 Through CIP-009-3*, North American Elect. Rel. Corp., Atlanta, GA, USA, 2010. [Online]. Available: http://www.nerc.com/page.php?cid=2|20

[3] *NISTIR 7628 Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cybersecurity Strategy, Architecture and High-Level Requirements*, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2014. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf

[4] D. Powner and D. Trimble, "Electricity grid modernization: Progress being made on cyber-security guidelines, but key challenges remain to be addressed," Gov. Acc. Office, Washington, DC, USA, Tech. Rep. GAO-11-117, 2011. [Online]. Available: http://www.gao.gov/new.items/d11117.pdf

[5] 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

[6] C. W. Ten, J. Hong, and C. C. Liu, "Anomaly detection for cybersecurity of the substations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 865–873, Dec. 2011.

[7] Y. Chen and B. Lou, "S2A: Secure smart household appliances," in *Proc. 2nd ACM Conf. Data Appl. Sec. Privacy*, San Antonio, TX, USA, 2012, pp. 217–228.

[8] 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.

[9] Y. Yang *et al.*, "Intrusion detection system for network security in synchrophasor systems," in *Proc. IET Int. Conf. Inf. Commun. Technol.*, Beijing, China, 2013, pp. 246–252.

[10] Y. Zhang, L. Wang, W. Sun, R. C. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.

[11] H. Hadeli, R. Schierholz, M. Braendle, and C. Tuduce, "Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Mallorca, Spain, 2009, pp. 1–8.

[12] R. Berthier and W. H. Sanders, "Specification-based intrusion detection for advanced metering infrastructures," in *Proc. IEEE 17th Pac. Rim Int. Symp. Depend. Comput.*, Pasadena, CA, USA, 2011, pp. 184–193.
[13] J. Valenzuela, J. Wang, and N. Bissinger, "Real-time intrusion detection in power system operations," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1052–1062, May 2013.
[14] M. Talebi, J. Wang, and Z. Qu, "Secure power systems against malicious cyber-physical data attacks: Protection and identification," *World Acad. Sci. Eng. Technol.,* vol. 6, no. 6, pp. 112–119, 2012. [Online]. Available: http://[2]waset.org/publications/6605/secure-power-systems-against-malicious-cyber-physical-data-attacks-protection-and-identification
[15] W. Lee, S. Stolfo, and K. Mok, "A data mining framework for building intrusion detection models," in *Proc. IEEE Symp. Sec. Privacy*, Oakland, CA, USA, 1999, pp. 120–132.
[16] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, Taipei, Taiwan, 1995, pp. 3–14.
[17] J. L. Lin, X. S. Wang, and S. Jajodia, "Abstraction-based misuse detection: High-level specifications and adaptable strategies," in *Proc. 11th IEEE Comput. Sec. Found. Workshop*, Rockport, MA, USA, 1998, pp. 190–201.
[18] 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.*, Big Island, HI, USA, 2002, pp. 1995–2002.
[19] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2012.
[20] H. Saadat, *Power System Analysis*. New York, NY, USA: McGraw-Hill, 2010.
[21] R. Nylén, "Auto-reclosing," *ASEA J.*, vol. 52, no. 6, pp. 127–132, 1979.
[22] R. Borges *et al.*, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. Int. Symp. Resil. Control Syst.*, Denver, CO, USA, 2014, pp. 1–8.
[23] 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, Dec. 2014.
[24] G. Coates, K. Hopkinson, S. Graham, and S. Kurkowski, "Collaborative, trust-based security mechanisms for a regional utility intranet," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 831–844, Aug. 2008.

Shengyi Pan (S'12–M'15) received the B.Eng. degree in electronic information engineering from Fuzhou University, 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, 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 current research interests include smart grid technologies, cyber security, data mining, and big 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, in 2001 and 2008, respectively.

He joined Mississippi State University, Starkville, MS, USA, in 2008, where he currently serves as an Associate Professor of Electrical and Computer Engineering, an Associate Director of the Distributed Analytics and Security Institute, and the Director of the Critical Infrastructure Protection Center. His current 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, Kirtipur, Nepal, in 2005. He is currently pursuing the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA.

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