IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 23, 2021, accepted August 5, 2021, date of publication August 24, 2021, date of current version September 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106873

Multi-Source Multi-Domain Data Fusion for Cyberattack Detection in Power Systems

ABHIJEET SAHU[1], **ZEYU MAO**[1], **(Graduate Student Member, IEEE), PATRICK WLAZLO**[2], **HAO HUANG**[1], **(Member, IEEE), KATHERINE DAVIS**[1], **(Senior Member, IEEE), ANA GOULART**[2], **AND SAMAN ZONOUZ**[3]

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA
[2]Electronics Systems Engineering Technology Program, Texas A&M University, College Station, TX 77843, USA
[3]Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ 08854, USA

Corresponding author: Abhijeet Sahu (abhijeet_ntpc@tamu.edu)

This work was supported by the U.S. Department of Energy's (DoE) Cybersecurity for Energy Delivery Systems Program under Award DE-OE0000895.

ABSTRACT Modern power systems equipped with advanced communication infrastructure are cyber-physical in nature. The traditional approach of leveraging physical measurements for detecting cyber-induced physical contingencies is insufficient to reflect the accurate cyber-physical states. Moreover, deploying conventional rule-based and anomaly-based intrusion detection systems for cyberattack detection results in higher false positives. Hence, independent usage of detection tools of cyberattacks in cyber and physical sides has a limited capability. In this work, a mechanism to fuse real-time data from cyber and physical domains, to improve situational awareness of the whole system is developed. It is demonstrated how improved situational awareness can help reduce false positives in intrusion detection. This cyber and physical data fusion results in cyber-physical state space explosion which is addressed using different feature transformation and selection techniques. Our fusion engine is further integrated into a cyber-physical power system testbed as an application that collects cyber and power system telemetry from multiple sensors emulating real-world data sources found in a utility. These are synthesized into features for algorithms to detect cyber intrusions. Results are presented using the proposed data fusion application to infer False Data and Command Injection (FDI and FCI)-based Man-in-The-Middle attacks. Post collection, the data fusion application uses time-synchronized merge and extracts features. This is followed by pre-processing such as imputation, categorical encoding, and feature reduction, before training supervised, semi-supervised, and unsupervised learning models to evaluate the performance of the intrusion detection system. A major finding is the improvement of detection accuracy by fusion of features from cyber, security, and physical domains. Additionally, it is observed that the semi-supervised co-training technique performs at par with supervised learning methods with the proposed feature vector. The approach and toolset, as well as the dataset that is generated can be utilized to prevent threats such as false data or command injection attacks from being carried out by identifying cyber intrusions accurately.

INDEX TERMS Multi-sensor data fusion, intrusion detection system, co-training, supervised learning, unsupervised learning, cyber-physical systems, power systems.

I. INTRODUCTION

Multi-sensor data fusion is a widely-known research area adopted in many sectors, including military, medical science, finance, and energy. In certain natural systems, data fusion occurs automatically. For example, human cognition of events seamlessly combines inputs from a human's senses. The brain can make a union, intersection, or *exclusive or* with the data and enact a complex decoding or decrypting techniques. The brain will react the way it is trained to process data since childhood. This ability streamlines decision-making during typical as well as extreme events, e.g., to recognize that a house is on fire and quickly escape. However, this natural fusion process does not occur

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

automatically for cyber-physical systems, yet it serves as a model for what engineered fusion systems strive to achieve.

In the brain example, *intra-domain* sensor fusion refers to the data collection from similar sensors such as vision from left and right eyes. The *inter-domain* sensor fusion refers to the fusion of sight, smell, acoustics, etc. Supervised learning refers to how the mind is trained to perceive such sensor data by guidance from an instructor. Unsupervised learning refers to how without any instructions, the mind trains. In this narrative, if the victim forgets to wear glasses, he loses some labels from the accumulated information.

Automatic driving systems are cyber-physical systems that widely use data fusion to fuse images and videos from similar or disparate sensor types [1]. A power system is also a cyber-physical system, yet most of its fusion applications are currently intra-domain and consider only physical data. Examples include fault detection [2] and intrusion detection using principal component analysis (PCA) [3]. Similarly, for network protection in industrial control systems, intrusion detection systems (IDS) such as Snort, BRO, or Suricatta, are increasingly used [4]. They offer a pure cyber-centric approach that results in high false alarms [5]. Traditional physical measurements are not sufficient to reflect the accurate state of the cyber-physical system, e.g., to classify it as cyber-secure, cyber-insecure, physical-secure, physical-insecure, physical-abnormal. Thus, data fusion can fill these gaps and improve situational awareness of the whole system. Combining the benefits of visibility of both cyber and physical systems, cross-domain data fusion has the potential to methodically and accurately detect mis-operation and measurement tampering in power systems caused by cyber intrusions.

In power system operations, the telemetry used for collecting wide area measurements may have errors due to sensor damage or cyber-induced compromise; if undetected, applications that rely on these data can become unreliable and untrustworthy. Sensor verification based on multi-source multi-domain measurement collection and fusion solves such problems. It is a valuable mechanism for detection and detailed forensics of cyber intrusions targeting physical impact. While offering numerous potential benefits, fusion for attack detection in real-world utility-scale power systems presents challenges that hinder adoption, including the creation, storage, processing, and analysis of the associated large datasets. Fortunately, with the proliferation of affordable computing capability for processing high-dimensional data, it is becoming more feasible to deploy fusion techniques for accurately detecting intrusions. Thus, research is needed to take advantage of these data and computing capabilities and create fusion-based detection techniques that solve this problem.

Cyberattacks often progress in multiple stages, e.g., starting with a reconnaissance phase, executing intrusions and vulnerability exploitations, and culminating in actions targeting the physical system such as manipulating measurements and commands. The events that comprise these incidents and forensics about what occurred are not reflected using only coarse cyber-side features. For example, an intruder may take months in the reconnaissance phase, but during this period, none of the physical side features reflect any abnormality. Similarly, later when an intruder is injecting false commands or tampering measurements, most of the cyber side features do not reflect any abnormality, assuming the adversary is stealthy. Additionally, the system dynamics in both cyber and physical space vary considerably; this causes challenges in merging data. The homogenization of cyber and physical data with preservation of temporal information and appropriate handling of inconsistent data fields is addressed in this work.

Sensor time resolution varies across domains and within domains, which challenges merging the data. The time resolution of physical measurements depends on polling rates and specifications of the field device. For example, phasor measurement units (PMUs) provide GPS synchronized data at subsecond data rates, SCADA systems provide data on the seconds to minutes time frame, and smart meters deployed residentially may have hourly resolution [6]. Relays monitoring system transients have a time resolution on the order of milliseconds. Similarly, the network logs and alerts from IDS such as Snort have a resolution of milliseconds. Data fusion solutions for cyber-physical power systems must effectively handle varying data rates.

The use of machine learning (ML) and deep learning (DL) for intrusion detection faces the problem that the trained model's effectiveness depends on the data collected [7] ; it is a challenge to obtain a realistic baseline and to use realistic data to validate the solution for a real-time cyber-physical system [8]. A natural problem that arises with fusion for ML is feature expansion, selection, and cyber-physical state space explosion, which results in the *curse of dimensionality* [9]. This problem can be handled through feature reduction. However, detection is affected by the choices of data processing techniques applied (e.g., feature reduction, balancing, scaling, encoding) [10]. The impact of such factors on detection accuracy must therefore be quantified before the techniques can be trusted for securing critical infrastructure.

This work hypothesizes that the use of fused data from cyber and physical domains can enable better attack detection performance than either domain separately if the aforementioned challenges are addressed. Hence, a multi-sensor multi-domain platform is presented, that fuses data and detects cyber intrusions. First, interfaces for collecting data sources from cyber and physical side emulators are provided. Then, these interfaces are used to collect real-time data from cyber, physical, and security domains; finally, the datasets are fused before detecting cyber intrusions. Aggregation of real-time sensor data from multiple sources, including Elasticsearch [11], TShark [12], raw packet captures with Distributed Network Protocol 3 (DNP3) traffic, and Snort logs [13] is performed, that is extracted during the emulation of Man-in-The-Middle (MiTM) attacks on a synthetic electric grid, modeled in the Resilient Energy Systems Laboratory (RESLab) testbed [14]. Fig. 1 gives an overview of the

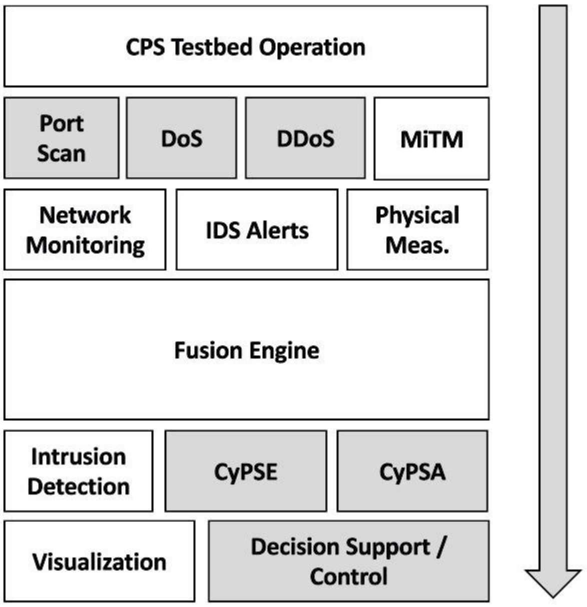


FIGURE 1. Top down approach for designing testbed, incorporating cyberattacks, aggregating real-time sensor alerts, leveraging data fusion engine for intrusion detection, followed by intrusion detection, cyber-physical situation awareness (CyPSA), and state estimation (CyPSE). The white highlighted blocks indicate the components incorporated in this work.

multi-source data fusion presented. The major contributions of the work in this paper are the following:

1) A cyber-physical intrusion detection solution is proposed based on a data-driven hybrid information fusion algorithm that leverages real-time data from cyber and power-based sensors. The solution utilizes the cyber-physical logical interconnections and allows accurate detection of malicious misbehaviors in either cyber controllers or power system components in a timely manner.
2) A machine learning-based approach is proposed to improve the scalability of the detection framework for large-scale platforms. Moreover, the proposed techniques can cope with different levels of data heterogeneity from various low-level cyber-physical security probes.
3) The proposed solution is deployed and validated against cyber-physical attacks on a real-world power grid testbed that included several types of distributed and commonly-used host- and network-based detection probes. Furthermore, a working visualization prototypes is developed for online cyber-physical situational awareness during the attack progress in real-time.
4) An end-to-end data fusion engine from multiple sources is developed and presented for cyberattack detection in a real-time testbed emulation of a synthetic electric grid.
5) Data pre-processing techniques such as balancing, normalization, encoding, imputation, feature reduction, and correlation are evaluated to address feature explosion and tackle data inconsistencies, before training the machine learning models.
6) Improvement in cyberattack detection capability of the trained supervised, unsupervised and semi-supervised models, built from the fused dataset performance, compared to pure cyber or physical feature based IDS models is demonstrated.
7) An orchestration application is designed to visualize each stage of data pipelining, pre-processing, followed by training IDSes for different attack use cases.

The paper proceeds as follows. Section II provides a background on the types of multi-sensor data fusion and their applications in cyber-physical systems and power systems in particular. In Section III, the RESLab architecture, the attack types considered, and the data fusion procedure is discussed. The details on the data sources, the data fusion types, and the dataset transformations used in this work are presented in Sections IV, V, and VI respectively. Finally, intrusion detection based on unsupervised, supervised, and semi-supervised learning methods are presented in Section VII. Experiments are performed for four use cases, and results are analyzed in Section IX. Section X concludes the paper with a discussion of the results.

II. DATA FUSION BACKGROUND

A. CYBER-PHYSICAL THREAT OVERVIEW

The cyber-physical threats motivating this work constitute a diversity of potential mechanisms that can compromise the confidentiality, integrity, and availability of the system, targeting power system impact such as by exploiting a series of vulnerabilities to compromise the normal operation of the system.

As prevalent types of attack, Denial of Service (DoS) cyberattacks exhaust target networks with random traffic to disrupt the normal operation, while Distributed DOS attacks leverage botnets to exhaust links at multiple locations to cause more severe damage [15]. A Telephony DoS attack hit three distribution utilities blocking incoming and outgoing calls from customers [16], which contributed to power loss for a quarter-million people in Ukraine.

Authors in [17] provide a taxonomy of cyberattacks in ICS networks based on timeliness, confidentiality, integrity, availability. The risk of an attack variant will vary under different power system scenarios. For example, latency caused by DoS can delay restorative actions post-compromise. Data and command poisoning can disrupt situational awareness, mislead state estimation, and misoperate devices, where certain actions can have the potential to cause further contingencies (outages) or blackouts.

A quantitative assessment of risk and situational awareness requires cyber-physical state estimation that is both accurate and timely, which must be inferred using data fusion from both cyber and physical sensors. Prior works on cyber-physical situational awareness that leveraged Markov Decision Process (MDP) [18], Attack Graphs [19], and

Bayesian Attack Graphs [20], were based on the exploration of static vulnerabilities in the system to construct attack graph models for ranking critical assets, contingencies, as well as learning graph structure. Similarly, [21], proposes a stochastic Bayesian network model for calculating cyber-physical security index for risk management. An expected load curtailment index against cyber attacks on protection devices and their control logic is presented in [22]. Reference [23] models Stuxnet attack, using a Boolean Logic Driven MDP, that leverages estimated values of the success probabilities and rates of the elementary attack steps. Dynamic updates on these models, which enable them to be useful for applications such as risk quantification and targeted system restoration and response, require aggregation of real-time data from multiple sensors. The data collected in sensor logs will depend on the attack and sensor types. Hence, a complex attack affecting a collection of sensors requires the fusion of data from multiple sensors.

The threat model this paper focuses on, targets the integrity of critical devices. Specifically, it emulates multi-stage attacks on a large-scale synthetic electric grid, where the intruder first gains Secure Shell (SSH) access to a device within the substation LAN, then performs coordinated MiTM attacks targeting different combinations of FCI and FDI attacks sequentially on multiple DNP3 outstations to cause transmission line overloading. To accomplish it, the intruder performs Address Resolution Protocol (ARP) spoofing to impersonate the DNP3 master for the outstation and vice-versa, compromising the *integrity*, then further sniffs the measurement and commands within the two end-points. The details of the attack scenarios are elaborated in the testbed paper [14] and the MiTM attack paper [24].

B. MULTI-SENSOR DATA FUSION

Multi-sensor data fusion aims to make better inferences than those that could be accrued from a single source or sensor. According to *Mathematical Techniques in Multisensor Data Fusion* [25], multi-sensor data fusion is defined as *"a technique concerned with the problem of how to combine data from multiple (and possibly diverse) sensors to make inferences about a physical event, activity, or situation."* A data fusion process is modeled in three ways: a) functional, b) architectural, and c) mathematical [25]. A functional model illustrates the primary functions, relevant databases, and inter-connectivity to perform the fusion. It involves primarily filtering, database creation, and pre-processing such as scaling and encoding. An architectural model specifies hardware and software components, associated data flows, and external interfaces [26]. For example, it models the location of the fusion tool in a testbed. There are three types of fusion architecture: centralized, autonomous, or hybrid [25]. In centralized architectures, either raw or derived data from multiple sensors are fused before they are fed into a classifier or state estimator. In autonomous architectures, the features extracted are fed to the classifiers or estimators for decision making before they are fused. The fusion techniques used in the second case involve Bayesian [27] and Dempster Shafer inference [28] because these fusion algorithms are fed with the probability distributions computed from the classifiers or the estimators. The hybrid type mixes both centralized and autonomous architectures. The mathematical model describes the algorithms and logical processes.

A holistic data fusion method must consist of all three: functional, architectural, and mathematical models. The functional model defines the objective of the fusion. Since the work aims to detect intrusions, determining which data are due to cyber compromise is essential. Functional goals may also include estimating the position of the intruder in the system or estimating the state of an electric grid, where the pre-processing techniques vary based on the goal. The architecture model defines the sequence of operations. The proposed fusion technique follows the centralized architecture. Finally, the mathematical model defines how these features are processed and merged. Section IV details the proposed fusion models.

C. MULTI-SENSOR FUSION APPLICATIONS

Recently, multi-sensor fusion has been adopted in computer vision, automatic vehicle communication, and it is entering power systems. The authors in [29] review multi-sensor data fusion technology, including the benefits and challenges of different methods. The challenges are related to data imperfection, outliers, modality, correlation, dimensionality, operational timing, and inconsistencies. For example, without the usage of specific estimation method such as Kalman filtering, sensors with multiple time resolutions requires under-sampling or over-sampling. The response time of certain sensors also varies depending on the sensor age and type. Data received from multiple sensors must be transformed to a common spatial and temporal reference frame [25]. Imperfection is dealt with using fuzzy set theory, rough set theory, or Dempster Shafer theory.

Multi-sensor data fusion is used in military applications for automated target recognition, battle-field surveillance, and guidance and control of autonomous vehicles [30]. Further, the idea has been expanded to non-defense areas such as medical diagnosis, smart buildings, and automatic vehicular communications [31]. Authors in [32], explore techniques in multi-sensor satellite image fusion to obtain better inferences regarding weather and pollution. Data fusion has also been proposed to accurately detect energy theft from multiple sensors in advanced metering infrastructure in power distribution systems [33].

Data fusion is expanded in [34] from cyber-physical systems (CPS) to cyber-physical-social systems with the use of tensors. Algorithms proposed for mining heterogeneous information networks cannot be directly applied to cross-domain data fusion problems; the fusion of the knowledge extracted from each dataset gives better results [35].

D. DATA FUSION IN POWER SYSTEMS

The data from diverse domains play a major role in power system operation and control. Weather data is vital for forecasting, e.g., solar, wind, and load, to schedule generation. Data in cyberspace include data that provide for automation in power system ICS and play a crucial role in wide-area control and operation in the electric grid. However, to proceed with multi-domain data fusion, the following question must first be answered: To what measurable quantities do *cyber data* and *physical data* refer?

A simple example of *cyber data* in ICS is a spool log of a network printer in the control network. It is crucial to question, could the attack on the centrifuge in the Natanz Uranium Enrichment plant be prevented, if it had a logger to record the events of a machine with a shared printer, to prevent the exploitation of remote code execution on this machine? The answer is *no* because there were many other vulnerabilities such as WinCC DB exploit, network share, and server service vulnerability, in parallel to print server vulnerability that compromised the Web Navigation Server which was connected to the Engineering Station that configured the S7-315 PLCs which over-speeded the centrifuge [36]. Hence, the deployment of cyber telemetry in every computing node in an ICS network is a solution that seems attractive but results in numerous false alarms. Then, the question arises, can alerts be reduced by amalgamating such data with data from physical sensors?

Data fusion proposed in the areas of power systems is mainly intra-domain. Existing works do not consider the fusion of cyber and physical attributes for intrusion detection together. A probabilistic graphic model (PGM) based power systems data fusion is proposed in [37], where the state variables are estimated based on the measurements from heterogeneous sources by belief propagation using factor graphs. These PGM models require the knowledge of the priors of the state variables, and also assume the measurements to be trustworthy. Hence, such solutions cannot detect cyber-induced stealth false data injection attacks. Several works on false data injection detection are based on machine learning [38]–[41] and deep learning [42]–[47] techniques. The authors in [48] address stealthy attacks using multi-dimensional data fusion by collecting information from the power consumption of physical devices, control operation, and system states feed to the cascade detection algorithm to identify stealthy attacks using Long Short Term Memory. Machine learning techniques including clustering are used in power system security for grouping similar operating states (emergency, alert, normal, etc.) to automatically identify the subset of attributes relevant for the prediction of the security class. A decision tree-based transient stability assessment of the Hydro-Quebec system is presented in [49]. Techniques of fusion for fault detection [2] and real-time intrusion detection using PCA [3] are specific to the physical domain. The design of such models requires data fusion and must consider impending system instabilities caused by cyber intrusions.

Cymbiote [50] multi-source sensor fusion platform is similar to this work, that has leveraged fusion from multiple cyber and physical streams and trained with only supervised learning-based IDS. Moreover, their work does not clearly describe the features extracted from different sources.

E. MULTI-DOMAIN FUSION TECHNIQUES

Techniques such as co-training, multiple kernel learning, and subspace learning are used for data fusion problems. Co-training-based algorithms [51] maximize the mutual agreement between two distinct views of the data. This technique is used in fault detection and classification in transmission and distribution systems [52] and network traffic classification [53]. To improve learning accuracy, Multiple kernel learning algorithms [54] are also considered, which utilize kernels that implicitly represent different views and combines them linearly or non-linearly. Subspace learning algorithms [55] aim to obtain a latent subspace shared by multiple views, assuming that the input views are generated from this latent subspace. DISMUTE [56] performs feature selection for multi-view cross-domain learning. Multi-view Discriminant Transfer [57] learns discriminant weight vectors for each view to minimize the domain discrepancy and the view disagreement simultaneously. These techniques can be used for cross-domain data fusion.

Coupled matrix factorization and manifold alignment methods are used for similarity-based data fusion [35]. These methods can be implemented intra-domain with multiple data sources. Manifold alignment is another technique that generates projections between disparate data sources but assumes the generating process shares a common manifold. Since the primary goal in this work is to fuse datasets from inter-domain, such methods may not be effective enough. Still, manifold learning is explored for the purpose of feature reduction to train the supervised learning classifier.

To the best of our knowledge, co-training has not yet been implemented in an intrusion detection system that uses inter-domain fusion. Hence, in this work, co-training is performed in inter-domain fused datasets by splitting the dataset into cyber and physical views.

F. DATA CREATION, STORAGE, AND RETRIEVAL

The storage and retrieval of multi-sensor data play a major role in fusion and learning. A relational database management system is predominantly used in traditional Energy Management System (EMS) applications. For example, B.C. Hydro proposes a data exchange interface in a legacy EMS and populates a relational database with the schematic of the common information model defined in IEC 61970 [58]. With the proliferation of multiple protocols and data from diverse sources, it is not easy to construct the Entity-Relationship model of a relational database management system, since the schema cannot be fixed. Since NoSQL stores unstructured or semi-structured data, usually in the key-value pairs or Java Script Object Notation documents, NoSQL is highly encouraged to make use of databases such as Elasticsearch [11], MongoDB [59], or Cassandra [60], for multi-sensor fusion with heterogeneous sources.

Creating multi-domain datasets to advance the research is a challenging task since it requires the development of a cyber-physical testbed that processes real-time traffic from different simulators, emulators, hardware, and software. Currently, few datasets are publicly available that provide features from diverse domains and sources. Most of the datasets are simulator-specific, which restricts the domain to either purely physical or cyber. The widely-known KDD [61] and CIDDS [62] datasets used in developing ML-based IDS for bad traffic detection and attack classification are centric to features in the cyber domain [63]. Tools such as MATPOWER [64] and pandapower [65] provide datasets for physical-side bad data detection. Datasets that include measurements related to electric transmission systems, including normal, disturbance, control, and cyberattack behaviors are presented in [66]–[69]. The datasets contain PMU measurements, data logs from Snort, and also data from a gas pipeline and water storage tank plant. The features in these datasets lack fine-grained details in the cyber, relay, and control spaces, as all the features are binary in nature. A cyber-physical dataset is presented in [70] for a subsystem consisting of liquid containers for fuel or water, with its automated control and data acquisition infrastructure showing 15 real-world scenarios; while it presents a useful way of framing the data fusion problem and approaches for CPS, it is not power system-specific.

A problem in training ML or DL models for intrusion detection through classification, clustering, and fine-tuning hyperparameters is that its effectiveness depends on the data collected. That is, a practical challenge is to obtain a baseline that needs to come from realistic data. Emulation is preferred to simulation for CPS networks since a simulator demonstrates a network's behavior while an emulator functionally replicates its behavior and produces real data. Using real data is important to validate that ML or DL solutions address the actual challenges faced in the data from a real-time cyber-physical system.

The performance of ML and DL models is impacted by the choice of data processing techniques applied to the inputs such as balancing, scaling, or encoding before training the models. The effect of these preprocessing techniques needs to be quantified on the outputs of such ML models before they can be trusted for use in industry.

III. DATA FUSION ARCHITECTURE

Before discussing the data fusion procedures, it is essential to understand the architecture of the RESLab testbed that produces the data during emulation of the system under study.

A. TESTBED ARCHITECTURE

The RESLab testbed consists of a network emulator, a power system emulator, an OpenDNP3 master and an RTAC based master, an intrusion detection system, and data storage, fusion, and visualization software. A brief overview of each component is given below. A detailed explanation of RESLab including its architecture and use cases is provided in [14].

- *Network Emulator* - Common Open Research Emulator (CORE) is used to emulate the communication network that consists of routers, Linux servers, switches, firewalls, IDSes, and bridges with other components emulated with other virtual machines (VMs) in the vSphere environment.
- *Power Emulator* - Power World Dynamic Studio (PWDS) is a real-time simulation engine for operating the simulated power system case in real-time as a DS server [71]. It simulates the substations in the Texas 2000 case as DNP3 outstations [72].
- *DNP3 Master* - DNP3 Masters are incorporated using an open DNP3 based application (both GUI and console-based) and an SEL-3530 Real-Time Automation Controller (RTAC) that polls measurements and operates outstations, sending its traffic through CORE to the emulated outstations in PowerWorld DS.
- *Intrusion Detection System* - Snort is used in the testbed as the rule-based, open-source IDS. It is configured to generate alerts for DoS, MiTM, and ARP cache poisoning-based attacks. Currently, Snort is running as a network IDS in the router in the substation network.
- *Storage and Visualization* - The Elasticsearch, Logstash, and Kibana (ELK) stack is used to probe and store all virtual and physical network interface traffic. In addition to storing all Snort alerts generated during each use case, this data can be queried using Lucene queries to perform in-depth visualization and cyber data correlation.
- *Data Fusion* - A different VM is dedicated to operating the fusion engine that collects network logs and Snort alerts from ELK stack using an Elasticsearch client and raw packet captures from CORE using pyshark. This engine constructs cyber and physical features and merges them using the timestamps from different sources to ensure correct information alignment. Further, it pre-processes them using imputation, scaling, and encoding before training them for intrusion detection using supervised, unsupervised, and semi-supervised learning techniques. This VM is equipped with resources to utilize ML and DL based library such as Scikit, Tensorflow, and Keras to train the engine for classification, clustering, and inference problems.

There are three broad kinds of IDS for Industrial Control Systems: protocol analysis based IDS, traffic mining based IDS, and control process based IDS [73]. The fusion engine in RESLab combines all these types. It performs protocol-specific feature extraction from data link, network, transport layers along with DNP3 layer, control and measurement specific information through DNP3 payload and headers, traffic mining by extracting network logs from multiple sources.

B. ATTACK EXPERIMENTS

Now that the testbed architecture is discussed, the utilization of the testbed to demonstrate a few cyberattacks targeting the grid operation is presented. The threat model considered here is based on emulating multi-stage attacks in a large-scale power system communication network. In the initial stage, the adversary gains access to the substation Local Area Network (LAN) through SSH access, further performing DoS and ARP cache poisoning based MiTM attack to cause FDI and FCI.

Usually, in the Man-in-the-Middle attacks, the adversary secretly observes the communication between sender and receiver and sometimes manipulates the traffic between ends. There are different ways to perform MiTM, such as IP spoofing, ARP spoofing, DNS spoofing, HTTPS spoofing, SSL hijacking, stealing browser cookies, etc. In this current work, MiTM using ARP spoofing is focussed. ARP spoofing or poisoning is an attack-type, in which an adversary sends false ARP messages over a LAN. This results in the linking of an adversary's MAC address with the IP address of a legitimate machine on the network (here, the DNP3 outstation VM). This attack enables the adversary to receive packets from the master, as an impersonator for the outstation and modify commands and forward them to the outstation. In this way, the adversary can cause contingencies such as misoperation of the breakers. The attack is not only to modify but also to sniff the current state of the system since it can receive the outstation response to the master.

The MiTM attacks are performed considering the four use cases targeting a different part of the Texas synthetic grid following different strategies presented in detail in [14]. The use cases are combinations of FDI and FCI attacks performed with different polling rates from the DNP3 Master and the number of master applications considered. In previous work, a Snort IDS-based detection [24] method is demonstrated, which resulted in many false positives. In this work, we employ fusion techniques, and machine learning techniques, to enhance the accuracy of detection by evaluating them using F1-scores, Recall, and Precision values.

C. DATA FUSION PROCEDURE

The steps followed in the data fusion engine, from extracting the features from different sources, with their merge of pyshark, snort, packetbeat, raw packet capture to form cyber table, and the final fusion of cyber and physical table, with the steps of imputation, encoding and visualization is presented in Alg. 1. The details of the sensor sources and the data processing are discussed in details in the next sections.

D. FUSION CHALLENGES

The most challenging task in data fusion is to perform merge operations, because of the different time stamps generated at different sensors. An event will trigger the time-stamped measurements at the sensors. Hence, each sensor's location impacts the time at which the event is recorded. Domain knowledge has been used to write the algorithm to merge different sources meticulously. For example, Elasticsearch's Packetbeat index stores each record reflecting the traffic between a given small time interval. Each record has an event start and end time. While merging Elasticsearch features, such as flow count attribute, a comparison of the raw packet timestamp and event start and end time of Elasticsearch is required, to calculate the flow counts. Moreover, the number of records on the power system side will be less than the cyber side, as events on the power system side are triggered based on the polling frequency as well as on the time at which an operator performs a control operation. Hence missing data for the records are filled using data imputation.

Algorithm 1 Data Fusion Procedure

```
1. Load JSON from raw pcaps.
2. Extract cyber features: network, transport, datalink layer information and store as raw cyber data.
3. Extract features using pyshark.
4. Merge pyshark to the raw cyber data.
5. Extract snort alert.
6. Merge snort to the raw cyber data.
7. Extract features from packetbeat index in elasticsearch.
8. Merge packetbeat features to raw cyber data.
9. Extract DNP3 features (DNP3 points and headers) from raw packet capture.
10. Fuse cyber data with physical data.
11. Imputate missing values.
12. Encode categorical features.
13. Visualize the merged table.
```

IV. MULTI SENSOR DATA

A sensor's data is the output or readings of a device that detects and responds to changes in the physical environment. Every sensor has a unique purpose that helps create crucial features that can assist in intrusion detection. In RESLab [14], the cyber sensors are deployed as Wireshark instances at different network locations for raw packet capture. Additionally, monitoring tool such as Packetbeat is integrated for extracting network flow-based information. For security sensors, Snort IDS logs and alerts are considered. Since the physical system is emulated with PWDS acting as a collection of DNP3 outstations, the real-time readings provided by physical sensors are extracted from the observed measurements at the DNP3 master, from the application layer of the raw packet captured at the DNP3 master. The extractions of these multiple sensors are explained in detail:

A. RAW PCAPS FROM JSON

The packet captures from Wireshark are packet dissected and saved in the JSON format, which is loaded using the panda data frame. Further, from the JSON, around 12 features from the physical, datalink, network, and transport layer of the OSI stack are extracted, as shown in Table 1. The features

TABLE 1. Description of the features used in data fusion.

Features	Description	Def
Frame Len	Length of the frame after network, transport and application header and payload are added and fragmented based on the channel type. For ethernet, the frame length can be max. 1518 bytes, which varies for wireless channels.	0
Frame Prot.	Determines the list of protocols in the layers above link layer encapsulated in the frame.	Nan
Eth Src	Unique source MAC address. Crucial for detection in ARP spoof attacks.	0
Eth Dst	Unique destination MAC address. Crucial for detection in ARP spoof attacks.	0
IP Src	Unique source IP address.	0
IP Dst	Unique destination IP address.	0
IP Len	Stores the length of the header and payload in a IP-based packet. This correlates well with the DNP3 payload size.	0
IP Flags	Indicator of fragmentation caused due to link or router congestion in the intermediary nodes.	0x00
Src Port	Indicates the port number used by the source application using TCP in transport layer. Ex: if the source is the DNP3 outstation, the default port is 20000.	0
Dest Port	Indicates the port number used by the destination application using TCP in transport layer.	0
TCP Len	Stores the length of the header and payload in a TCP-based segment. This correlates well with the DNP3 payload size.	0
TCP Flags	Flags are used to indicate a particular state of connection such as SYN, ACK, etc.	0x00
Retrans.	Indicates if the current record is from a retransmitted packet, caused due to attack or network congestion.	0
RTT	Indicator of propagation and processing delay. High RTT can be caused due to MiTM attack.	-1
Flow Cnt	Indicates the number of TCP flows in a specific time interval. Indicates the connected and disconnected DNP3 masters. Flow is the collection of packets.	-1
Flow Fin Cnt	Indicates if the current flow carries the final packet.	-1
Packets	Number of packets transmitted in a specific time interval.	-1
Snort Alert	Boolean indicating an alert from snort.	0
Alert Type	Indicates the alert type such as DNP3, ARP spoof, ICMP flood or any other types.	Nan
LL Src	Source id of the DNP3 master or outstation. Indicator of which outstation communicates with the master in that specific record.	-1
LL Dest	Destination id of the DNP3 master or outstation. Indicator of which outstation communicates with the master in that specific record.	-1
LL Len	Indicator of the DNP3 payload size as well as the function type. Usually the response carries DNP3 point information, hence this length correlates with the function code as well as the outstation currently communicating.	0
LL Ctrl	This indicates the initiator of the communication. Determines the primary/secondary server.	0x00
TL Ctrl	Indicates the FIN/FIR/Sequence number for determining if the DNP3 payload is the first or final segment.	0x00
Func. code	Indicates the function code: either READ, WRITE, OPERATE, DIRECT OPERATE, etc.	-1
AL Ctrl	Indicates the FIN/FIR/Seq/Confirm and Unsolicited flags. This indicates if there are unsolicited, first, final from application layer standpoint.	0x00
Obj count	This count determines the number of BI, BO, AI, AO points associated with a substation.	0
AL Payload	Contains the DNP3 points used to extract the physical features such as branch status, real power flows and injections in branch and buses for a substation.	Nan

primarily consist of the source and destination IP and MAC addresses, along with the port numbers, flags, and lengths in these layers.

B. ELASTICSEARCH

Real-time traffic collection is performed from network interfaces in CORE, using the Packetbeat plugin in the ELK stack. The Packetbeat plugin helps us extract the flow-based information such as *Flow Count*, *Flow Count Final*, *Packets* shown in Table 1. Elasticsearch queries are based on Lucene, the search library from Apache. Kibana is used to visualize the graphs and real-time data visualization for the Packetbeat index. An example query is shown below:

```
"query": {
"bool": {
"must": [
    { "range": {
    "event.end": {
    "gte": "2020-01-22T00:00:00.000Z",
    "lte": "2020-01-26T00:00:00.000Z"}}},
    {"range": {
        "event.duration": {
        "gte": 0,
        "lte": 3000000}}},
      {"bool":
      {"should":
      [{"match": {
        "destination.port": "20000"}},
      {"match": {
        "source.port": "20000"}}]}},
    {"match":
      {"flow.final": "true"}
    }
]}}
```

The above query returns the records with event start time $2020-01-22T00:00:00.000Z$ and end time $2020-01-26T00:00:00.000Z$, and the event duration is within $0-300000$ ms, and the source or destination port is 20000 (port number associated with DNP3), and the flow is a *final* flow. The keyword *must* designate an *AND* operation, *should* is an *OR* operation, and *match* is an *equals to* operation. A logstash index is also created in Elasticsearch to store the logs of Snort alerts, which is also extracted along with the packetbeat index.

There are two operations on the response from Elasticsearch: a) *Extraction* of essential features b) *Merge* of features to the existing cyber features data frame *cb_table* from raw packet captures. Each record in the packetbeat index is stored in the form of an event with start and end times. In the extraction phase, the *source.packets*, *flow.id*, *flow.final*, *event.end*, *event.start*, *flow.duration* features are extracted and stored in a new data frame *pb_table*. The merge operation of *pb_table* into the existing cyber features is non-trivial due to different timestamps in existing features and features from packetbeat. The features in Table 1 *flow.count*, *flow.final_count*, and *packets*, using the features of *event.end*(*end*), *event.start*(*start*) in *pb_table* and *Time* in the *cb_table* based on the logical OR of three conditions:

1) *Condition* 1 : add counters if the event start is within the range of current and next records in the *cyber_table*

$$cb\_table[i][t] \leq start \wedge cb\_table[i+1][t] \geq start \qquad (1)$$

2) *Condition* 2 : if the event end is within the range of current and next records in the *cyber_table*.

$$cb\_table[i][t] \leq end \wedge cb\_table[i+1][t] \geq end \qquad (2)$$

3) *Condition* 3 : if the event start is less than the current record and event end is greater than the next record in the *cyber_table*.

$$cb\_table[i][t] \geq start \wedge cb\_table[i+1][t] \leq end \qquad (3)$$

The $\vee$ and $\wedge$ are the logical *or* and *and* operators respectively. In this manner, the three features from *pb_table* to the *cb_table* are merged.

C. PYSHARK

Pyshark is a Python wrapper for tshark, allowing python packet parsing using Wireshark dissectors. Using *Pyshark* features such as *Retransmissions* and *RoundTripTime*(*RTT*) is obtained. The RTT is the time duration for a signal or message to be sent plus the time it takes to acknowledge that signal to be received. It has been observed that if congestion is created in any location in between the source and destination such as router or switch, the RTT increases. It also increases due to DoS attacks on the servers or any intermediary nodes in the path between source and destination. The *TCP* based packet follows different retransmission policies based on the TCP congestion control flavor. Hence, the number of *retransmission* packets observed within a given time frame is an indicator of loss of communication or increased delay. Usually, a sender retransmits a request if it did not receive an acknowledgment after some multiples of an *RTT*, whose multiplicity is dependent on the TCP flavor. The *retransmission* and *RTT* features are selected, as features are correlated and directly related to attacks targeting availability and integrity.

D. SNORT

The router inside the CORE emulator runs the Snort daemon based on the specific rules, pre-processors, and decoders enabled in the configuration file to create logs. Snort operates in three modes: packet sniffer, packet logger, and IDS modes. In this work Snort is primarily operated in the IDS mode. The alerts generated at the router in the substation network are continuously probed during the simulation. The alerts are recorded in the form of the *unified*2 format as well as pushed to the Logstash index created in Elasticsearch. Unified2 works in three modes, packet logging, alert logging, and true unified logging. Snort runs in alert logging mode to capture the alerts, timestamped with alert time. Further, the *idstools* python package is utilized to extract these *unified*2 formatted logs. The Snort configuration determines which rules and preprocessor are enabled. The features extracted are the *alert*,*alert_type*, and *timestamp*. The merge into the *cb_table* is performed based on the *timestamp* of each Snort record. The record is inserted based on the condition:

$$cb\_table[i][t] \geq timestamp \leq cb\_table[i+1][t] \qquad (4)$$

E. PHYSICAL FEATURES FROM DNP3

The Distributed Network Protocol version 3 is widely used in SCADA systems for monitoring and control. This protocol has been upgraded to use TCP/IP in its transport and network layer. It is based on the master/outstation architecture, where field devices are at outstations and the monitoring and control are done by the master. DNP3 has its own three layers: a) Data Link Layer, to ensure the reliability of physical link by detecting and correcting errors and duplicate frames, b) Transport Layer, to support fragmentation and reassembly of large application payload, and c) Application Layer, to interface with the DNP3 user software that monitors and controls the field devices. Every outstation consists of a collection of measurements such as breaker status, real power output, etc., which are associated with a DNP3 point and classified under one of the five groups: binary inputs (BI), binary outputs (BO), analog inputs (AI), analog outputs (AO), and counter input. The physical features consist of the information carried in the headers in the three layers of DNP3, along with the values carried by the DNP3 points in the application layer payload. Every DNP3 payload's purpose is indicated by a header in the application layer called function code (FC). For simulations, the features with FCs: 1(READ), 5(DIRECT OPERATE), 20 (ENABLE spontaneous message), 21(Disable spontaneous message), and 129 (DNP3 RESPONSE) are extracted. The details of the features are in Table 1.

V. FUSION

As presented in Fig. 2, the *Fusion* block involves different types of fusion. Intra-domain and inter-domain are considered for training the IDS using supervised and unsupervised learning techniques. A location-based fusion and visualization for causal inference of the impact of the intrusion in different locations of the network is explored. Finally, co-training with feature split is used to train the IDS using semi-supervised learning with labeled and unlabeled data.

A. INTRA-DOMAIN AND INTER-DOMAIN FUSION

The fusion of cyber sensor information from different sources is homogeneous source fusion. For example, fusing Elasticsearch logs with pyshark or raw packet capture to form the *cyber_table* is intra-domain fusion.

The fusion of cyber and physical sensor information from different sources is heterogeneous source fusion. For example, the operation of fusing *cyber_table* with *physical_table* is inter-domain fusion.

B. LOCATION-BASED FUSION

In multi-sensor data fusion, sensor location plays a major role. For example, the military uses location-based multi-sensor fusion to estimate the location of enemy troops

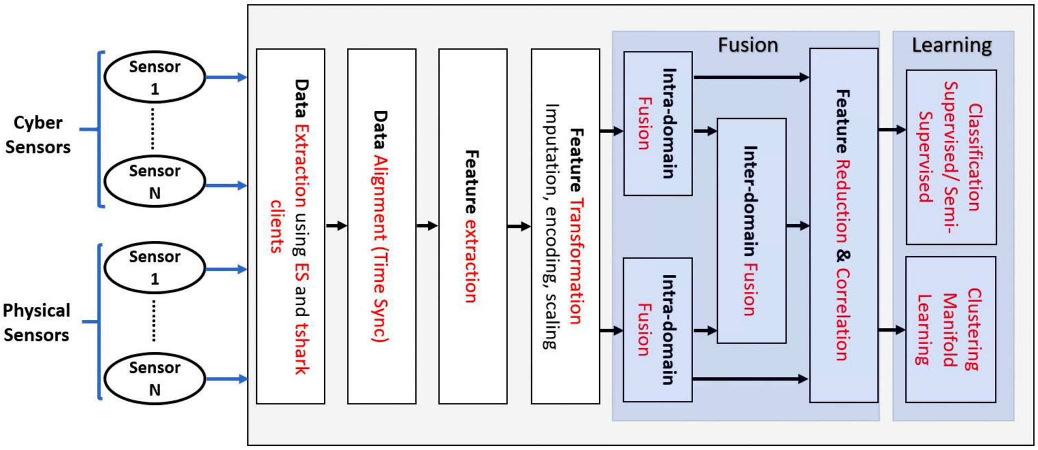

Cyber Sensors
Sensor 1
Sensor N
Physical Sensors
Sensor 1
Sensor N
Data Extraction using ES and tshark clients
Data Alignment (Time Sync)
Feature extraction
Feature Transformation
Imputation, encoding, scaling
Fusion
Intra-domain Fusion
Inter-domain Fusion
Intra-domain Fusion
Feature Reduction & Correlation
Learning
Classification Supervised/ Semi-Supervised
Clustering Manifold Learning

FIGURE 2. Centralized fusion architecture. In the autonomous architecture, the Fusion and Learning blocks will be interchanged with another Learning block post fusion.

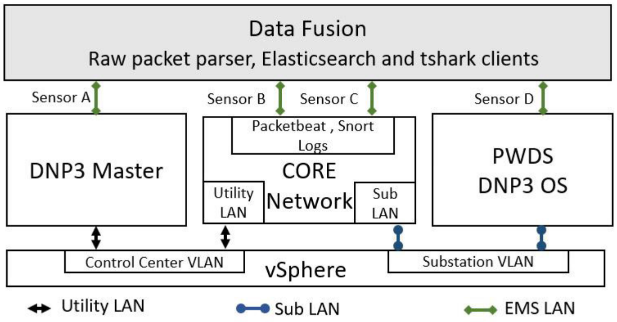


FIGURE 3. Testbed architecture with data fusion.

by amalgamating sensor information from multiple radars and submarines. The challenges associated with different locations stem from time differences in event recognition. A radar can pick up a signal with a different latency than a submarine due to the difference in communication medium as well as its location relative to the enemy troop. Similarly, sensors such as IDS, firewall alerts, and network logs are positioned at different network locations. It is essential to correlate events among different locations before merging them for inferring any attacks.

C. CO-TRAINING BASED SPLIT AND FUSION

There exist scenarios where labels cannot be captured. The co-training algorithm [51] uses feature split when learning from a dataset containing a mix of labeled and unlabeled data. This algorithm is usually preferred for datasets that have a natural separation of features into disjoint sets [74]. Since the cyber and physical features are disjoint, feature split based co-training is adopted. The approach is to incrementally build classifiers over each of the split feature sets. Here, the fused features are splitted into cyber and physical features. Each classifier, *cy_cfr* (first 17 features in Table 1) and *phy_cfr*(last 9 features in Table 1), is initialized using a few labeled records. Each classifier chooses one unlabeled record per class at every loop of co-training to add to the labeled set. The record is selected based on the highest classification confidence, as provided by the underlying classifier. Further, each classifier rebuilds from the augmented labeled set, and the process repeats. Finally, the two classifiers *cy_cfr* and *phy_cfr* obtained from the co-training algorithm gives a probability score against the classes for each record, which is added and normalized to determine the final class of the record [74]. The classifiers selected in the experiments are Linear Support Vector Machine (SVM) and Logistic Regression.

VI. DATA TRANSFORMATION

Real-time testbed data is usually insufficient, conflicting, in diverse formats, and at times lacks in certain pattern or trends. Hence, data pre-processing is essential in transforming raw data into an understandable format. The raw data extracted from multiple sensors are processed through three steps: a) data imputation, b) data encoding, c) data scaling, and d) feature reduction.

A. DATA IMPUTATION

Imputation is a statistical method of replacing the missing data with substituted values. Substitution of a data point is unit imputation, and substituting a component is item imputation. Imputation tries to preserve all the records in the data table by replacing missing data with an estimated value based on other available information or feeds from domain experts. There are other forms of imputation such as mean, stochastic, regression imputation, etc. Imputation can introduce a substantial amount of bias and can also impact efficiency. In this work, discrepancies of bias introduced due to imputation is not addressed. Since data is merged from different sources with unique features, the chances of missing data are high. Hence, imputation is performed in the dataset based on the default values in the *Def* column of Table 1.

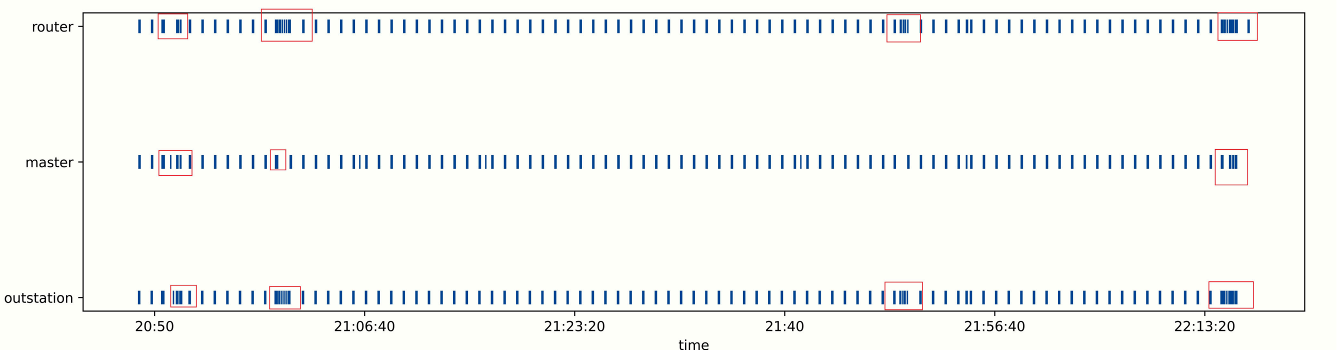


FIGURE 4. Location-based fusion from the master, outstation, and substation router. The high-density traffic observed in the places marked with red rectangles is an indicator of DoS attack. This fusion assists in causal analysis for determining the initial victim of the DoS intrusion as well as inferring the pattern of impact across other devices in the network.

B. DATA ENCODING

There are numerous features in the fused dataset which are categorical. These categorical features are encoded using the preprocessing libraries in Scikit-learn, so that the predictive model can better understand the data. There are different types of encoders such as an ordinal encoder, label encoder, one hot encoder, etc. Label encoding is preferred over one hot encoding when the cardinality of the categories in the categorical feature is quite large as it results in the issue of high dimensions. An ordinal encoder is also not considered, as it is processed on the 2D dataset (*samples*features*). Since cross-domain features are processed, encoding on individual features is performed separately using label encoding.

C. SCALING AND NORMALIZATION

Scaling and normalizing the features is essential for various ML and DL techniques such as PCA, Multi-Layer Perceptrons (MLP), SVM, etc. Though certain techniques such as Decision Trees or Random Forest are scale-invariant, it is still essential to normalize and train. Before performing normalization, log transformation and categorical encoding are performed for the features with high variance and varied range of values, respectively. Hence, both log transformation as well as scaling are evaluated. Additionally, *Min-Max scaling* is performed as considered in prior works on intrusion detection on KDD and CIDDS datasets [63].

D. FEATURE REDUCTION

Once the features from multiple sensors are merged, dimension reduction (inter-feature correlation) is performed to remove the trivial features using PCA. PCA is a linear dimensionality reduction method that uses Singular Value Decomposition on the data to project it to a lower-dimensional space [75]. The inter-feature correlation for the fused dataset from RESLab is based on the Pearson Coefficient [76], shown in as shown in Fig. 6, where it can be observed that intra-domain features have higher correlation amongst each other. There is also some correlation observed across the cyber and physical features. Features with higher correlation are more linearly dependent and thus have a similar effect on dependent variables. For example, if two features have a high correlation, one of the two features can be eliminated.

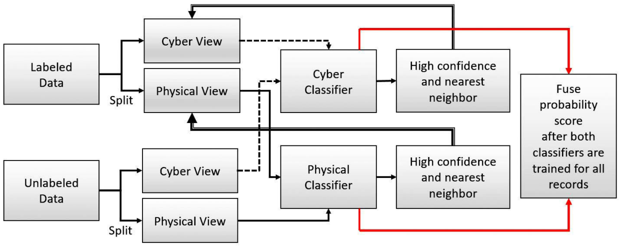

Labeled Data
Split
Cyber View
Physical View
Cyber Classifier
High confidence and nearest neighbor
Fuse probability score after both classifiers are trained for all records
Unlabeled Data
Split
Cyber View
Physical View
Physical Classifier
High confidence and nearest neighbor

FIGURE 5. Co-training based fusion for labeled and unlabeled datasets. The fused dataset is split into cyber and physical views and trained in the cyber and physical classifiers separately, finally fusing and normalizing the probability scores for final classification.

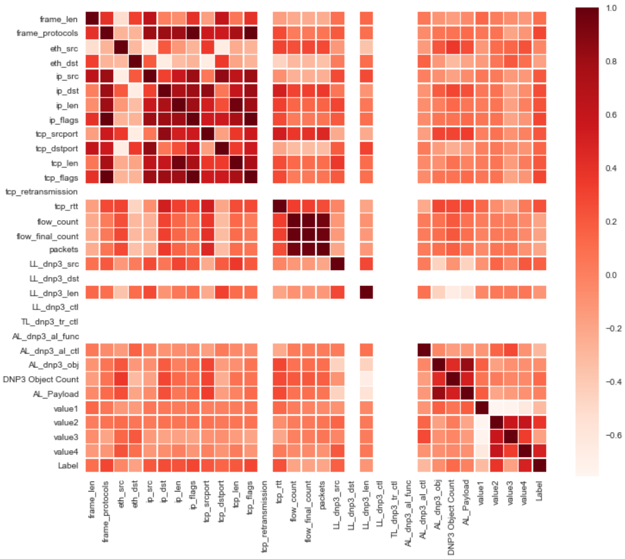


FIGURE 6. Inter-feature correlation based on Pearson Coefficient.

VII. INTRUSION DETECTION POST FUSION

After the features are extracted, merged, and pre-processed, we design IDS using different ML techniques. We have considered manifold learning and clustering as the unsupervised learning techniques, a few linear and non-linear supervised learning techniques, and co-training-based semi-supervised learning methods for training the IDS. In this section, ML techniques are briefly explained.

A. MANIFOLD LEARNING

PCA for feature reduction does not perform well when there are nonlinear relationships within the features. Manifold learning is adopted in the scenarios where the projected data in the low dimensional planar surface is not well represented and needs more complex surfaces. Multi-featured data are described as a function of a few underlying latent parameters. Hence, the data points can be assumed to be samples from a low-dimensional manifold embedded in a high-dimensional space. These algorithms try to decipher these latent parameters for low-dimensional representation of the data. There are many approaches to solve this problem, such as Locally Linear Embedding, Spectral Embedding, Multi-Dimensional Scaling, IsoMap, etc.

1) LOCALLY LINEAR EMBEDDING (LLE)

LLE computes the lower-dimensional projection of the high-dimensional data by preserving distances within local neighborhoods. It is equivalent to a series of local PCA which are globally compared to obtain the best non-linear embedding [77]. The LLE algorithm consists of 3 steps [78]: a) Compute k-nearest neighbor for a data point. b) Construct a weight matrix associated with the neighborhood of each data point. Obtains the weights that best reconstruct each data from its neighbors, minimizing the cost. c) Compute the transformed data point Y best reconstructed by the weights, minimizing the quadratic form.

2) SPECTRAL EMBEDDING

Spectral embedding builds a graph incorporating neighborhood information. Considering the Laplacian of the graph, it computes a low dimensional representation of the data set that optimally preserves local neighborhood information [79]. Minimization of a cost function, based on the graph ensures that points closer on the manifold are mapped closer in the low dimensional space, preserving local distances [77]. The Spectral Embedding algorithm consists of 3 steps: a) Weighted Graph Construction in which raw data are input into a graph representation using an adjacency matrix. b) Construction of unnormalized and a normalized graph Laplacians as $L = D{-}A$ and $L = D^{-0.5}(D-A)D^{-0.5}$, respectively. c) Finally, partial eigenvalue decomposition is done on the graph Laplacian.

3) MULTI DIMENSIONAL SCALING (MDS)

MDS performs projection to lower dimensions to improve interpretability while preserving ‘dissimilarity’ between the samples. It preserves the dissimilarity by minimizing the square difference of the pairwise distances between all the training data between the projected, lower-dimensional and the original higher-dimensional space,

$$\mathrm{Diff}_P\left(X_1,\ldots,X_n\right)=\left(\sum_{i=1}^{n}\sum_{j=1|i\neq j}^{n}\left(\left\|x_i-x_j\right\|-\delta_{i,j}\right)^2\right)^{1/2} \tag{5}$$

where $\delta_{i,j}$ is the general dissimilarity metric in the original higher dimensional space and $\left\|x_i-x_j\right\|$ is the projected/ lower dimensional dissimilarity pairwise between training samples i and j. The model can be finally validated by a scatter plot of pairwise distance in projected and original space. There are two types of MDS: Metric and Non-Metric based. In Metric MDS, the distances between the two points in projection are set to be as close as possible to the dissimilarity (or distance) in original space. Non-metric MDS tries to preserve the order of the distances and hence seeks a monotonic relationship between the distances in the embedded and original space.

4) T-SNE VISUALIZATION

The manifold learning technique called t-distributed Stochastic Neighbor Embedding is useful to visualize high-dimensional data, as it reduces the tendency of points to crowd together at the center. This technique converts similarities between data records to joint probabilities and then tries to minimize the Kullback-Leibler divergence (a technique used to compare two probability distributions) between the joint probabilities of the low-dimensional embedding and the high-dimensional data using gradient descent. The only issue with this technique is that it is computationally expensive and is limited by two or three embeddings in some methods. In the intrusion detection methods, the purpose is to evaluate if in the low-dimensional embedding one can find some correlation of the data points with the labels.

5) IsoMap EMBEDDING

IsoMap stands for isometric mapping and is an extension to the MDS technique discussed earlier. It uses geodesic paths instead of euclidean distance for nonlinear dimensionality reduction. MDS tries to preserve large pairwise distance over the small pairwise distance. IsoMap first determines a neighborhood graph by finding the k-nearest neighbor of each point, further connecting these points in the graph, and assigns weights. Then, it computes the shortest geodesic path between all pairs of points in the graph, to use this distance measure between connected points as weights to apply MDS to the shortest-path distance matrix [80].

B. CLUSTERING

One of the fundamental problems in multi-sensor data fusion is *data association*, where different observations in the dataset are grouped into clusters [25]. Hence, various clustering techniques are considered for data association.

1) K-MEANS CLUSTERING

The k-means algorithm clusters data by separating samples in n groups of equal variance, minimizing a criterion known as inertia. The algorithm starts with a group of randomly selected centroids, which are used as the beginning points for every cluster, then performs iterative calculations to optimize the positions of the centroids by minimizing inertia. The process stops when either the centroids have stabilized or the number of iterations has been achieved.

2) SPECTRAL CLUSTERING

The main concept behind spectral clustering is the graph Laplacian matrix. The algorithm takes the following steps [81]:

1) Construct a similarity graph either based on an ϵ-neighborhood graph, a $k - nearest$ neighbor graph, or a fully connected graph.
2) Compute the normalized Laplacian L.
3) Compute the first k eigen-vectors $u_1, u_2 \ldots, u_k$ of L. The first eigen-vectors are related to the k smallest eigen values of L.
4) Let $U \in R^{n*k}$ be the matrix containing the vectors $u_1, u_2 \ldots, u_k$ as columns.
5) For $i = 1, , , , n$, let $y_i \in R^k$ be the vector corresponding to the i^{th} row of U.
6) Cluster points (y_i) in R^k with k-means algorithm into clusters $C_1, \ldots C_k$.

3) AGGLOMERATIVE CLUSTERING

Agglomerative clustering is done in a bottom-up manner, where at the beginning, each object belongs to one single-element cluster, which are the leaf clusters of a dendogram. At each step of the algorithm, the two clusters that are most similar (based on a similarity metric such as distance) are combined into a larger cluster. The procedure is followed until all points are members of a single big cluster. The steps form a hierarchical tree, where a distance threshold is used to cut the tree to partition the data into clusters. As per scikit, this algorithm recursively merges the pair of clusters that minimally increases a given linkage distance [82]. The parameter *distance_threshold* in the scikit-learn implementation is used to cut the dendrogram.

4) BIRCH CLUSTERING

The Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) [83] algorithm is more suitable for the cases where the amount of data is large and the number of categories K is also relatively large. It runs very fast, and it only needs a single pass to scan the data set for clustering.

C. SUPERVISED LEARNING

Though manifold learning and clustering techniques help visualize and cluster the data samples in the intrusion time-interval from the non-intrusion ones, still the results of these techniques are hard to validate without any labels, hence various supervised learning techniques are also considered in designing the anomaly-based IDS.

1) SUPPORT VECTOR CLASSIFIER

Support vector machine builds a hyperplane or set of hyperplanes in a higher dimensional space which are further used as a decision surface for classification or outlier detection. It is a supervised learning based classifier which performs better even for scenarios with higher feature size than the sample size. The decision function, or support vectors, defined using the kernel type such as sigmoid, polynomial, linear or radial basis function plays a major impact on the classifier performance. Different variants of SVCs have been predominantly proposed in intrusion detection solutions [84], [85].

2) LOGISTIC REGRESSION (LR) CLASSIFIER

LR is a classification algorithm, used mainly for discrete set of classes. It is a probability-based classification technique which minimizes the error cost using the logistic sigmoid function. It uses the gradient descent technique to reduce the error cost function. Industries make a wide use of it, since it is very efficient and highly interpretable [86].

3) NAIVE BAYES (NB) CLASSIFIER

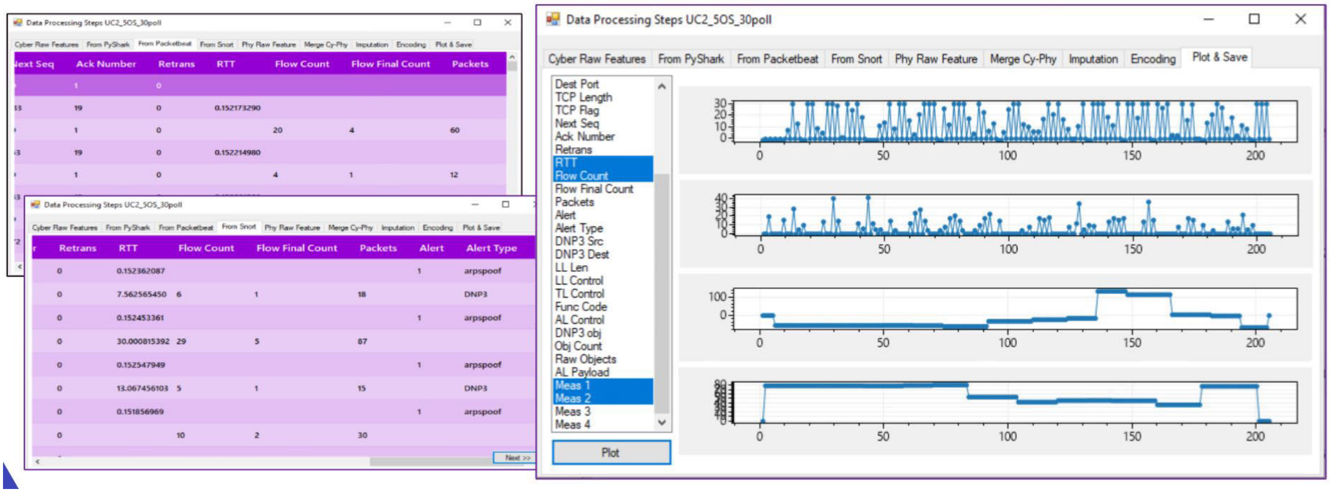
NB is a supervised learning technique using Bayes Theorem, with the naive assumption of independent features, conditioned on the class. Based on feature likelihood distribution, they possess different forms: Gaussian, Bernoulli, Categorical, Complement, etc. Though it is computationally efficient, the selection of feature likelihood may alter results. In spam filtering, text classification, and also network intrusion detection, it is used profusely [87]. An NB based solution was proposed for IDS in a smart meter network [88].

4) DECISION TREE (DT) CLASSIFIER

The advantage of using DT is that it requires the least data transformation. Fundamentally, it creates internal models that predict the target class by learning decision rules inferred from the features. This technique sometimes meets with over-fitting issues while learning complex trees that are hard to generalize. Hence, it adopts pruning techniques such as reducing the tree max-depth to deal with over-fitting. If data in the samples are biased, it may be highly likely to create biased trees. The computation cost of using this classifier is logarithmic in the number of data records. It has been used in the protocol classification problem [89], [90] for classifying anomalous packets.

5) RANDOM FOREST (RF) CLASSIFIER

Basically, RF creates decision trees on randomly picked data samples, and further computes a prediction from each tree and selects the best solution through voting. More trees result in a more robust forest. It is an ensemble-based classifier in which a diverse collection of classifiers (decision trees) is constructed by incorporating randomness in tree construction. Randomness decreases the variance to address the overfit issues prevailing in DT. Compared with SVMs, RF is fast and works well with a mixture of numerical and



Data Processing Steps UC2_5OS_30poll

Cyber Raw Features | From PyShark | From Packetbeat | From Snort | Phy Raw Feature | Merge Cy-Phy | Imputation | Encoding | Plot & Save

ext Seq	Ack Number	Retrans	RTT	Flow Count	Flow Final Count	Packets
	1	0				
33	19	0	0.152173290			
	1	0		20	4	60
3	19	0	0.152214980			
	1	0		4	1	12

Data Processing Steps UC2_5OS_30poll

Cyber Raw Features | From PyShark | From Packetbeat | From Snort | Phy Raw Feature | Merge Cy-Phy | Imputation | Encoding | Plot & Save

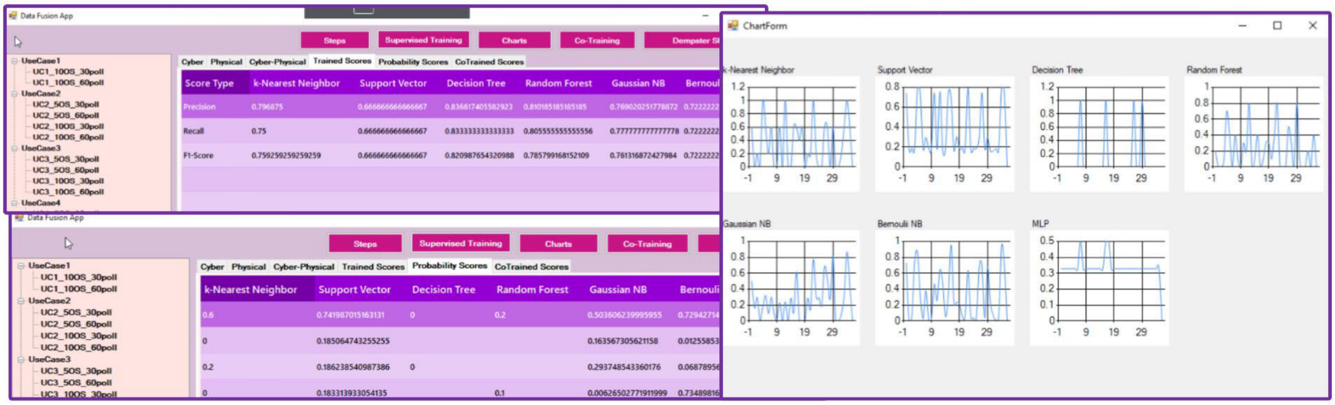
Retrans	RTT	Flow Count	Flow Final Count	Packets	Alert	Alert Type
0	0.152362087				1	arpspoof
0	7.562565450	6	1	18		DNP3
0	0.152453361				1	arpspoof
0	30.000815392	29	5	87		
0	0.152547949				1	arpspoof
0	13.067456103	5	1	15		DNP3
0	0.151856969				1	arpspoof
0		10	2	30		

Next >>

Data Processing Steps UC2_5OS_30poll

Cyber Raw Features | From PyShark | From Packetbeat | From Snort | Phy Raw Feature | Merge Cy-Phy | Imputation | Encoding | Plot & Save

FIGURE 7. Left figures illustrate features extracted from multiple stages such as after connection with the Packetbeat server and the Snort IDS. Right-side figure illustrates the imputed and encoded features graphically.



Data Fusion App

Cyber | Physical | Cyber-Physical | Trained Scores | Probability Scores | CoTrained Scores

Score Type	k-Nearest Neighbor	Support Vector	Decision Tree	Random Forest	Gaussian NB	Bernoul
Precision	0.796875	0.666666666666667	0.836617405582923	0.810185185185185	0.769020251778872	0.7222222
Recall	0.75	0.666666666666667	0.833333333333333	0.805555555555556	0.777777777777778	0.7222222
F1-Score	0.759259259259259	0.666666666666667	0.820987654320988	0.785799168152109	0.761316872427984	0.7222222

Data Fusion App

Cyber | Physical | Cyber-Physical | Trained Scores | Probability Scores | CoTrained Scores

k-Nearest Neighbor	Support Vector	Decision Tree	Random Forest	Gaussian NB	Bernouli
0.6	0.741987015163131	0	0.2	0.503606239995955	0.72942714
0	0.185064743255255			0.163567305621158	0.01255853
0.2	0.186238540987386	0		0.293748543360176	0.06878956
0	0.183313933054135		0.1	0.00626502771911999	0.73489816

ChartForm

FIGURE 8. Left figures illustrate precision, recall, F1-score as well as the raw probability scores from the classifiers. Right-side figure visualizes the probability scores for each technique graphically.

categorical features. It has a variety of applications, such as in recommendation engines, image classification, and feature selection. Due to its variance-reduction feature and its low need for data pre-processing, it is also preferred in the cyber security area [91], [92].

6) NEURAL NETWORK (NN) CLASSIFIER

Neural networks are effective in the case of complex non-linear models. In the IDS classification problem, MLP is used as the supervised learning algorithm. It learns a non-linear function approximator whose inputs are the features for a record and outputs the class. Unlike a logistic regressor, it comprises multiple hidden layers. A major issue with NN models is they require a large set of hyper tuning parameters such as the number of hidden neurons, layers, iterations, dropouts, etc., that can affect the hyper-parameter tuning process for improving accuracy. Additionally, NNs are quite sensitive to feature scaling. Following Occam's razor, security professionals tend to avoid neural networks in intrusion detection, wherever possible. Still, NNs can be explored to capture temporal patterns with the use of Recurrent Neural Networks and spatial patterns with Graph Neural Networks.

VIII. DATA FUSION SOFTWARE APPLICATION

A desktop application as a data fusion framework is developed for data aggregation, feature extraction, transformation, fusion, and learning purposes, as illustrated in Fig. 2. The purpose of the application is to extract sensor information for different use cases and visualize features at different stages. In Fig. 7 and 8, the application visualizes the features extracted from multiple sources and the results of supervised learning techniques, respectively. In the application, the user can also select features based on correlations to infer the timing and cause of the attack.

This application is scalable and re-usable for different use-cases and is currently deployed in the Data Fusion block in the testbed (Fig. 3). This application will be further augmented to provide cyber network reconfiguration under different types of attacks detected. For example, a classifier detecting an ARP spoof attack will trigger an ARP-tables based filtering in the firewall to regulate ARP cache poisoning traffic from the intruder.

IX. RESULTS AND ANALYSIS

In this section, the improvement of the detection performance of the IDS, when a fused dataset is considered in comparison to the use of only cyber or physical features, is studied. The IDS is designed as a classifier when training with supervised and semi-supervised based ML techniques. The IDS's performance is analyzed based on the different types of MiTM attack carried out in the RESLab testbed. For supervised learning techniques, the impact of labeling and feature reduction on the detection accuracy is investigated. For unsupervised learning techniques, comparison of the performance of the clustering techniques is based on different metrics. In most of the experiments, the highest scores for either 2 or 3 clusters were expected, since the objective was to cluster attacked traffic from non-attacked, with the third cluster being undetermined. Additionally, a co-training-based semi-supervised learning technique is tested, by assuming a loss of labels for some experiments and comparing them with supervised learning techniques.

A. SUPERVISED TECHNIQUE INTRUSION DETECTION WITH SNORT ALERT AS LABEL

1) METRICS FOR EVALUATION

The IDS performance is evaluated by classifier accuracy computed using metrics such as *Recall*, *Precision*, and *F1-score*. A recall is the ratio of the true-positives to the sum of true-positives and false-negatives. Precision is the ratio of the true-positives to the sum of true-positives and false-positives. High precision is ensured by a low false-positive rate. A high recall is an indication of low false-negative rate. False negatives are highly unwanted in security, since an undetected attack may result in more privilege escalations and can impact a larger part of network. False positives are expensive, as time and money is invested for security professionals to investigate a non-critical alert. Hence, harmonic mean of recall and precision, called F1-score, is a preferred metric for a balanced evaluation.

2) LABELS EVALUATION

The performances are compared, considering labels from Snort alerts and labels based on the intruders' attack windows, to train the supervised learning-based IDS classifiers. The intruders' attack window is the difference between the attack script end and start time. Every record is labeled in this window belonging to the compromised class. It is interesting to observe from Table 2 that the classifier trained using the attack window label performed better than the Snort labels, based on the average F1-score, Recall, and Precision. These metrics are computed by taking the average of all the metrics from different use cases. This analysis indicates that training a model from well-known IDS may not act as an ideal classifier for intrusion detection. Hence, for further studies, the classifier is trained using the attack window-based label.

TABLE 2. Comparison of the labels using a different classifier based on the evaluation metrics.

Classifier	Snort Label			Label from Attack Window		
Avg.	F1 score	Rec.	Prec.	F1 score	Rec.	Prec.
SVC	.566	.69	.496	.752	.776	.799
DT	.738	.73	.757	.909	.909	.92
RF	.764	.789	.776	.891	.896	.903
GNB	.598	.574	.745	.724	.729	.748
BNB	.57	.589	.621	.634	.655	.676
MLP	.561	.671	.491	.621	.695	.604

3) USE CASE SPECIFIC EVALUATION

The datasets constructed from four use cases is analyzed based on different strategies of FDI and FCI attacks (measurement and control, respectively). These cases use different polling rates and DNP3 masters on the synthetic 2000-bus grid case illustrated in the RESLab paper [14]. Use Case 1 and 2 are FCI attacks on binary and mixed binary/analog commands from the control center to some selected outstations, selected from prior work on graph-based contingency discovery [93]. Use Case 3 and 4 are a mix of FCI and FDI attacks. These use cases differ based on the type and sequence of modifications done by the intruder, as shown in Table 3.

Due to the variation of attempts an intruder needs to take to implement the use cases, the number of samples collected for every scenario differs. In the MLP-based classifier, the number of samples plays a vital role; hence, MLP performs better for scenarios with the number of DNP3 masters equal to 10 versus 5 and with a DNP3 polling interval of 30 s versus 60 s. The DT and RF classifiers outperform the other classifiers in almost all the scenarios. The NB classifiers, both Gaussian and Bernoulli, need the features to be independent for optimal performance. Since most of the features are strongly correlated based on Fig. 6, the performance of NB is relatively weak compared to other classifiers. Usually, Gaussian Naive Bayes (GNB) is considered for features that are continuous and Bernoulli Naive Bayes (BNB) for discrete features. In fused dataset, since both types of features exist, both techniques are considered for evaluation. In the majority of the scenarios, GNB performed better than BNB, indicating the physical features have more impact on the detection compared to categorical cyber features. Table 4 shows the comparison of classifiers for different use cases, and Table 5 shows the comparison using grid search cross-validation based tuning of hyper-parameters for each classifier.

4) IMPACT OF FUSION

The classifier's performance is evaluated by considering pure physical and pure cyber-based intra-domain fusion as well as cyber-physical inter-domain fusion. The pure physical and cyber physical based fusion outperforms pure-cyber based fusion for all the classifiers shown in Table 6. Hence, it indicates that the introduction of physical side features can

TABLE 3. Use cases based on the type and sequence of modifications.

FCI		FCI with FDI	
UC1	UC2	UC3	UC4
Binary Commands	Analog, Binary Commands	Measurements=>Commands	Measurements=> Commands=>Measurements

TABLE 4. Comparison of the classifier based on the scenarios i.e. use cases, number of masters and the polling interval (PI) in sec.

Scenarios			Classifiers					
uc	masters	PI	SVC	DT	RF	GNB	BNB	MLP
UC1	10	30	.70	.74	.75	.59	.70	.70
	10	60	.78	.87	.81	.75	.49	.58
UC2	5	30	.88	.76	.92	.73	.52	.86
	5	60	.88	.89	1.0	.94	.89	.66
	10	30	.84	.93	.93	.73	.59	.77
	10	60	.64	.97	.88	.33	.58	.52
UC3	5	30	.95	.98	.93	.93	.57	.72
	5	60	.50	1.0	.88	.72	.33	.40
	10	30	.85	1.0	.97	.83	.66	.86
	10	60	.89	.98	.91	.84	.73	.91
UC4	5	30	.59	.86	.88	.56	.54	.39
	5	60	.63	.81	.77	.74	.77	.31
	10	30	.65	.96	.97	.63	.78	.57
	10	60	.75	.98	.88	.83	.80	.50

TABLE 5. Optimal Hyper-parameter with GridSearch Comparison of the classifier based on the scenarios i.e. use cases, number of masters and the polling interval (PI) in sec.

Scenarios			Classifiers					
uc	masters	PI	SVC	DT	RF	GNB	BNB	MLP
UC1	10	30	.70	.78	.75	.70	.69	.70
	10	60	.54	.87	.81	.78	.52	.7
UC2	5	30	.51	.88	.84	.72	.51	.67
	5	60	.66	1.0	1.0	.83	.89	.62
	10	30	.45	.94	.89	.81	.44	.86
	10	60	.52	.97	.85	.75	.61	.58
UC3	5	30	.36	.98	.93	.93	.50	.91
	5	60	.40	1.0	.96	.88	.26	.44
	10	30	.41	1.0	.99	.84	.63	.69
	10	60	.40	.93	.88	.89	.76	.82
UC4	5	30	.39	.97	.93	.57	.56	.61
	5	60	.31	.63	.68	.65	.77	.68
	10	30	.44	.96	.95	.65	.78	.65
	10	60	.50	.88	.85	.80	.80	.50

improve the accuracy of conventional IDS that only considers network logs in the communication domain. The pure physical features relatively performed better than cyber-physical because, in the testbed, only a few features (i.e. measurements for the impacted substation) are considered for extraction. If all the measurements is considered from the grid simulation, the detection accuracy will decrease due to feature explosion. Feature reduction techniques such as PCA for the physical features may not be an ideal solution for a huge synthetic grid.

5) IMPACT OF FEATURE REDUCTION

In this subsection, feature reduction techniques such as PCA and Shapiro ranking are considered for feature reduction and feature filtering to evaluate the performance of the IDS. Table 7 illustrates the performance scores for different classifiers with PCA transformed features and Shapiro features selected for scores more than 0.7. It can be observed that except for the DT and RF, other classifier's performance improved by both operations. DT and RF behave the best when most of the features are kept intact. In most of the cases, the selection of features based on Shapiro features performed better than PCA transformation. Still, the total variance threshold taken may impact the number of principal components considered, which can affect the results.

TABLE 6. Comparison of the classifier with pure cyber fusion, pure physical fusion, and cyber-physical fusion features.

Clfr	Pure Cyber			Pure Physical			Cyber Physical		
Avg.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.
SVC	.62	.68	.59	.75	.77	.80	.75	.77	.80
DT	.77	.77	.77	.93	.93	.94	.91	.91	.92
RF	.69	.69	.68	.92	.92	.93	.89	.90	.90
GNB	.58	.57	.59	.78	.77	.81	.72	.73	.75
BNB	.52	.56	.55	.65	.68	.66	.63	.66	.68
MLP	.56	.66	.53	.72	.76	.77	.62	.70	.61

TABLE 7. Comparison of the classifier with all features, reduced feature with PCA transformation, and feature selection based on shapiro ranking.

Clfr	All Features			PCA			Shapiro Ftrs ≥ 0.7		
Avg.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.
SVC	.75	.77	.80	.77	.80	.81	.77	.78	.79
DT	.91	.91	.92	.82	.82	.83	.89	.89	.91
RF	.89	.90	.90	.86	.86	.87	.84	.84	.84
GNB	.72	.73	.75	.77	.78	.78	.83	.84	.87
BNB	.63	.66	.68	.74	.76	.76	.80	.82	.86
MLP	.62	.70	.61	.61	.68	.64	.50	.64	.41

B. UNSUPERVISED LEARNING TECHNIQUES

1) METRICS FOR EVALUATION

For evaluating the performance of the clustering techniques, the Silhouette scores, Calinski Harabasz score, Adjusted Rand score, and Davies Bouldin scores are considered. The *Silhouette score* (S) is the mean Silhouette Coefficient of all samples. The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample, using $\frac{b-a}{max(a,b)}$. The *Calinski Harabasz score* (CH) is computed based on [94]. It is the ratio between the within-cluster dispersion and the between-cluster dispersion. The *Rand Index* computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs assigned in the same or different clusters in the predicted and true clusterings. This index is further adjusted to be called the Adjusted Rand Index (AR). The *Davies Bouldin score* (DB) is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances [95]. Thus, clusters

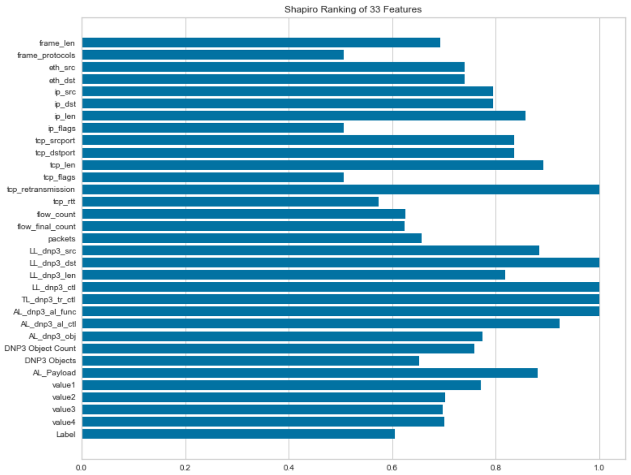


FIGURE 9. Ranking feature importance for extracting features. Of all the features, scores above 0.7 is selected for training.

that are farther apart and less dispersed will result in a better score.

2) CLUSTERING

Prior to the clustering techniques, the datasets are sclaed and normalized using scaler and normalize functions since otherwise there will be feature-based bias. Four types of clustering techniques: Agglomerative, k-means, Spectral, and Birch clustering, are implemented, to evaluate the optimal number of clusters based on the S, CH, AR, and DB scores. For determining the clusters, the samples from all the use-cases are merged, to form a larger dataset and then trained the clustering methods by tuning the number of clusters hyper-parameter (N_c) from 2 to 10. Fig 10 (a-e) show the clustered plots using Agglomerative clustering with a different number of clusters. The number of clusters, or centroids, are selected for hyper-parameter tuning since it is found to be the most important factor for the success of the algorithm [96]. Ideally, there need to be 3 clusters for un-attacked, attacked with DNP3 alerts, and attacked with ARP alerts, but the distance metric considered results in a greater number of clusters in some methods. Among all the clustering techniques presented in the previous section, the affinity propagation technique does not converge to obtain the exemplars with default parameters ($damping$ =50, $convergence\_iter$ =200). Hence, the damping and maximum convergence iteration parameters are increased to 0.95 and 2000 respectively, resulting in 34 clusters. The S, CH, DB, and AR scores obtained are 0.605, 3658.1, 0.736, and 0.00085 respectively.

3) IMPACT OF FUSION

Considering only physical side features, most of the evaluation metrics computed very low or negative (in the case of Adjusted Rand index) values, indicating inefficient clusters. The scores of the optimal clusters with combined cyber-physical features had an AR score of more than 0.8, but its maximum is 0.01 for 6 clusters with only physical features. The pure cyber features performed similar to the cyber-physical case, but the scores are less compared to the merged features. Hence, it is essential to fuse cyber and physical features prior to performing clustering-based unsupervised learning. Table 8 shows the optimal cluster, based on the scores, with considering cyber and physical features separately. While, Table 9 shows the optimal cluster with features merged from cyber and physical domain. Using fused feature, optimal cluster is found to be three in majority cases.

TABLE 8. Comparison of Optimal clusters (Opt N_c) using different algorithm considering pure cyber and physical features separately.

	Pure Cyber				Pure Physical			
Clustering	S	CH	AR	DB	S	CH	AR	DB
Agglo.	3	5	2	6	2	6	6 (neg)	2
K-means	3	6	2	5	3	6	6 (neg)	2
Spectral	3	5	2	6	3	3	6 (neg)	2
Birch	3	3	2	2	3	3	3 (neg)	2

TABLE 9. Optimal clusters (Opt N_c) using different algorithm obtained using four different evaluation metric with cyber and physical features combined.

Clustering Algo	S	CH	AR	DB
Agglomerative	3	3	2	3
K-means	3	3	2	3
Spectral	3	5	2	3
Birch	3	3	2	3

4) ROBUSTNESS

The robustness of the clustering techniques can be evaluated based on the variance of these evaluation metrics with respect to a) hyper-parameter tuning and b) dataset alterations. In the first case, the mean, variance, and normalized variance ($NVar = \frac{sd}{mean}$) of the evaluation metric S, CH, AR, and DB are computed by altering N_c from 2 to 10 and using the complete dataset extracted for all the use cases. In the second case, similar statistics are computed by keeping the number of clusters fixed at $N_c = 3$ and altering the dataset i.e. by using different use cases. A clustering technique with lower normalized variance is more robust, and a better mean score is more accurate. Based on the silhouette scores (S) from Table 10, k-mean based clustering is found to be more robust to varying data sources and has a better mean score, but a main limitation of k-means is its strong dependence on N_c. Still, k-means is used in many practical situations such as anomaly detection [97] due to its low computation cost.

5) MANIFOLD LEARNING

Manifold learning is adopted for visualization. Classification techniques need to be employed on the features projected in the lower dimensions using these embeddings for quantitative comparisons. The performance of manifold learning methods are evaluated by testing them with the classifiers presented in the previous subsection. Table 11 presents the comparison of the LLE, MDS, spectral, t-SNE, and IsoMap [98] embeddings considered for classification using SVC, k-NN, DT,

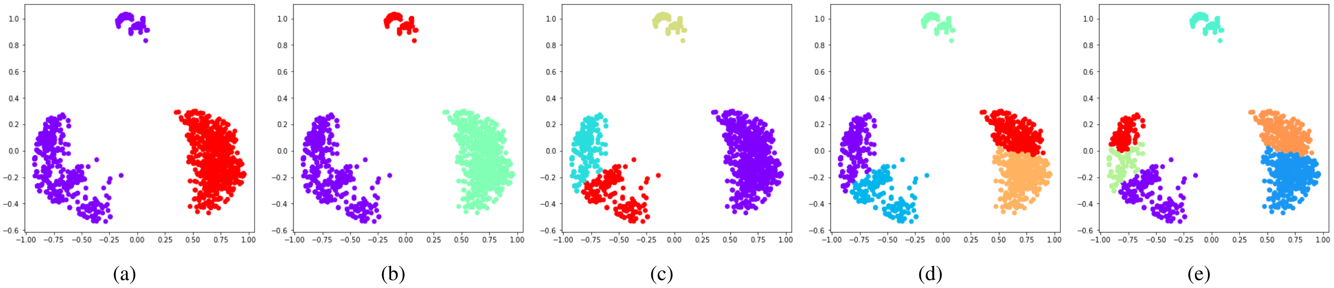


FIGURE 10. Agglomerative clustering with different number of clusters. Clustering with size 2 and 3 outperforms others, validating the detection accuracy of a attacked traffic from a non-attacked one.

TABLE 10. Evaluation of the Robustness of the Clustering Algorithm by varying hyper-parameters and data source.

Scenarios		Effect of Parameters			Effect of Data Alt.		
Met	Algo	Mean	Var	NVar	Mean	Var	NVar
S	Agg	.52	.0175	.254	.609	.01	.164
	K-m	.54	.013	.212	**.615**	.008	**.145**
	Spec	.504	.021	.287	.581	.015	.213
	Bir	**.74**	.011	**.146**	.599	.010	.172
CH	Agg	9965	5.5×10^6	**.235**	337	36880	.569
	K-m	**10822**	6.6×10^7	.237	**346**	35690	.545
	Spec	8765	1.0×10^7	.362	311	34047	.592
	Bir	10484	1.3×10^7	.349	331	35637	.57
AR	Agg	.703	.035	.266	.534	.029	.319
	K-m	.672	.027	**.248**	.529	.022	**.281**
	Spec	**.714**	.039	.278	**.638**	.049	.349
	Bir	.342	.014	.35	.589	.047	.368
DB	Agg	.026	0.0	**.32**	.053	.003	1.038
	K-m	.54	0.0	.322	.063	.003	.925
	Spec	.504	0.0	.559	.058	.003	1.037
	Bir	**.74**	0.0	1.344	**.065**	.003	**.895**

TABLE 11. Comparison of the different manifold learning embeddings considered with different classifiers.

Clfr→	SVC	k-NN	DT	RF	GNB	BNB	MLP
Manifold ↓	F1 scores						
LLE	.66	.74	.66	.64	.38	.39	.49
MDS	.65	.78	.77	.80	.54	.48	.55
Spectral	.61	.75	.73	.75	.61	.62	.54
t-SNE	.64	.74	.73	.76	.63	.57	.63
IsoMap	.65	.78	.77	.79	.54	.48	.55

RF, GNB, BNB, and MLP. Inter-domain fusion does not gain much from manifold learning, but an interesting observation is made on the decrease in the difference of F1 scores among the high performing DT and RF classifiers, with the low performing SVC and k-NN classifiers. Hence, it is inadvisable to perform manifold learning for the datasets, if training using Decision Tree or Random Forest. The IsoMap embedding that preserves local features of the data by first determining neighbor-hood graph and uses MDS in its last stage performs better than MDS for all the classifiers only with the exception of SVC.

C. SEMI-SUPERVISED LEARNING

1) CO-TRAINING

For co-training, first the dataset is split into labeled and unlabeled sets randomly in the ratio of 1:2. In the real world, this randomness may be caused due to accidental cessation of the Snort application or if a network security expert cannot make an inference of intrusion. Further, both the labeled and unlabeled data are split into cyber and physical views consisting of respective features. In these experiments, the supervised learning techniques are compared on the labeled dataset with the co-training technique which uses supervised learning cyber and physical classifiers, as shown in Fig. 5. It is expected to have a reduction in performance from supervised learning techniques, due to lack of labels for some samples, but it can be observed from Table 12, that the co-training based classification outperforms supervised for some classifiers such as *LR*,*GNB*,*BNB*,*MLP* and performs at par with other classifiers with a difference of a mere 8 percent in the case of *RF*. The probable reason for improved performance using co-training may be due to the training of two different classifiers using intra-domain features.

TABLE 12. Comparison of the classifier using supervised and co-training based unsupervised learning.

Classifier	Supervised			Co-Training		
	F1-score	Rec.	Prec.	F1-score	Rec.	Prec.
LR	.63	.67	.64	.64	.73	.58
SVC	.63	.67	.64	.59	.70	.52
DT	.69	.71	.69	.64	.71	.65
RF	.73	.77	.72	.65	.72	.72
GNB	.28	.33	.66	.30	.32	.56
BNB	.53	.51	.67	.58	.66	.52
MLP	.59	.71	.51	.61	.71	.55

X. CONCLUSION

A data fusion framework for detecting false command and measurement injections due to cyber intrusion is presented in this paper. To design an IDS that uses cyber and physical features, features from cyber and physical sensors are aggregated and the data aligned, then we perform pre-processing techniques, followed by inter-domain fusion.

The results find that classifier performance improves on an average of 15-20% (based on F1-score) when cyber-physical features are considered instead of pure cyber features. Results also show that the performance improves on an average of 10-20% (based on F1-score) when labels from Snort are replaced by the labels considered based on intrusion timestamps. From the evaluations of the IDS, it is also found that scenarios with balanced and larger records result in better performance. Additionally, the co-training-based semi-supervised learning technique, which is realistic for a real world scenario, is found to perform similarly to supervised techniques and even better by 2-5% (based on F1-score) using some classifiers. Among the unsupervised learning techniques, the k-mean clustering technique is found to be more robust and accurate. Moreover, training the classifier with the embeddings from manifold learning does not improve the accuracy. Hence, manifold learning should only be considered for visualization rather than relied on for accuracy.

It is believed by us, that the fused dataset [99], the data fusion engine [100], and results provided are one of the first publicly available studies with cyber and physical features, particularly for power systems, where the experimental data is collected from a testbed that contains both cyber and physical emulation. This benefits research in multi-disciplinary areas such as cyber-physical security and data science.

REFERENCES

[1] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[2] M. Kordestani and M. Saif, "Data fusion for fault diagnosis in smart grid power systems," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Mar. 2017, pp. 1–6.

[3] 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.

[4] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 8, pp. 1–14, Aug. 2018.

[5] N. Clarke, S. Furnell, G. Tjhai, and M. Papadaki, "Investigating the problem of IDS false alarms: An experimental study using snort," in *Proc. IFIP Int. Inf. Secur. Conf.*, vol. 278, Jul. 2008, pp. 253–267.

[6] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118.1-2011 (Revision IEEE Std C37.118-2005), 2011, pp. 1–61.

[7] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Comput. Mater.*, vol. 5, no. 1, p. 83, Aug. 2019, doi: 10.1038/s41524-019-0221-0.

[8] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. D. Yao, "Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities," 2021, *arXiv:2003.13213*. [Online]. Available: http://arxiv.org/abs/2003.13213

[9] H. Liu, L. T. Yang, Y. Guo, X. Xie, and J. Ma, "An incremental tensor-train decomposition for cyber-physical-social big data," *IEEE Trans. Big Data*, vol. 7, no. 2, pp. 341–354, Jun. 2021.

[10] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *Comput. Secur.*, vol. 30, nos. 6–7, pp. 353–375, Sep. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404811000691

[11] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2015.

[12] *TShark Manual*. Accessed: Jan. 2021. [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html

[13] A. D. Orebaugh, S. Biles, and J. Babbin, *Snort Cookbook*. Sebastopol, CA, USA: O'Reilly Media, 2005.

[14] A. Sahu, P. Wlazlo, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Design and evaluation of a cyber-physical resilient power system testbed," Nov. 2020, *arXiv:2011.13552*. [Online]. Available: http://arxiv.org/abs/2011.13552

[15] B. Genge, C. Siaterlis, and G. Karopoulos, "Data fusion-base anomay detection in networked critical infrastructures," in *Proc. 43rd Annu. IEEE/IFIP Conf. Dependable Syst. Netw. Workshop (DSN-W)*, Jun. 2013, pp. 1–8.

[16] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," E-ISAC, SANS, Washington, DC, USA, Mar. 2016.

[17] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," in *Proc. Int. Conf. Internet Things, 4th Int. Conf. Cyber, Phys. Social Comput.*, Oct. 2011, pp. 380–388.

[18] S. Zonouz, C. M. Davis, K. R. Davis, R. Berthier, R. B. Bobba, and W. H. Sanders, "SOCCA: A security-oriented cyber-physical contingency analysis in power infrastructures," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 3–13, Jan. 2014.

[19] K. Davis, R. Berthier, S. Zonouz, G. Weaver, R. Bobba, E. Rogers, P. Sauer, and D. Nicol, "Cyber-physical security assessment (cypsa) for electric power systems," IEEE-HKN: THE BRIDGE, The Magazine of IEEE-Eta Kappa Nu, May 2016, vol. 112, no. 2.

[20] A. Sahu and K. Davis, "Structural learning techniques for Bayesian attack graphs in cyber physical power systems," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2021, pp. 1–6.

[21] C. Vellaithurai, A. Srivastava, S. Zonouz, and R. Berthier, "CPIndex: Cyber-physical vulnerability assessment for power-grid infrastructures," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 566–575, Mar. 2015.

[22] X. Liu, M. Shahidehpour, Z. Li, X. Liu, Y. Cao, and Z. Li, "Power system risk assessment in cyber attacks considering the role of protection systems," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 572–580, Mar. 2017.

[23] S. Kriaa, M. Bouissou, and L. Piètre-Cambacédès, "Modeling the stuxnet attack with BDMP: Towards more formal risk assessments," in *Proc. 7th Int. Conf. Risks Secur. Internet Syst. (CRiSIS)*, Oct. 2012, pp. 1–8.

[24] P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Man-in-the-middle attacks and defense in a power system cyber-physical testbed," 2021, *arXiv:2102.11455*. [Online]. Available: http://arxiv.org/abs/2102.11455

[25] D. Hall, *Mathematical Techniques in Multisensor Data Fusion*. Norwood, MA, USA: Artech House, Jan. 1992.

[26] P. Roengruen, V. Tipsuwannaporn, A. Numsomran, and S. Harnnarong, "Evaporative estimation using data fusion," in *Proc. SICE Annu. Conf.*, Aug. 2008, pp. 1692–1697.

[27] P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy, "Using Bayesian networks for cyber security analysis," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2010, pp. 211–220.

[28] L. A. Zadeh, "A simple view of the Dempster–Shafer theory of evidence and its implication for the rule of combination," *AI Mag.*, vol. 7, no. 2, pp. 85–90, Jul. 1986.

[29] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Inf. Fusion*, vol. 14, no. 1, pp. 28–44, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253511000558

[30] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. IEEE*, vol. 85, no. 1, pp. 6–23, Jan. 1997.

[31] A. Miloslavov and M. Veeraraghavan, "Sensor data fusion algorithms for vehicular cyber-physical systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1762–1774, Sep. 2012.

[32] D. Jiang, D. Zhuang, Y. Huang, and J. Fu, "Advances in multi-sensor data fusion: Algorithms and applications," *Sensors*, vol. 9, pp. 7771–7784, Sep. 2009.

[33] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, and S. Zonouz, "A multi-sensor energy theft detection framework for advanced metering infrastructures," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1319–1330, Jul. 2013.

[34] P. Wang, L. T. Yang, J. Li, J. Chen, and S. Hu, "Data fusion in cyber-physical-social systems: State-of-the-art and perspectives," *Inf. Fusion*, vol. 51, pp. 42–57, Nov. 2019.

[35] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE Trans. Big Data*, vol. 1, no. 1, pp. 16–34, Mar. 2015.

[36] H. Nguyen, K. Palani, and D. Nicol, "An approach to incorporating uncertainty in network security analysis," in *Proc. Hot Topics Sci. Secur., Symp. Bootcamp*, Apr. 2017, pp. 74–84.

[37] F. Fusco, S. Tirupathi, and R. Gormally, "Power systems data fusion based on belief propagation," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Sep. 2017, pp. 1–6.

[38] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.

[39] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 1395–1402.

[40] M. Esmalifalak, H. Nguyen, R. Zheng, and Z. Han, "Stealth false data injection using independent component analysis in smart grid," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2011, pp. 244–248.

[41] M. Mohammadpourfard, A. Sami, and A. R. Seifi, "A statistical unsupervised method against false data injection attacks: A visualization-based approach," *Expert Syst. Appl.*, vol. 84, pp. 242–261, Oct. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417417303317

[42] D. Wilson, Y. Tang, J. Yan, and Z. Lu, "Deep learning-aided cyber-attack detection in power transmission systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2018, pp. 1–5.

[43] J. J. Q. Yu, Y. Hou, and V. O. K. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3271–3280, Jul. 2018.

[44] S. Basumallik, R. Ma, and S. Eftekharnejad, "Packet-data anomaly detection in PMU-based state estimator using convolutional neural network," *Int. J. Electr. Power Energy Syst.*, vol. 107, pp. 690–702, May 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061518319884

[45] A. Kundu, A. Sahu, K. Davis, and E. Serpedin, "Learning-based defense of false data injection attacks in power system state estimation," in *Proc. North Amer. Power Symp. (NAPS)*, Oct. 2019, pp. 1–6.

[46] A. Kundu, A. Sahu, E. Serpedin, and K. Davis, "A3D: Attention-based auto-encoder anomaly detector for false data injection attacks," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106795. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779620305988

[47] M. Culler, K. Davis, and A. Sahu, "PAVED: Perturbation analysis for verification of energy data," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2019, pp. 1–6.

[48] A. Yang, X. Wang, Y. Sun, Y. Hu, Z. Shi, and L. Sun, "Multi-dimensional data fusion intrusion detection for stealthy attacks on industrial control systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[49] L. Wehenkel, "Machine learning approaches to power-system security assessment," *IEEE Expert*, vol. 12, no. 5, pp. 60–72, Sep./Oct. 1997.

[50] T. Rice, G. Seppala, T. W. Edgar, D. Cain, and E. Choi, "Fused sensor analysis and advanced control of industrial field devices for security: Cymbiote multi-source sensor fusion platform," in *Proc. Northwest Cybersecur. Symp.*, Apr. 2019, pp. 1–8.

[51] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory (COLT)*, Oct. 1998, pp. 92–100.

[52] T. S. Abdelgayed, W. G. Morsi, and T. S. Sidhu, "Fault detection and classification based on co-training of semisupervised machine learning," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1595–1605, Feb. 2018.

[53] H. He, X. Luo, F. Ma, C. Che, and J. Wang, "Network traffic classification based on ensemble learning and co-training," *Sci. China F, Inf. Sci.*, vol. 52, no. 2, pp. 338–346, Feb. 2009.

[54] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jul. 2011.

[55] N. Chen, J. Zhu, and E. P. Xing, "Predictive subspace learning for multi-view data: A large margin approach," in *Advances in Neural Information Processing Systems*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2010, pp. 361–369.

[56] Z. Fang and Z. Zhang, "Discriminative feature selection for multi-view cross-domain learning," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 1321–1330.

[57] P. Yang and W. Gao, "Multi-view discriminant transfer learning," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Aug. 2013, pp. 1848–1854.

[58] E. Vaahedi, A. Y. Chang, S. Mokhtari, N. Muller, and G. Irisarri, "A future application environment for BC Hydro's EMS," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 9–14, Feb. 2001.

[59] *The MongoDB 4.4 Manual*. Accessed: Jul. 2020. [Online]. Available: https://docs.mongodb.com/manual/

[60] N. Padalia, *Apache Cassandra Essentials*. Birmingham, U.K.: Packt Publishing, 2015.

[61] *KDD Cup 1999 Data*. Accessed: Dec. 2020. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[62] *Coburg Intrusion Detection DataSets*. Accessed: Feb. 2021. [Online]. Available: https://github.com/markusring/CIDDS

[63] A. Sahu, Z. Mao, K. Davis, and A. E. Goulart, "Data processing and model selection for machine learning-based network intrusion detection," in *Proc. IEEE Int. Workshop Tech. Committee Commun. Qual. Rel. (CQR)*, May 2020, pp. 1–6.

[64] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[65] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—An open-source Python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.

[66] U. Adhikari, T. Morris, and S. Pan, "WAMS cyber-physical test bed for power system, cybersecurity study, and data mining," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2744–2753, Nov. 2017.

[67] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.

[68] S. Pan, T. H. Morris, and U. Adhikari, "A specification-based intrusion detection framework for cyber-physical environment in electric power system," *IJ Netw. Secur.*, vol. 17, no. 2, pp. 174–188, Jan. 2015.

[69] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner, "An evaluation of machine learning methods to detect malicious SCADA communications," in *Proc. 12th Int. Conf. Mach. Learn. Appl. (ICMLA)*, vol. 2, Dec. 2013, pp. 54–59.

[70] P. M. Laso, D. Brosset, and J. Puentes, "Dataset of anomalies and malicious acts in a cyber-physical subsystem," *Data Brief*, vol. 14, pp. 186–191, Oct. 2017.

[71] Glover, T. Overbye, and Sarma. *Power-World Simulator*. Accessed: Jun. 2019. [Online]. Available: https://www.powerworld.com/products/simulator/overview

[72] P. Wlazlo, K. Price, C. Veloz, A. Sahu, H. Huang, A. Goulart, K. Davis, and S. Zounouz, "A cyber topology model for the Texas 2000 synthetic electric power grid," in *Proc. Princ., Syst. Appl. IP Telecommun. (IPTComm)*, Oct. 2019, pp. 1–8.

[73] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 8, pp. 1–14, Aug. 2018.

[74] K. Nigam and R. Ghani, "Understanding the behavior of co-training," in *Proc. KDD-Workshop Text Mining*, Aug. 2000, pp. 15–17.

[75] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[76] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proc. Roy. Soc. London*, vol. 58, nos. 347–352, pp. 240–242, 1895.

[77] *Manifold Learning*. Accessed: Nov. 2020. [Online]. Available: https://scikit-learn.org/stable/modules/manifold.html

[78] L. Saul and S. Roweis, "An introduction to locally linear embedding," *J. Mach. Learn. Res.*, vol. 7, pp. 1–13, Jan. 2001.

[79] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[80] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 796–809, May 2008.

[81] U. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, pp. 395–416, Jan. 2004.

[82] *Clustering*. Accessed: Oct. 2020. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html

[83] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, pp. 103–114, Jun. 1996.

[84] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *VLDB J.*, vol. 16, no. 4, pp. 507–521, Oct. 2007.

[85] S. A. Mulay, P. R. Devale, and G. V. Garje, "Intrusion detection system using support vector machine and decision tree," *Int. J. Comput. Appl.*, vol. 3, no. 3, pp. 40–43, Jun. 2010.

[86] Y. Wang, "A multinomial logistic regression modeling approach for anomaly intrusion detection," *Comput. Secur.*, vol. 24, no. 8, pp. 662–674, Nov. 2005.

[87] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technol.*, vol. 4, pp. 119–128, Dec. 2012.

[88] A. Sahu, H. N. R. K. Tippanaboyana, L. Hefton, and A. Goulart, "Detection of rogue nodes in AMI networks," in *Proc. 19th Int. Conf. Intell. Syst. Appl. Power Syst. (ISAP)*, Sep. 2017, pp. 1–6.

[89] T. Abbes, A. Bouhoula, and M. Rusinowitch, "Protocol analysis in intrusion detection using decision tree," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, vol. 1, 2004, pp. 404–408.

[90] N. Moustafa, B. Turnbull, and K.-K.-R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.

[91] N. Farnaaz and M. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Comput. Sci.*, vol. 89, pp. 213–217, Jan. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916311127

[92] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 649–659, Sep. 2008.

[93] M. R. Narimani, H. Huang, A. Umunnakwe, Z. Mao, A. Sahu, S. Zonouz, and K. Davis, "Generalized contingency analysis based on graph theory and line outage distribution factor," Jul. 2020, *arXiv:2007.07009*. [Online]. Available: http://arxiv.org/abs/2007.07009

[94] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat., Theory Methods*, vol. 3, no. 1, pp. 1–27, Jan. 1974.

[95] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[96] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. D. F. Costa, and F. A. Rodrigues, "Clustering algorithms: A comparative approach," *PLoS ONE*, vol. 14, no. 1, pp. 1–34, Jan. 2019, doi: 10.1371/journal.pone.0210236.

[97] K. Sequeira and M. Zaki, "ADMIT: Anomaly-based data mining for intrusions," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: Association for Computing Machinery, 2002, pp. 386–395, doi: 10.1145/775047.775103.

[98] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[99] A. Sahu, Z. Mao, P. Wlazlo, H. Huang, K. Davis, A. Goulart, and S. Zonouz, "Cyber-physical dataset for mitm attacks in power systems," IEEE Dataport, 2021, doi: 10.21227/e4dd-2163.

[100] A. Sahu and Z. Mao. (2021). *Multi-Source Data Fusion for Cyber Intrusions in Power Systems*. [Online]. Available: https://codeocean.com/capsule/3327036/tree

ABHIJEET SAHU received the B.S. degree in electronics and communications from the National Institute of Technology, Rourkela, India, in 2011, and the M.S. degree in electrical and computer engineering from Texas A&M University, TX, USA, in 2018, where he is currently pursuing the Ph.D. degree. He is also working with Dr. Katherine Davis and Dr. Ana Goulart on the Cyber-Physical Resilient Energy Systems Project with Texas A&M University. He worked as a Networking Engineer with National Thermal Power Corporation Ltd., India, from 2011 to 2015. During his graduate studies, he has interned at Real Time Power Inc., Houston, TX, USA. His research interests include network security, cyber-physical modeling for intrusion detection and response, and artificial intelligence for cyber-physical security in power systems.

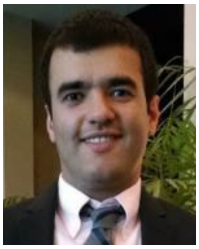
ZEYU MAO (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Chongqing University, Chongqing, China, in 2015, and the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, IL, USA, in 2017. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Texas A&M University, TX, USA. His research interests include power system cyber-physical modeling, data-driven power system control, and sparse matrix ordering.

PATRICK WLAZLO received the B.S. degree in electrical engineering systems technology from Texas A&M University, in 2020, where he is currently pursuing the M.S. degree in engineering technology. He is also a Graduate Research Assistant with Texas A&M University, working on the Cyber Physical Resilient Energy Systems Project. His current research interests include cybersecurity as it relates to electrical transmission and generation.

HAO HUANG (Member, IEEE) received the B.S. degree in electrical engineering from Harbin Institute of Technology, China, in 2014, with a focus on power system and its automation, and the M.S. degree in electrical engineering from the University of Southern California, in 2016, with a focus on electric power. He is currently pursuing the Ph.D. degree in electrical engineering with Texas A&M University, under the supervision of Prof. Katherine Davis. His research interests include power system resilience, power system situational awareness, and cyber-physical security.

KATHERINE DAVIS (Senior Member, IEEE) received the B.S. degree from The University of Texas at Austin, Austin, TX, USA, in 2007, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2009 and 2011, respectively, all in electrical engineering. She is currently an Assistant Professor of electrical and computer engineering with Texas A&M University.

ANA GOULART received the bachelor's degree in electrical engineering from the Federal School of Engineering of Itajuba (EFEI), Brazil, the M.Sc. degree in information systems management from the Pontifical Catholic University of Campinas, the M.Sc. degree in computer engineering from North Carolina State University, Raleigh, NC, USA, and the Ph.D. degree in electrical and computer engineering from Georgia Tech, Atlanta, GA, USA, in 2005. She is currently an Associate Professor with the Electronics Systems Engineering Technology Program, Texas A&M University, College Station, TX, USA. Her research interests include protocols for real-time communications, IP-based emergency communications, last-mile communication links and cybersecurity for the smart grid, and rural telecommunications.



SAMAN ZONOUZ received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign. He is currently an Associate Professor with the Electrical and Computer Engineering Department, Rutgers University. His research was awarded by the Presidential Early Career Awards for Scientists and Engineers (PECASE) by the United States President, in 2019; NSF CAREER Award, in 2015; and the National Security Agency (NSA) Significant Research in Cyber Security, in 2015.