Hindawi
Security and Communication Networks
Volume 2017, Article ID 9150965, 17 pages
https://doi.org/10.1155/2017/9150965

Review Article

Towards Large-Scale, Heterogeneous Anomaly Detection Systems in Industrial Networks: A Survey of Current Trends

Mikel Iturbe, Iñaki Garitano, Urko Zurutuza, and Roberto Uribeetxeberria

Department of Electronics and Computing, Mondragon Unibertsitatea, Goiru 2, 20500 Arrasate-Mondragón, Spain

Correspondence should be addressed to Mikel Iturbe; miturbe@mondragon.edu

Received 13 September 2017; Accepted 5 November 2017; Published 22 November 2017

Academic Editor: Javier Lopez

Industrial Networks (INs) are widespread environments where heterogeneous devices collaborate to control and monitor physical processes. Some of the controlled processes belong to Critical Infrastructures (CIs), and, as such, IN protection is an active research field. Among different types of security solutions, IN Anomaly Detection Systems (ADSs) have received wide attention from the scientific community. While INs have grown in size and in complexity, requiring the development of novel, Big Data solutions for data processing, IN ADSs have not evolved at the same pace. In parallel, the development of Big Data frameworks such as Hadoop or Spark has led the way for applying Big Data Analytics to the field of cyber-security, mainly focusing on the Information Technology (IT) domain. However, due to the particularities of INs, it is not feasible to directly apply IT security mechanisms in INs, as IN ADSs face unique characteristics. In this work we introduce three main contributions. First, we survey the area of Big Data ADSs that could be applicable to INs and compare the surveyed works. Second, we develop a novel taxonomy to classify existing IN-based ADSs. And, finally, we present a discussion of open problems in the field of Big Data ADSs for INs that can lead to further development.

1. Introduction

Industrial Networks (INs) refer to the networked environments where specialized, heterogeneous interconnected components, known collectively as Industrial Control Systems (ICSs), automate, control, and monitor physical processes. As such, they are responsible for running a wide range of physical processes, in different industrial sectors and in Critical Infrastructures (CIs) [1]. The European Council [2] defines a CI as “an asset, system or part thereof (...) which is essential for the maintenance of vital societal functions, health, safety, security, economic or social well-being of people, and the disruption or destruction of which would have a significant impact (...) as a result of the failure to maintain those functions.” Examples of CIs include power generation and transport, water distribution, water waste treatment, and transportation systems.

Therefore, the correct functioning of CIs has a vital importance. Miller and Rowe [3] surveyed previous security incidents that affected CIs. Nowadays, there are two main specific concerns about the impact of IN related attacks:

(1) Successful attacks against INs may have an impact on the physical process the IN is monitoring, potentially leading to safety-threatening scenarios. Examples of such incidents include Aurora [4], Stuxnet [5], the Maroochy water breach [6], and the German steel mill incident [7].

(2) The proliferation of ICS-specific malware for conducting espionage: the aim of these pieces of malware is to gather information about the controlled process and/or company running it. The purpose can be twofold: stealing confidential information about the process (e.g., recipe for manufacturing a product) or to gather information to conduct attacks against a third party. Examples of such malware include Duqu [8] and Dragonfly [9].

As a consequence, the critical or confidential nature of some of the controlled processes and the potential impact of service malfunction, IN security is an active research field. As such, IN protection has received wide attention from both industry and the scientific community. Among the different fields of

IN security research, Intrusion Detection Systems (IDSs) and, particularly, Anomaly Detection Systems (ADSs) have an important role and there are many proposals in this direction [10–12].

Alternatively, since the birth of distributed computing frameworks such as MapReduce [13] and distributed filesystems such as the Hadoop File System (HDFS) [14], a new computing paradigm known as *Big Data Analytics* (BDA) has emerged. Big Data refers to the set of information that is too complex to process by traditional IT mechanisms within an acceptable scope [15]. Although no total consensus exists, this data complexity is generally expressed in at least three qualities: the amount of data (volume), data generation and transmission pace (velocity), and diversity of data, both structured and unstructured (variety) [16]. More recently, a fourth quality is also widely mentioned: the ability to search for valuable information on Big Data (veracity) [15]. However, the term Big Data has transcended the type of information and it is also used to refer to set of methodologies and mechanisms developed to work with this type of data. BDA aims to extract valuable knowledge from Big Data by analyzing or modeling it in a scalable manner.

Among the multiple applications BDA has, Cárdenas et al. [19] and Everett [20] discussed its potential for intrusion detection research. They conclude that using BDA can lead to more efficient IDSs. However, both works center on regular Information Technology (IT) networks and do not examine its applicability to INs. In this work, we analyze different existing Big Data ADSs that can be used in INs and extract insight from them in order to identify some possible future research areas.

Our contributions can be summarized as follows:

(i) A literature review of Big Data ADSs that can be applied to INs.

(ii) A novel taxonomy to classify IN-based ADSs.

(iii) A discussion of open problems in existing IN-oriented, large-scale, heterogeneous ADS research.

The rest of the paper is organized as follows. Section 2 introduces INs, ADSs, and Big Data Security Mechanisms. Section 3 presents the taxonomy used for ADS classification. Section 4 analyzes the most relevant proposals applicable to IN intrusion detection. Section 5 discusses the proposals and evaluates their suitability for their usage in INs. Section 6 points to some open research areas that have not been covered by previous approaches. Finally, Section 7 draws the final conclusions.

2. Background

In this section we provide the necessary background to support our argumentation.

2.1. Industrial Networks. Since the invention of the Programmable Logic Controller (PLC) in the 1960s, INs have evolved significantly from the initial primitive proprietary and isolated environments to the complex, standard interconnected networks that are today. Traditionally, INs were isolated environments where communication was conducted through proprietary network protocols with limited or nonexistent interaction with external networks. However, since the 1990s, pushed by the increasing demand for location-independent access to network resources, INs became progressively interconnected with external networks such as the companies' internal IT network and even the Internet [21, 43].

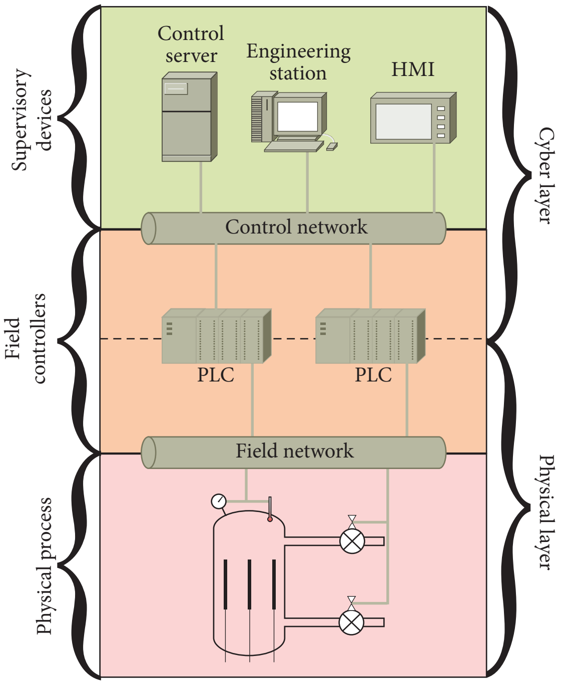

Figure 1: Example of a simple industrial network.

On the one hand, this increased network standardization led to the start of using standard network protocols (TCP/IP) and commercial-off-the-shelf (COTS) software, laying behind proprietary, ad hoc hardware and software solutions [1]. On the other hand, this merge significantly increased the attack surface of INs, as it exposed them to simple remote attacks and exploitation by using known vulnerabilities of COTS software. Traditional isolation and obscure characteristics that INs had relied on for security did no longer exist.

Figure 1 shows the network architecture of a simple IN. INs have a vertical architecture. At the bottom lays the physical process that is being controlled. The physical process has a set of sensors and actuators that are used to gather information about the state of the process and to perform actions on it. These sensors and actuators are connected to field controllers, normally PLCs, through buses or direct connections in the so-called field network. Field controllers are the workhorse of INs. They read process data from the field sensors and, based on their stored control algorithm, send orders to the actuators to interact with the process, generally trying to keep process variables' values around a set of certain setpoints. Nevertheless, except for the simplest installations, field controllers are not enough to conduct all the required tasks. Consequently, additional devices, called

Table 1: Differences between industrial and IT networks [21, 22].

	Industrial networks	IT networks
Primary function	Control of physical equipment	Data processing and transfer
Applicable domain	Manufacturing, processing and utility distribution	Corporate and home environments
Hierarchy	Deep, functionally separated hierarchies with many protocols and physical standards	Shallow, integrated hierarchies with uniform protocol and physical standard utilization
Failure severity	High	Low
Reliability required	High	Moderate
Round trip times	250 μs–10 ms	50+ ms
Determinism	High	Low
Data composition	Small packets of periodic and aperiodic traffic	Large, aperiodic packets
Temporal consistency	Required	Not required
Operating environment	Hostile conditions, often featuring high levels of dust, heat and vibration	Clean environments, often specifically intended for sensitive equipment
System lifetime	Some tens of years	Some years
Average node complexity	Low (simple devices, sensors, actuators)	High (large servers/file systems/databases)
Primary security requirement	Availability	Confidentiality

supervisory devices, are necessary. These devices usually run on normal IT-based hardware and software. Examples include control servers, Human Machine Interfaces (HMIs), and engineering stations. Control servers store process data and, optionally, implement second-level control logic, usually involving data from different field controllers. HMIs are the graphical user interfaces operators used to interact with the process. Critical processes are monitored by human operators 24/7. Process engineers use engineering stations to develop and test new applications regarding control logic.

INs can be further divided according to different layers. According to the definition by Genge et al. [44], on the one hand, there is the physical layer, composed of the actuators and sensors that directly interact with the physical process. On the other hand, there is the cyber layer, composed of all the IT devices and software which acquire the data, elaborate low level process strategies, and deliver the commands to the physical layer. Field controllers act as the bridge between both layers, as they read field data and send local commands to the actuators, but they also forward field information to the cyber layer components while executing commands they receive from the supervisory devices.

Hence, ICSs can be considered a subset of Cyber Physical Systems, as they are able to process and communicate data while also interacting with their physical environment.

There are different types of INs, such as Supervisory Control and Data Acquisition (SCADA), Distributed Control Systems (DCSs), and Process Control Systems (PCS). However, differences are getting blurred, and they can often be considered as a single entity when designing security solutions [1, 22].

Although they share a common part of technology stack, INs are inherently different to commercial IT networks. Table 1 shows a summary of the main differences between both network types. The main difference resides in the purpose of each of the networks: whereas, in IT, the purpose is the transfer and processing of data, in the case of INs the main objective is to control a physical process.

Additionally, security requirements in IT networks and INs differ in importance. There are three main security requirements that information systems or networks must fulfil in order to be considered secure: confidentiality, integrity, and availability [45, 46]. Dzung et al. [47] describe the requirements and relate them to INs.

Confidentiality. Prevention of information disclosure to unauthorized persons or systems: in the case of INs, this is relevant with respect to both domain specific information, such as product recipes or plant performance and planning data, and the secrets specific to the security mechanisms themselves, such as passwords and encryption keys.

Integrity. Prevention of undetected modification of information by unauthorized persons or systems: in INs, this applies to information such as product recipes, sensor values, or control commands. Violation of integrity may cause safety issues; that is, equipment or people may be harmed.

Availability. It refers to ensuring that unauthorized persons or systems cannot deny access or use to authorized users. In INs, it refers to all the devices of the plant, like control systems, safety systems, operator workstations, and so on, as well as the communication systems between these elements and to the outside world. Violation of availability, also known as Denial of Service (DoS), may not only cause economic damage but may also affect safety issues as operators may lose the ability to monitor and control the process.

On the one hand, IT networks, which are designed to store and transmit information, lean to keep the data confidential and information integrity and availability play a lesser role. On the other hand, in INs, availability is paramount, as losing control of a process or disrupting it can cause significant economic losses and, in the case of specially safety-critical INs, such as CIs, the consequences can be significantly more severe and potentially catastrophic [3].

These requirement differences mean that even when technically possible, blindly applying IT-based security mechanisms or procedures in industrial environments might lead to process malfunction or potentially safety-threatening scenarios, as they have been designed with different goals in mind. For instance, running antivirus software on PLCs might compromise the PLC's ability to perform real-time operations on a process, or conducting a penetration test can lead to dangerous scenarios [48].

However, these traits can also be leveraged to build security mechanisms for INs that would be impractical to use in IT networks. For instance, the deterministic nature of INs and its periodic traffic between different hosts makes them suitable candidates for using Anomaly Detection Systems [10].

2.2. Anomaly Detection Systems.

Anomaly Detection Systems (ADSs) are a subset of Intrusion Detection Systems (IDSs) [49]. IDSs are security mechanisms that monitor network and/or system activities to detect suspicious events. IDSs are classified according to two main criteria: the detection mechanism they use (signature detection or anomaly detection) and their source of information (where they collect the events to analyze).

Signature-based IDSs compare monitored data to a database with known malicious patterns (signature database). If there is a match, an alert is raised, as the activity has been identified as suspicious. Their efficiency is directly related to the completeness and accuracy of the signature database they are working with, as attacks will go undetected if their signature is not available. Among their operational characteristics, they have a low number of false positives but they are unable to detect unknown attacks. ADSs, on the other hand, identify malicious patterns by measuring their deviation from normal activity. ADSs build a model of the normal behavior of the process (through automated learning or manual specifications) and detect deviations with respect to the model [21]. Many ADSs are built using machine learning methods [50]. As opposed to signature-based IDSs, ADSs are able to detect unknown attacks, but they often yield a higher number of false positives.

Regarding the source of information, IDSs traditionally have been classified into two main categories: network-level and host-level IDSs. Network-based IDSs monitor network traffic to detect suspicious activity (suspicious connections, malicious packet payloads, etc.), while host-based IDSs monitor local data stored in a device (system logs, file integrity, etc.). In the case of INs, the limited processing ability of industrial devices has limited the deployment of host-based ICSs [21]. Therefore, when considering IN IDSs, the source of information criterion can be set based on the IN layer they use to gather information from the cyber-level or the physical layer. Cyber-level IDSs are similar to their IT counterparts as they generally monitor network-level data. Physical-level IDSs monitor the physical quantities of the process (pressures, temperatures, currents, etc.) in order to detect intrusions. Physical properties of the process are constantly monitored, often polling data every few milliseconds in the case of critical variables, which with large, continuous processes can lead to a scenario where it is necessary to use Big Data Analytics (BDA), covered in Section 1, in order to process field and control data. This is further confirmed by proposals that, outside the field of security research, point to this need and propose several BDA solutions focused on industrial applications, such as process monitoring [51–54], maintenance [55], fault detection [56], and fault diagnosis [57, 58].

Most IN ADSs work on the cyber layer (see surveys [10–12]). Physical-level ADSs can be divided into two main groups: ADSs where it is necessary to model the physical process [59, 60] or ADSs that do not need a specific model for the physical process [61, 62]. Few proposals combine data from both levels [63, 64].

2.3. Big Data Security Mechanisms.

Modern and complex IT networks create and process vast amounts of data continuously. Analysis of the created data for security purposes is a daunting task, and, before the advent of Big Data processing tools, data was normally sampled or only subsets of it were analyzed (e.g., only metadata). Since MapReduce [13] was introduced, several Big Data frameworks have been proposed, which allow the processing of large, heterogeneous datasets.

Traditionally, Big Data frameworks have been divided into two main groups, according to the nature of the data they work with. On the one hand, there are batch processing technologies that work with data at rest and are usually used when doing Exploratory Data Analysis (EDA). Examples of technologies that use this approach would include Hadoop [65] and Disco [66]. On the other hand, there are stream processing technologies that are designed to work with flowing data. Gorawski et al. [67] reviewed different Big Data streaming proposals.

However, hybrid tools such as Apache Spark [68] or Apache Flink [69] are able to work on both streaming and resting data. Spark uses microbatches to process incoming data while Flink does batch processing as a special case of stream processing.

Extracting insight from the large amount of information that could be leveraged for security event detection (e.g., logs, network flows, or packets) in a network can be considered a Big Data problem [19, 20]. Consequently, different types of Big Data Security Mechanisms have been proposed:

(i) Intrusion detection (see survey [70]).

(ii) Botnet detection ([71–74]).

(iii) Malware detection ([75–78]) and analysis ([79, 80]).

(iv) Distributed Denial of Service (DDoS) detection ([81–85]).

(v) Spam detection ([86–88]).

On a related note, other resources have been developed that even if they are not security mechanisms per se, they have been designed to handle large volumes of network data and,

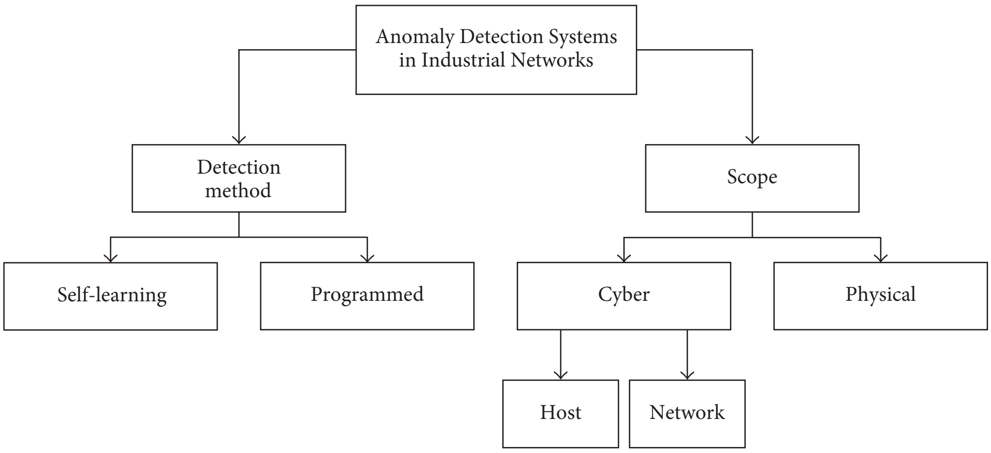

Figure 2: A taxonomy for Anomaly Detection Systems in Industrial Networks.

thus, can be useful in building security mechanisms on top of them:

(i) Frameworks for analyzing network flows ([89, 90]).

(ii) Frameworks for analyzing network packets ([91, 92]).

(iii) Frameworks for analyzing logs ([93, 94]).

However, in this paper we will limit the scope to Big Data ADSs that could potentially be applicable to the industrial domain.

3. Taxonomy

In this section, we describe the taxonomy or classification method that will be used in Section 4 for existing large-scale industrial ADSs. Figure 2 shows the created taxonomy tree. When classifying IDSs in general, two main criteria are used: the detection method and the scope of the IDS [10, 21, 49, 95]. We can apply these criteria to build an IN ADS classification method.

3.1. Detection Method. The main criterion to classify IDSs resides in the detection method. While the difference between signature-based IDSs and ADSs was already covered in Section 2.2, ADSs can be further classified based on their detection technique. According to Axelsson [95] and Mitchell and Chen [10] ADS detection techniques belong in two categories:

(1) Self-learning or behavior-based ADSs: the ADS detects anomalous features that are distinct from normal system behavior. Normal system behavior can be retrieved in an unsupervised (e.g., clustering historical data) or in a semisupervised manner (e.g., collection of training, generally attack-free, data).

(2) Programmed or behavior-specification-based ADSs: using expert knowledge, a human defines legitimate behaviors and implements them on the ADS. The ADS detects anomalies by detecting deviations from the specified behavior.

3.2. Scope. Apart from the detection method, the other main criterion for IDS and ADSs is their scope, that is, the source and nature of the data used for audit. In IT ADSs, there are two main types of ADSs depending on the data they use.

(1) Network ADSs: ADSs monitor a network without focusing on individual hosts. The most prominent data sources for these ADSs are network flows and packets.

(2) Host ADSs: the ADS monitors data from an individual host to check anomalies. Examples of host data include logs, files, or system calls.

While this split was conceived for IT-based ADSs, this classification has also held for IN ADSs [10, 21, 96]. And indeed, most IN ADS proposals can be classified in one of the two above categories. Nevertheless, due to the cyber physical nature of INs, this classification is not complete enough, as it only tackles the cyber part of INs, while not considering the physical dimension of INs that handles field data. Field data mainly consists of sensor signals that monitor physical quantities (temperature, pressure, etc.) although other process-based variables (counters, setpoint values, etc.) might be present. There are several examples of IN ADSs that leverage field-level data [59–62]. This data can come from logs on a control server, direct process measurements, and simulated data or can be scattered across different hosts or devices. Therefore, ADS proposes that leverage process data for anomaly detection do not fit well in the above classification. Consequently, we have created a novel taxonomy where the physical dimension of IN ADSs is taken into account as a proper data source. This taxonomy can be leveraged to classify IN ADSs, both conventional and Big Data proposals, as it encompasses more data sources and types that are present in INs than previous presented taxonomies that do not acknowledge the existence of ADSs based on the physical layer of INs.

4. Anomaly Detection Systems

In this section, we survey existing Big Data ADSs that could be used in INs. Proposals are divided according to the taxonomy described in Section 3.

4.1. Cyber-Level ADSs

4.1.1. Cyber-Level, Self-Learning ADSs. The proposal of Xu et al. [42] is an ADS based on host log mining. System logs are first parsed to provide a unified data-structure from different log formats, by getting log format templates from the application source code. Then, they build features from the extracted log data, focusing on state ratio vector (a vector representing a set of state variables on a time window) and the message count vector (a vector representing a set of related logs with different message types) features. These vector features are later mined using an algorithm based on Principal Component Analysis (PCA) for anomaly detection. The results are finally visualized in a decision tree to aid operators to find the root cause of an anomaly. The analysis is carried out in a Hadoop cluster to increase computing speed.

Yen et al. [23] introduce Beehive, a large scale log mining tool that uses Hive [97] to detect suspicious activities in corporate networks. For that purpose, Beehive mines logs coming from different sources and, specially, web proxy logs. Beehive clusters log data and identifies misbehaving hosts as cluster outliers, as they show a unique behavioral pattern. The clustering is done by an adapted version of the k-means algorithm. The incidents related to the outliers were labeled manually by using other system logs and showed that many of these outliers were not detected by traditional security mechanisms.

Ratner and Kelly [38] conduct a case study of network traffic anomalies in a corporate network. For this end, they extract packet metadata from a set of captured packets, and they perform specific queries in the gathered data to detect attacks, mainly IP scans. In order to process the large dataset, they use Apache Hadoop. They find a large number of IP scans and conclude that roughly half of the packets arriving from external IP addresses are anomalous. Those anomalies were found by comparing each packet's IP metadata to the average values for each day.

Therdphapiyanak and Piromsopa [98] expose an anomaly detection system based on host log analysis. First, the system parses log data and later clusters it by using k-means. Once the clusters are formed, the authors extract major characteristics from the clusters to examine differences and similarities. Minor clusters with important differences when compared to others are flagged as anomalous. While the system has been tested with Apache Web Server logs, the authors address aggregating logs from different network agents in future steps. Log parsing and clustering are performed in a Hadoop cluster.

Camacho et al. [17] use a PCA-based solution to detect anomalies in computer networks. The workflow of the approach can be seen in Figure 3. The anomaly detection is accomplished in two separate phases: a model building phase, where the ADS is tuned based on training data, and a monitoring phase, where the ADS analyzes incoming data and determines it as anomalous or legitimate based on the model built during the previous phase. In the first phase, incoming data (generally, IDS and Firewall logs) is preprocessed and converted into feature vectors. Later, this data is used to create a PCA model, where the original features are transformed into a new variable subspace. The dimensionality reduction helps to visually inspect the data and to identify outliers in it. These outliers can be considered as anomalies or attacks in training data, so this preliminary step cleans the data and allows building a more effective model, without requiring a completely attack-free initial dataset. The PCA model can also be used to create two different statistics that are widely used for process monitoring: Hotelling's T^2 [99], comprising the leverages of the PCA model, and the *SPE* [100], involving the residuals of the model. The proposed approach calculates the statistics for each of the observations in the training set and, based on it, calculates a control limit based on an arbitrary confidence level, where a given percentage of the training observations should be below the control limit. Once the control limits have been set, the training phase has ended and the ADS is now prepared to work in the monitoring phase. In this phase, incoming data is preprocessed and transformed using

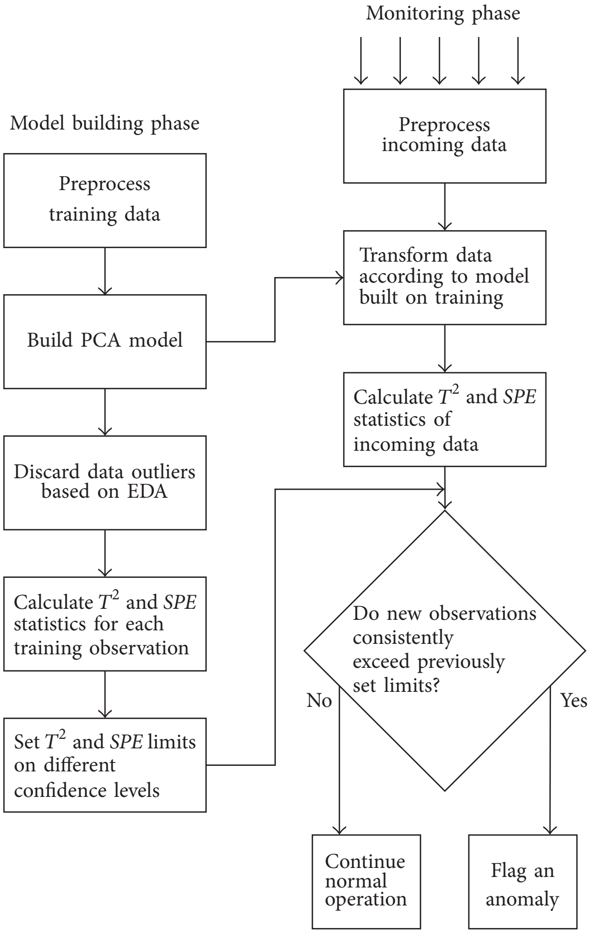
Monitoring phase
Model building phase
Preprocess training data
Build PCA model
Discard data outliers based on EDA
Calculate T^2 and *SPE* statistics for each training observation
Set T^2 and *SPE* limits on different confidence levels
Preprocess incoming data
Transform data according to model built on training
Calculate T^2 and *SPE* statistics of incoming data
Do new observations consistently exceed previously set limits?
No
Yes
Continue normal operation
Flag an anomaly

Figure 3: Anomaly Detection System proposed by Camacho et al. [17], based on Principal Component Analysis (PCA).

the previously created PCA model and it calculates the T^2 and *SPE* values for each incoming observation. If several consecutive observations surpass either of the set control limits (the necessary number of out-of-bounds observations depends on the confidence level), an anomaly is flagged. The process can be parallelized using hierarchical PCA and the workload shared through several slaves. The ability of PCA to work with high dimensional data ensures that the approach can be extended to a wide range of incoming data.

Hashdoop [31] is a MapReduce-based framework designed to run Anomaly Detection Systems in a distributed manner. It does not provide enhanced anomaly detection capacity to the original ADS but it speeds up its execution. First, it splits and hashes network traffic, preserving traffic structures. Later, each of the hashed traffic subsets is analyzed by an instance of the ADS to detect anomalies. Finally, the generated information about the subsets is summarized in a single output report. Results show that processing time is reduced when using Hashdoop-powered ADS compared to their single-node counterparts.

Marchal et al. [35] propose an intrusion detection system that uses honeypot data to detect similar intrusions in networks. First, it collects Domain Name System (DNS) replies, HTTP packets, and IP flow records from the network, along with honeypot data. Based on the collected data, three different scores are computed in order to quantify the maliciousness of the recorded DNS, HTTP, and flow communications. This quantification uses other gathered or publicly available data such as domain blacklists or the data compiled by the in-house honeypot. When one of these maliciousness indices reaches a certain threshold, a flag is raised to inform about the anomaly. The authors test different data-intensive frameworks that are designed to work with potentially very large data volumes. According to their tests, Apache Spark and its subproject, Shark, are faster than Hadoop, Hive, or Pig. However, several concerns arise with this mechanism: the performance of the proposed system is directly related to the performance of the honeypot. If an attacker does not interact with the honeypot and their domain is not explicitly blacklisted, the mechanism will not be able to raise an alert, even in the case of known attacks.

MATATABI [36] is a threat analysis platform that stores data from different sources (DNS captures and querylog, Network flows, and spam email) in a Hadoop cluster and organizes it in Hive [97] tables. Later, different modules query this data via a Javascript Object Notation (JSON) Application Programming Interface (API). Although the exact implementation details of each of the analysis modules are vague, each module queries the stored data looking for anomalous patterns, such as hosts receiving or sending a large number of packets, specific port scans by counting the number of packets to a specific port number, or botnet activity through abnormal DNS activity. While the gathered data is varied, the modules are designed to query a single type of data. If suspicious activity is detected, it is in the operator's hand to query other types of data to find additional evidence of the attack.

TADOOP [40] is a network flow ADS that implements an extension of the Tsallis Entropy [101] for anomaly detection, dubbed DTE-FP (Dual q Tsallis Entropy for flow Feature with Properties). In short, TADOOP gathers network flows and computes a pair of q values aiming to accentuate high and low probability feature distributions, usually linked to traffic anomalies. TADOOP is based on four main modules. (i) The *Traffic Collector* gathers network flow packets and decodes them. (ii) The *Entropy Calculation Module* extracts flow features from each flow and it computes the DTE-FP q values for each flow feature distribution. (iii) The *Semiautomatic Training Module* is responsible for setting optimal q pair detection thresholds for each of the distributions. The criterion is keeping false positive rate below an arbitrary maximal threshold. (iv) The *Detection module* calculates entropy values for all the flows in a given time window and compares them to the thresholds computed by the training module to detect anomalies. TADOOP uses Hadoop for storing and processing historical flow data. TADOOP is evaluated using the flow data of a university network.

Gonçalves et al. [29] present an approach for detecting misbehaving hosts by mining server log data. In the first phase, they extract features from DHCP, authentication, and firewall logs, and for each host a feature vector is created. These vectors are later clustered using the Expectation-Maximization (EM) algorithm which are later used to build a classification model. Smaller clusters in the set correspond to anomalous host behavior. In the second phase, once the classification model is built, incoming data is clustered in a similar way as in the first phase; however, these newly created clusters are classified with the previously created model in order to detect if they are anomalous. While the feature extraction from the log data is done in Hadoop, clustering and classification of the data are carried out with the Weka [102] data mining tool.

Dromard et al. [25] extend the UNADA [103] ADS to detect anomalies in Big Data network environments. UNADA is a three-step unsupervised ADS. (i) *Flow Change Detection*. Flows gathered in a given time window are aggregated on different levels defined by network masks. For each level, UNADA computes a simple metric or feature of the aggregated flows: number of bytes, number of packets, number of IP flows, and so on. Then, when a new set of flows is gathered, these metrics are recomputed for the new flow and compared to the previous set. If there is a change in the values, the time window is flagged and further computed. (ii) *Clustering*. In this phase, UNADA clusters the feature vectors from the previously flagged flow sets using DBSCAN [104]. Network flow feature vectors can have numerous variables and DBSCAN does not perform well in multivariate environments. In order to overcome this issue, UNADA splits the feature space into smaller, two-dimensional subspaces and computes DBSCAN independently on each of them. (iii) *Evidence Accumulation*. In the last phase, data from each of the subspaces is aggregated to identify anomalies. In each subspace, independently, data points that do not belong to a cluster are flagged as anomalous and UNADA records the distance to the nearest cluster centroid. A dissimilarity vector is built with the accumulated abnormality scores for each flow across all subspaces. To ease anomaly detection, dissimilarity vectors are later sorted and a threshold is defined to finally flag flows as anomalous. The authors evaluate the performance of UNADA over Apache Spark [68] to compute the ADS over the network data gathered on a core network of an Internet Service Provider. Results show that the approach is able to detect flow anomalies while speeding up execution time in regard to the original UNADA proposal.

The proposal of Rathore et al. [37] is a flow ADS built on four layers. (i) *Traffic capturing*. The traffic is captured from the network and forwarded to the next layer. (ii) *Filtration and load balancing*. This layer checks whether the flow has been previously registered as a legitimate or anomalous in a database. If it has not, data is forwarded to the next layer. (iii) *Hadoop layer*. This layer extracts the features from the gathered data. It uses Apache Spark [68] on top of Hadoop for faster computation. (iv) *Decision Server*. The extracted features are classified as legitimate or anomalous sets by a set of classifiers implemented in Weka. The authors use the well-known intrusion detection NSL-KDD dataset for result evaluation and conclude that the C4.5 and REPTree are the best performing classifiers for this task.

Wang et al. [41] propose a continuous, real-time flow ADS based on Apache Storm. For this end, they combine three different detection methods: (i) Network flows: They count the number of flows in a small enough time slot that allows online processing. Afterwards, they compute the standard deviation and mean of this count and calculate a confidence interval based on them. Later, they perform a set of operations over the flows involving hashing into groups and calculating Intergroup Flow Entropy [105]. In all steps, the system checks that the observations are inside the confidence interval; otherwise an alarm is raised. (ii) Intuitive Methods based on Traffic Volume: The system applies the same approach as in network flows but taking into account the number of packets in a time window instead the number of flows. (iii) Least-Mean-Square-based detection: The system uses a Least-Mean-Square-based (LMS) filtering method that aims to find inconsistencies between the intergroup flow and packet entropies, which should be strongly correlated. LMS also operates in an online manner. They evaluate the approach by replaying a capture of an Internet backbone while introducing in parallel two types of anomalies that were not present in the capture: an attack involving a large number of small network flows and an attack involving a small number of large flows.

Gupta and Kulariya [28] compare a set of feature extraction and classification algorithms for anomaly detection. They benchmark the different approaches using the popular intrusion detection KDD'99 and NSL-KDD datasets and the algorithms implemented in Spark's MLlib library. They evaluate correlation based feature selection and hypothesis based feature selection for feature extraction. For classification they measure the performance of Naïve Bayes, Logistic Regression, Support Vector Machines, Random Forests, and Gradient Boosted Decision Trees. They conclude that hypothesis based feature selection helps to achieve a better classification score. Among the classifiers, Random Forests and Gradient Boosted Decision Trees yield better results than the rest.

4.1.2. Cyber-Level, Programmed ADSs.

The work presented by Giura and Wang [27] uses large-scale distributed computing to detect APTs. First, they model the APT using an Attack Pyramid, a multiplane extension of an attack tree [106, 107] where the top of the pyramid represents the asset to be protected. The planes of the pyramid represent different environments where attack events can be recorded (e.g., user plane, application plane, and physical plane). The detection method groups all potential security events from different planes and maps the relevant events that are related to a specific attack context. This context information is later leveraged to detect a security incident if some indicators surpass a set of user-defined thresholds. The method uses MapReduce to consider all the possible events and related contexts.

Bumgardner and Marek's approach [24] consists in a hybrid network analysis system that uses both stream and batch processing, capable of detecting some network anomalies. First, it uses a set of probes that collect network traffic to build and send network flows to the specified processing unit. Then, the created flows are stream processed through Storm to enrich it with additional data (e.g., known state of the internal network) and anomalies are detected based on previously defined event detection rules (bot activity, network scans). Once the flows have been processed, they are stored in a HBase table, a column oriented database, to perform EDA to get further insight that it is not explicitly stated in each of the flows. This batch data processing is executed on top of Hadoop. The main drawback of Bumgardner and Marek's approach is that the system's anomaly detection capability is directly related to the capability of describing network events or anomalies using rules when doing stream processing.

Iturbe et al. [33] propose a visual flow monitoring system for INs based on whitelisting and chord diagrams. In their approach, they detect flow-based anomalies (forbidden connections, missing hosts, etc.) based on a previously created set of whitelists. These whitelists can be created through network learning or established by a human operator. The proposed system's scalability is achieved through a distributed search server where data from different networks is sent to store it. Later, a visual application queries the relevant flow data and compares it to the corresponding whitelist. Based on the comparison, a chord diagram is built depicting the legitimate and anomalous flows.

4.2. Physical-Level ADSs.

Hadžiosmanović et al. [30] present a log mining approach to detect process-related threats from legitimate users' unintentional mistakes. They identify unusual events in log data to detect these threats. In order to extract the unusual events from the potentially large log data, they first use a FP-growth algorithm to count matching log entries. Later, unusual events are defined as the ones whose number of occurrences is below of a user-set, absolute threshold. FP-growth algorithms do not use candidate generation and, thus, are able to effectively count occurrences in two data scans.

Difallah et al. [26] propose a scalable ADS for Water Distribution Networks. Specifically, they use Local Indicators of Spatial Association (LISA) [108] as a metric for anomaly detection, by extending the metric to consider temporal associations. In the proposal, wireless sensors send process data to a set of base stations that perform part of the anomaly detection process by computing a limited set of LISA calculations on the streaming data they receive. Thus, it uses a distributed approach for a first phase of anomaly detection. Later, data is sent to a central Array Database Management System (ADBMS). The ADBMS allows global analytics of the distribution network as a whole. Evaluation of the proposal is done using Apache Storm for the stream processing in the base stations and SciDB [109] as the ADBMS, analyzing data from a simulated environment created after the Water Distribution Network of a medium-sized city.

Hurst et al. [32] introduce a Big Data classification system for anomaly detection on CIs. They extract process data from a simulated nuclear power plant and extract relevant features from it, by selecting a number of variables that best describe the overall system behavior. However, this feature extraction relies on expert knowledge to identify the subset of variables that are most suitable. Moreover, the needed features will vary between different types of processes, even different installations, and, thus, the approach is process-dependent. They do not specify the used criteria for feature selection. After feature extraction, they perform anomaly detection using five different classifiers by splitting the gathered data into two halves for the training and testing. They demonstrate that increasing both dataset size and the number of features used for anomaly detection yields better classification results. They do not specify the framework they used for this large-scale classification.

Kiss et al.'s system [34] is designed to detect field-level anomalies in Industrial Networks. By leveraging the field data that sensors and actuators periodically send, they classify normal and abnormal operation cases. To this end, field parameters are used to build feature vectors that are later clustered using k-means to identify operation states and anomalous states of the physical process. In order to deal with the growing field data, the system uses Hadoop to create the different clusters. As the vectors to be clustered are built using field data, these feature vectors depend on process nature. Furthermore, in case of complex physical processes, building the features and identifying different operation states can be a challenging problem that can complicate the deployment of the proposal.

Wallace et al. [18] propose a Smart Grid ADS by mining Phasor Measurement Unit (PMU) data. The overview of their proposal is depicted in Figure 4. The system first models normal grid operation by measuring voltage deviation from each of the PMUs and creating a cumulative probability distribution to represent the likelihood for a signal to have a given voltage deviation. After the distribution function has been created, the likelihood of a given divergence of two voltage signals can be estimated. The system evaluates this calculated likelihood in order to classify an incoming observation as anomalous or legitimate. In detail, the system calculates the voltage deviation from two consecutive signals and then, using the probability distribution function constructed with the historical data, establishes an event as anomalous if this deviation is unlikely to happen. That is, consecutive signals with high discrepancies in voltage values are more unlikely to arise, and, therefore, when they happen they can be classified as anomalous situations in the grid. After an anomaly has been flagged, further analysis of the data can explain the nature of the anomaly. This anomaly identification is carried out by a classification decision tree algorithm that infers the type of anomaly based on three hand-coded events, developed with expert knowledge. The evaluation is done using real PMU data of an electrical grid and using Apache Spark for data computation.

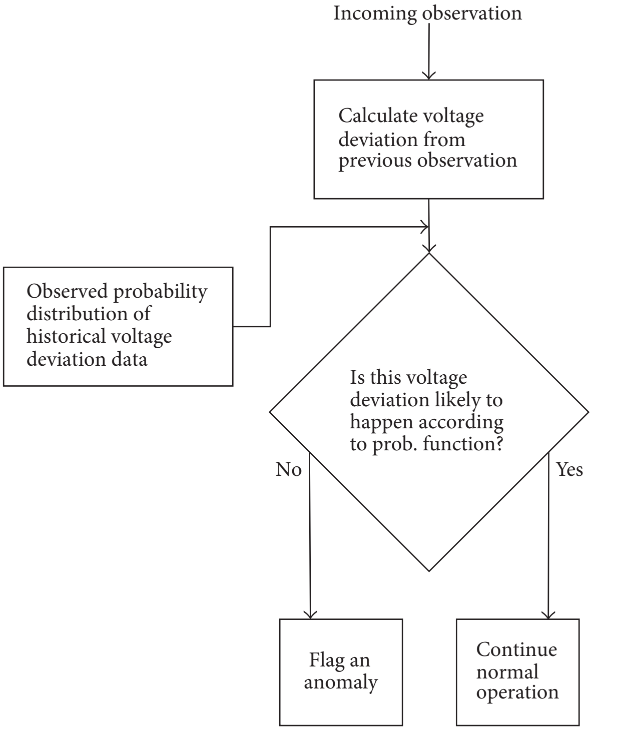
Incoming observation
Calculate voltage deviation from previous observation
Observed probability distribution of historical voltage deviation data
Is this voltage deviation likely to happen according to prob. function?
No
Yes
Flag an anomaly
Continue normal operation

Figure 4: Flowchart of the Smart Grid anomaly detection procedure proposed by Wallace et al. [18].

5. Discussion

In this section, we discuss the proposals presented in Section 4, pointing to the advantages and disadvantages of the proposals, stressing their applicability to INs.

Table 2 shows a comparison of the presented works, according to different criteria:

(i) Domain refers to the network type the proposal has been defined to work in: IT or IN.

(ii) Granularity: Axelsson [95] defines granularity of data processing as a "category that contrasts systems that process data *continuously* with those that process data in *batches* at a regular interval."

(iii) Time of detection: Axelsson [95] defines this category by defining the two main groups that compose it: systems that give results in *real-time* or near real-time and those that process data with some delay *(nonreal)*. Though related to the previous category,

Table 2: Comparison of the surveyed works.

Name	Ref.	Domain	Granul.	Time of detect.	Sources	Main Detect. technique
Beehive	[23]	IT	Batch	Nonreal	Proxy logs	k-means
Bumgardner and Marek	[24]	IT	Both	Real	Network flows	Established thresholds
Camacho et al.	[17]	IT	Both	Nonreal	Firewall & IDS logs	PCA
Dromard et al.	[25]	IT	Batch	Nonreal	Network flows	DBSCAN
Difallah et al.	[26]	IN	Both	Real	Process data	LISA
Giura and Wang	[27]	IT	Batch	Nonreal	Network and application data	Threshold establishing
Gupta and Kulariya	[28]	IT	Batch	Nonreal	Network captures	Several feature extraction and classification algorithms
Gonçalves et al.	[29]	IT	Batch	Nonreal	DHCP, Authentication and Firewall logs	EM
Hadžiosmanović et al.	[30]	IN	Batch	Nonreal	SCADA logs	FP-Graph
Hashdoop	[31]	IT	Batch	Nonreal	Network traffic (textual format)	None
Hurst et al.	[32]	IN	Batch	Nonreal	Process data	Multiple classification algs.
Iturbe et al.	[33]	IN	Batch	Nonreal	Network flows	Whitelisting
Kiss et al.	[34]	IN	Batch	Nonreal	Process data	k-means
Marchal et al.	[35]	IT	Batch	Nonreal	Honeypot, DNS, HTTP and Network flow data	Threshold establishing
MATATABI	[36]	IT	Batch	Nonreal	DNS records, Network flows, Spam email	Multiple
Rathore et al.	[37]	IT	Batch	Nonreal	Network flows	C4.5, RepTree
Ratner and Kelly	[38]	IT	Batch	Nonreal	Network packets	Manual data querying
Therdphapiyanak and Piromsopa	[39]	IT	Batch	Nonreal	Network logs	k-means
TADOOP	[40]	IT	Batch	Nonreal	Network flows	DTE-FP
Wallace et al.	[18]	IN	Continuous	Real	Process data	Cumulative Probability Distribution
Wang et al.	[41]	IT	Continuous	Real	Network flows	Intergroup entropy, LMS
Xu et al.	[42]	IT	Batch	Nonreal	Console logs	PCA

they do not overlap, as some real-time systems might process microbatches, thus giving real-time or almost real-time performance.

(iv) Source of information refers to the type of input data the ADS collects and audits for anomaly detection.

(v) Main detection technique refers to the main technique the ADS leverages to detect anomalies in the gathered information.

As Table 2 shows, most of the proposals are both batch and in non-real-time. Moreover, a similar set of proposals use a single type of data input as the source for audit information. Thus, it can be stated that the majority of these proposals focus on handling large, resting data volumes for anomaly detection (one of the V Big Data dimensions) while the other dimensions (mainly velocity and variety) are not as relevant.

Table 3 shows the Big Data adoption level of the proposals, by listing the following metrics:

(i) Locus of data collection (LDC): Axelsson [95] notes that “audit data can be collected from many different sources in a distributed fashion, or from a single point using the centralised approach.”

(ii) Locus of data processing (LDP): similarly, Axelsson states that “audit data can either be processed in a central location, or is collected and collated from many different sources in a distributed fashion.”

(iii) Underlying solution lists the underlying Big Data technology the ADS uses for Big Data computing.

(iv) Evaluation environment shows the nature of the evaluation data used to test the performance of the ADS.

Table 3 shows that most of the proposals use distributed computing for data processing. However, distributed data collection, where data from different sources is analyzed, is not as widespread. Hadoop and Spark are the most prominent Big Data frameworks that are used for anomaly detection. It is worth mentioning that, in some proposals, although Big Data mechanisms are used, they are only used in a part of the data pipeline. For instance, they use the Big Data tools for feature

Table 3: Big Data comparison of the surveyed works.

Name	Ref.	LDC	LDP	Solution	Eval. environ.
Beehive	[23]	Dist.	Dist.	Hadoop, Hive	Operational network
Bumgardner and Marek	[24]	Dist.	Dist.	Storm, HBase, Hadoop	Operational network
Camacho et al.	[17]	Dist.	Unknown	Custom	Public dataset
Dromard et al.	[25]	Dist.	Dist.	Spark	Operational network
Difallah et al.	[26]	Dist.	Dist.	Storm	Simulated process data
Giura and Wang	[27]	Dist.	Dist.	Hadoop	Operational network
Gupta and Kulariya	[28]	Cent.	Dist.	Spark	Public dataset
Gonçalves et al.	[29]	Dist.	Dist.	Hadoop, Weka	Operational network
Hadžiosmanović et al.	[30]	Cent.	Cent.	Custom	Operational network
Hashdoop	[31]	Cent.	Dist.	Hadoop	Public dataset
Hurst et al.	[32]	Cent.	Unknown	Custom	Simulated process data
Iturbe et al.	[33]	Cent.	Dist.	Elasticsearch	Operational network
Kiss et al.	[34]	Cent.	Dist.	Hadoop	Simulated process data
Marchal et al.	[35]	Dist.	Dist.	Hadoop, Hive, Pig, Spark	Operational network
MATATABI	[36]	Dist.	Dist.	Hive	Operational network
Rathore et al.	[37]	Cent.	Dist.	Spark, Weka	Public Dataset
Ratner and Kelly	[38]	Cent.	Dist.	Hadoop	Operational network
Therdphapiyanak and Piromsopa	[39]	Cent.	Dist.	Hadoop, Mahout	Public Dataset
TADOOP	[40]	Cent.	Dist.	Hadoop	Operational network
Wallace et al.	[18]	Dist.	Dist.	Spark	Operational network
Wang et al.	[41]	Dist.	Dist.	Storm	Operational network
Xu et al.	[42]	Cent.	Dist.	Hadoop	Operational network

Table 4: Suitability of IT-based solutions for their use in INs.

Name	Ref.	OSI layer	IN interoperability	Self-security
Beehive	[23]	7	Low	Medium
Bumgardner and Marek	[24]	3, 4	Medium	Medium
Camacho et al.	[17]	3, 4, 7	Medium	Unknown
Dromard et al.	[25]	3, 4	Medium	Medium
Giura and Wang	[27]	3, 4, 7	Medium	Medium
Gupta and Kulariya	[28]	3, 4, 7	Medium	Medium
Gonçalves et al.	[29]	3, 4, 7	Low	Medium
Hashdoop	[31]	Packet captures	Dependent on implementation	Medium
Marchal et al.	[35]	3, 4, 7	Medium	Medium
MATATABI	[36]	3, 4, 7	Medium	Medium
Rathore et al.	[37]	3, 4	Medium	Medium
Ratner and Kelly	[38]	Packet captures	Medium	Medium
Therdphapiyanak and Piromsopa	[39]	3, 4, 7	Medium	Medium
TADOOP	[40]	3, 4	Medium	Medium
Wang et al.	[41]	3, 4	Medium	Medium
Xu et al.	[42]	7	Medium	Medium

extraction, while once the features have been extracted into a smaller feature dataset, other conventional tools are used for the data classification.

Table 4 summarizes the suitability of the IT-based solutions to be used in INs. For that end, it defines the following metrics:

(i) OSI layer refers to the corresponding layer of the Open Systems Interconnection (OSI) model the network data belongs to. In the case of logs, it shows the layer of the network application that created the logs.

(ii) IN interoperability refers to the performance of running the IT ADS, out-of-the-box in an industrial environment. Low interoperability means that the ADS would not be usable. Medium means that the ADS is expected to run on INs and to detect anomalies to some extent. High means that the ADS is also tailored to work in IN environments.

(iii) Response type categorizes the ADSs in two categories, not related to the detection mechanism, but to their response when an anomaly is flagged. Passive responses consist of logging and sending alerts, without interacting with the traffic, while active responses try to tackle the source of the intrusion or anomaly. Active response mechanisms are often referred to as Intrusion Prevention Systems (IPSs). In this paper, all surveyed works have passive responses. The usage of active responses that fit well into the availability constraints of INs is still an undeveloped field [10].

(iv) Self-security: Zhu and Sastry [96] define self-security as "whether the proposed ADS itself is secure in the sense it will fail-safe." Availability is an important concern in INs. As such, redundant and fail-safe mechanisms are widespread in INs.

As Table 4 lists, most proposals, especially the ones that work with network flows, are able to work in INs, as, nowadays, IT networks and the cyber layer of IT networks share the same network stack at the OSI 3 and 4 layers (Network and Transport) and similar network infrastructure coexists in both types of networks (e.g., firewalls). However, even if technically possible, it is yet to be seen how they would perform.

It is worth mentioning that even if not listed in Table 4, the Time of Detection feature (covered in Table 2) becomes a relevant aspect of the ADSs when measuring their suitability for INs, as their real-time nature requires fast detection to raise alerts as fast as possible and to perform mitigation actions if necessary [10].

Furthermore, although Big Data ADSs listed were not designed for the availability constraints of INs, the usage of distributed file-systems for data storage and the distributed nature of Big Data processing give most solutions a relative defense against faults, as shown by the self-security field. However, this distributed approach might not be enough when considering the high availability requirements that ICSs and INs have and additional measures might be necessary to improve the availability of the ADS, but still, it makes Big Data ADSs better candidates in this aspect than their conventional counterparts.

6. Future Research Lines

There are several open research lines in the area of Big Data ADSs for INs. We categorize them based on the different Big Data dimensions.

6.1. Dealing with Volume. Most surveyed Big Data ADSs have dealt with large volumes of data, and in many cases it has been the main focus of the Big Data ADS. Indeed, some of the surveyed works go no further than applying conventional algorithms and approaches using Big Data mechanisms.

Therefore, the volume requirement for Big Data ADSs can be considered as partially fulfilled. However, there is still room for improvement that can lead to further research:

(i) No large-scale cyber-level ADSs for IN specific protocols: some IT counterparts deal with application-level data (7th OSI layer) but no proposals exist for INs. While lower OSI level proposals exist and could be applied to INs, these kind of mechanisms have been more studied and attackers expect related defensive measures [10]. Therefore, it is necessary to develop large-scale ADSs that will gather information from IN specific protocols, opening the way of analyzing packet payload information.

(ii) Big data IN storage: though process data has been traditionally stored in historian servers, novel approaches for the storage of IN related data are necessary: not only process readings, but wider types of data (network traces, process readings, alerts, etc.). This can help not only with anomaly detection but also for other fields of research regarding INs and Big Data.

6.2. Dealing with Velocity. As stated in Section 5, the vast majority of the proposals are neither continuous nor real-time. This presents the issue that the mentioned approaches are only capable of finding anomalies over historical data, and when new data arrives, a new, larger version of the original dataset that contains the new data is computed again in order to find anomalies. In some of the proposals historical data is divided in time bins and only data corresponding to an specific time bin is executed.

However, this is an impractical approach for a realistic ADS, more so in INs where, as previously stated, real-time detection is an important aspect. It is necessary to develop streaming models where incoming data is treated on arrival in order to detect anomalies. An issue regarding streaming models is that it is not possible to perform Exploratory Data Analysis (EDA) on them. EDA and the building of several models require data at rest, so relationships between different observations can be defined. Similarly, most streaming models need well-defined models for acting on incoming data.

A solution to this problem might lay in building hybrid models based on a two-phase approach where (i) a model is defined based on gathered historical data at rest. (ii) After building a model, this model is applied to compute incoming streaming data. INs have the advantage over IT networks that they are more static and deterministic by nature, so two-phase ADSs seem a viable solution, as once an ADS model is built, it will seldom require an update.

It is necessary to mention that, to encourage and compare different contributions in the area of real-time ADSs for INs, it is necessary to create and use a set of metrics where latency should be taken into account [10].

6.3. Dealing with Variety. ICSs are multivariate and heterogeneous by nature; they deal with very diverse types of data, both at the network level (packets, flows, logs, etc.) and, notably, at the field level where they keep track of a large number of different physical quantities simultaneously. However, existing large-scale ADSs do not leverage data from both levels and instead focus on a single or few data sources for anomaly detection. This issue is extensible to most conventional ADSs as well, as only a few proposals deal with both process-level and network-level data [63, 64] to detect anomalies.

Moreover, IN networks are also heterogeneous in the sense that various technologies coexist at a network and field level. INs are multivendor environments, where devices might be powered by different technologies and communicate using different protocols. This requires the development of different tools to extract data from devices belonging to different vendors. In addition, many of these devices might be limited in terms of computation, so, in order to avoid latencies and availability issues, it is necessary to extract ICS data out of the devices themselves in an unobtrusive manner, where it can later be computed in a cluster using a distributed computing framework, separate from the critical IN sections.

In this sense, analyzing and aggregating information sources from different levels can aid in the detection of complex attacks directed against INs [110]. BDA gives the opportunity to use this heterogeneous data and leverage it in a unified manner to detect anomalies. In this direction, the work of Camacho et al. [17, 111] gives promising insight. The usage of multivariate algorithms, such as PCA, can help to build a model where parametrized cyber and process data can be used to build a single ADS that leverages data from both levels. PCA-based techniques scale well horizontally and are used in fields such as genomics where they are used to handle massively dimensional data.

6.4. Dealing with Veracity. From our point of view, Big Data veracity for anomaly detection is not only related to correctly flagging a relevant anomaly on a large dataset, but also to communicating and alerting the anomaly correctly, instead of overwhelming the operator with too much alert noise. In a related note, we believe that properly testing different Big Data ADSs on neutral, relevant environments such as using public datasets is also ensuring the veracity of ADSs in Big Data. Therefore, we can identify the following research areas:

(i) Closing the semantic gap: Sommer and Paxson [112] define the semantic gap as the lack of actionable reports for the network operator. In other words, the ADS does not provide sufficient diagnosis information to aid decision making for the operator. In INs, it is necessary for an operator to know what is the cause for an anomaly, as successful attacks or serious disturbances could have potentially catastrophic outcomes. BDA can help to provide useful information about the cause of the anomaly. Big Data visualization techniques or Visual Analytics might play a significant role in this matter.

(ii) Necessity to have realistic, large-scale datasets: few datasets exist for anomaly detection evaluation in INs and existing datasets [113] are too small to evaluate Big Data ADSs. Therefore, it is necessary to have public, realistic, large-scale IN datasets that would allow the evaluation of the ADS performance independently.

(iii) Integration of honeypots: when trying to find anomalies in Big Data, it is important to keep the value of false positives and false negatives low. The task of finding anomalies is equivalent to finding a needle in a haystack. Trusted data sources can help in this endeavor. Honeypots can constitute such a trusted information source, as by definition they do not yield any false positives [114]. The field of IN-oriented honeypots is still maturing, though a few approaches have been proposed [114, 115], but the possibility of feeding and correlating IN honeypot data to a Big Data IN ADS, in a similar fashion as Marchal et al. [35], opens the way to a new field of research.

7. Conclusions

We have presented a survey paper comprising three main contributions: (i) a review of current proposals of Big Data ADSs that can be applied to INs, (ii) a novel taxonomy to classify existing IN-based ADSs, and (iii) a collection of possible future research areas in the field of large-scale, heterogeneous, and real-time ADSs for INs.

Big Data anomaly detection in Industrial Networks is still a developing field. Few proposals exist for INs exclusively, but some IT-based solutions show that it is possible to have similar counterparts on INs. Nevertheless, while most proposals focus on large-volume solutions for anomaly detection, other aspects, such as dealing with data with high velocity or variety, are still largely untackled. We finally have offered some future research work areas regarding these open issues.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been developed by the Intelligent Systems for Industrial Systems group supported by the Department of Education, Language Policy and Culture of the Basque Government. This work has been partially funded by the European Union's Horizon 2020 research and innovation programme project PROPHESY, under Grant Agreement no. 766994.

References

[1] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security, Special publication 800-82," Tech. Rep., National Institute of Standards and Technology, June 2011.

[2] European Council, "Council Directive 2008/114/EC," Tech. Rep., Official Journal of the European Union, December 2008.

[3] B. Miller and D. C. Rowe, "A survey of SCADA and critical infrastructure incidents," in *Proceedings of the 1st Annual Conference on Research in Information Technology, RIIT 2012*, pp. 51–56, ACM, October 2012.

[4] M. Zeller, "Myth or reality - Does the Aurora vulnerability pose a risk to my generator?" in *Proceedings of the 64th Annual*

Conference for Protective Relay Engineers, pp. 130–136, April 2011.

[5] R. Langner, “Stuxnet: dissecting a cyberwarfare weapon,” *IEEE Security and Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[6] J. Slay and M. Miller, *Lessons learned from the Maroochy Water Breach*, Springer, 2007.

[7] “Bundesamt für Sicherheit in der Informationstechnik,” Tech. Rep., Die Lage der IT-Sicherheit in Deutschland, 2014.

[8] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, “Duqu: Analysis, detection, and lessons learned,” in *ACM European Workshop on System Security (EuroSec)*, 2012.

[9] Symantec Incident Response, “Dragonfly: Cyberespionage attacks against energy suppliers,” Tech. Rep., July 2014.

[10] R. Mitchell and I.-R. Chen, “A survey of intrusion detection techniques for cyber-physical systems,” *ACM Computing Surveys*, vol. 46, no. 4, article no. 55, 2014.

[11] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on SCADA systems,” in *Proceedings of the 2011 IEEE International Conference on Internet of Things, iThings 2011 and 4th IEEE International Conference on Cyber, Physical and Social Computing, CPSCom 2011*, pp. 380–388, October 2011.

[12] I. Garitano, R. Uribeetxeberria, and U. Zurutuza, “A review of SCADA anomaly detection systems,” in *Proceedings of the6th International Conference Soft Computing Models in Industrial and Environmental Applications SOCO 2011*, vol. 87, pp. 357–366, Springer, 2011.

[13] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[14] D. Borthakur, “The hadoop distributed file system: Architecture and design,” 2007.

[15] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[16] D. Laney, “3D data management: Controlling data volume, velocity and variety,” *META Group Research Note*, vol. 6, 2001.

[17] J. Camacho, G. Macia-Fernandez, J. Diaz-Verdejo, and P. Garcia-Teodoro, “Tackling the big data 4 vs for anomaly detection,” in *Proceedings of the IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2014*, pp. 500–505, May 2014.

[18] S. Wallace, X. Zhao, D. Nguyen, and K.-T. Lu, “Big data analytics on smart grid: Mining pmu data for event and anomaly detection,” in *Big Data: Principles and Paradigms*, R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, Eds., chapter 17, pp. 417–429, Morgan Kaufmann, 2016.

[19] A. A. Cárdenas, P. K. Manadhata, and S. P. Rajan, “Big Data Analytics for Security,” *Security & Privacy, IEEE*, vol. 11, no. 6, pp. 74–76, 2013.

[20] C. Everett, “Big data - The future of cyber-security or its latest threat?” *Computer Fraud and Security*, vol. 2015, no. 9, pp. 14–17, 2015.

[21] M. Cheminod, L. Durante, and A. Valenzano, “Review of security issues in industrial networks,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 277–293, 2013.

[22] B. Galloway and G. P. Hancke, “Introduction to industrial control networks,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 860–880, 2013.

[23] T.-F. Yen, A. Oprea, K. Onarlioglu et al., “Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks,” in *Proceedings of the 29th Annual Computer Security Applications Conference, ACSAC 2013*, pp. 199–208, December 2013.

[24] V. K. C. Bumgardner and V. W. Marek, “Scalable hybrid stream and hadoop network analysis system,” in *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering, ICPE 2014*, pp. 219–224, Association for Computing Machinery, Dublin, Ireland, March 2014.

[25] J. Dromard, G. Roudire, and P. Owezarski, “Unsupervised network anomaly detection in real-time on big data,” in *New Trends in Databases and Information Systems*, T. Morzy, P. Valduriez, and L. Bellatreche, Eds., vol. 539 of *Communications in Computer and Information Science*, pp. 197–206, Springer, Berlin, Germany, 2015.

[26] D. E. Difallah, P. Cudre-Mauroux, and S. A. McKenna, “Scalable anomaly detection for smart city infrastructure networks,” *IEEE Internet Computing*, vol. 17, no. 6, pp. 39–47, 2013.

[27] P. Giura and W. Wang, “Using large scale distributed computing to unveil advanced persistent threats,” *SCIENCE*, vol. 1, no. 3, pp. 93–105, 2012.

[28] G. P. Gupta and M. Kulariya, “A framework for fast and efficient cyber security network intrusion detection using apache spark,” *Procedia Computer Science*, vol. 93, pp. 824–831, 2016.

[29] D. Gonçalves, J. Bota, and M. Correia, “Big data analytics for detecting host misbehavior in large logs,” in *Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015*, pp. 238–245, fin, August 2015.

[30] D. Hadžiosmanović, D. Bolzoni, and P. H. Hartel, “A log mining approach for process monitoring in SCADA,” *International Journal of Information Security*, vol. 11, no. 4, pp. 231–251, 2012.

[31] R. Fontugne, J. Mazel, and K. Fukuda, “Hashdoop: A MapReduce framework for network anomaly detection,” in *Proceedings of the 2014 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2014*, pp. 494–499, May 2014.

[32] W. Hurst, M. Merabti, and P. Fergus, “Big data analysis techniques for cyber-threat detection in critical infrastructures,” in *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, pp. 916–921, May 2014.

[33] M. Iturbe, I. Garitano, U. Zurutuza, and R. Uribeetxeberria, “Visualizing Network Flows and Related Anomalies in Industrial Networks using Chord Diagrams and Whitelisting,” in *Proceedings of the International Conference on Information Visualization Theory and Applications*, pp. 99–106, Rome, Italy, Feburary 2016.

[34] I. Kiss, B. Genge, P. Haller, and G. Sebestyen, “Data clustering-based anomaly detection in industrial control systems,” in *Proceedings of the 2014 10th IEEE International Conference on Intelligent Computer Communication and Processing, ICCP 2014*, pp. 275–281, Cluj Napoca, Romania, September 2014.

[35] S. Marchal, X. Jiang, R. State, and T. Engel, “A big data architecture for large scale security monitoring,” in *Proceedings of the 3rd IEEE International Congress on Big Data, BigData Congress*, pp. 56–63, IEEE Computer Society, Anchorage, Alaska, USA, July 2014.

[36] H. Tazaki, K. Okada, Y. Sekiya, and Y. Kadobayashi, “MATATABI: Multi-layer Threat Analysis Platform with Hadoop,” in *Proceedings of the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS 2014*, pp. 75–82.

[37] M. M. Rathore, A. Ahmad, and A. Paul, “Real time intrusion detection system for ultra-high-speed big data environments,”

The Journal of Supercomputing, vol. 72, no. 9, pp. 3489–3510, 2016.

[38] A. S. Ratner and P. Kelly, "Anomalies in network traffic," in *Proceedings of the 11th IEEE International Conference on Intelligence and Security Informatics, IEEE ISI 2013*, pp. 206–208, June 2013.

[39] J. Therdphapiyanak and K. Piromsopa, "An analysis of suitable parameters for efficiently applying K-means clustering to large TCPdump data set using Hadoop framework," in *Proceedings of the 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2013*, pp. 1–6, May 2013.

[40] G. Tian, Z. Wang, X. Yin et al., "TADOOP: Mining Network Traffic Anomalies with Hadoop," in *Security and Privacy in Communication Networks*, B. Thuraisingham, X. Wang, and V. Yegneswaran, Eds., vol. 164 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 175–192, Springer, 2015.

[41] Z. Wang, J. Yang, H. Zhang, C. Li, S. Zhang, and H. Wang, "Towards online anomaly detection by combining multiple detection methods and Storm," in *Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium, NOMS 2016*, pp. 804–807, tur, April 2016.

[42] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proceedings of the 22nd ACM SIGOPS Symposium on Operating Systems Principles, SOSP'09*, pp. 117–131, October 2009.

[43] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in SCADA networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006.

[44] B. Genge, C. Siaterlis, and M. Hohenadel, "Impact of network infrastructure parameters to the effectiveness of cyber attacks against Industrial Control Systems," *International Journal of Computers, Communications & Control*, vol. 7, no. 4, pp. 674–687, 2012.

[45] ISO, "EnglishInformation technology – Security techniques – Information security management systems – Requirements. ISO/IEC 27001:2013," Tech. Rep., International Organization for Standardization, 2013.

[46] M. Bishop, *Computer Security: Art and Science*, Addison-Wesley Professional, 2002.

[47] D. Dzung, M. Naedele, T. P. von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, 2005.

[48] D. Duggan, M. Berg, J. Dillinger, and J. Stamp, "Penetration testing of industrial control systems," Tech. Rep. SAND2005-2846P, Sandia National Laboratories, 2005.

[49] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805–822, 1999.

[50] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[51] M. Obitko, V. Jirkovský, and J. Bezdíček, "Big Data Challenges in Industrial Automation," in *Industrial Applications of Holonic and Multi-Agent Systems*, V. Marík, J. Lastra, and P. Skobelev, Eds., vol. 8062 of *Lecture Notes in Computer Science*, pp. 305–316, Springer, Berlin, Heidelberg, 2013.

[52] H. P. Zhu, Y. Xu, Q. Liu, and Y. Q. Rao, "Cloud service platform for big data of manufacturing," *Applied Mechanics and Materials*, vol. 456, pp. 178–183, 2014.

[53] S. Windmann, A. Maier, O. Niggemann et al., "Big data analysis of manufacturing processes," *Journal of Physics: Conference Series*, vol. 659, no. 1, Article ID 012055, 2015.

[54] M. Kezunovic, L. Xie, and S. Grijalva, "The role of big data in improving power system operation and protection," in *Proceedings of the 2013 IREP Symposium on Bulk Power System Dynamics and Control - IX Optimization, Security and Control of the Emerging Power Grid, IREP 2013*, August 2013.

[55] J. Wan, S. Tang, D. Li et al., "A Manufacturing Big Data Solution for Active Preventive Maintenance," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2039–2047, 2017.

[56] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic, "Big-data-driven anomaly detection in industry (4.0): An approach and a case study," in *Proceedings of the 4th IEEE International Conference on Big Data, Big Data 2016*, pp. 1647–1652, December 2016.

[57] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial Big Data for Fault Diagnosis: Taxonomy, Review, and Applications," *IEEE Access*, 2017.

[58] W. Shi, Y. Zhu, T. Huang et al., "An Integrated Data Preprocessing Framework Based on Apache Spark for Fault Diagnosis of Power Grid Equipment," *Journal of Signal Processing Systems*, vol. 86, no. 2-3, pp. 221–236, 2017.

[59] N. Svendsen and S. Wolthusen, "Using Physical Models for Anomaly Detection in Control Systems," in *Critical Infrastructure Protection III*, vol. 311 of *IFIP Advances in Information and Communication Technology*, pp. 139–149, Springer, Berlin, Heidelberg, 2009.

[60] M. Krotofil, J. Larsen, and D. Gollmann, "The process matters: Ensuring data veracity in Cyber-physical systems," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2015*, pp. 133–144, Singapore, April 2015.

[61] I. Kiss, B. Genge, and P. Haller, "A clustering-based approach to detect cyber attacks in process control systems," in *Proceedings of the 13th International Conference on Industrial Informatics, INDIN 2015*, pp. 142–148, July 2015.

[62] M. Iturbe, J. Camacho, I. Garitano, U. Zurutuza, and R. Uribeetxeberria, "On the Feasibility of Distinguishing between Process Disturbances and Intrusions in Process Control Systems Using Multivariate Statistical Process Control," in *Proceedings of the 46th IEEE/IFIP International Conference on Dependable Systems and Networks, DSN-W 2016*, pp. 155–160, France, July 2016.

[63] B. Genge, C. Siaterlis, and G. Karopoulos, "Data fusion-base anomay detection in networked critical infrastructures," in *Proceedings of the 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, pp. 1–8, Budapest, Hungary, June 2013.

[64] W. Jardine, S. Frey, B. Green, and A. Rashid, "SENAMI: Selective non-invasive active monitoring for ICS intrusion detection," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and PrivaCy, CPS-SPC 2016*, pp. 23–34, Vienna, Austria, October 2016.

[65] A. Bialecki, M. Cafarella, D. Cutting, and O. OMalley, "Hadoop: a framework for running applications on large clusters built of commodity hardware," AT Wiki, 2005, http://hadoop.apache.org.

[66] P. Mundkur, V. Tuulos, and J. Flatow, “Disco: A computing platform for large-scale data analytics,” in *Proceedings of the 10th ACM SIGPLAN Erlang Workshop, Erlang 2011, Co-located with the Annual ACM SIGPLAN International Conference on Functional Programming, ICFP*, pp. 84–89, jpn, September 2011.

[67] M. Gorawski, A. Gorawska, and K. Pasterak, “A survey of data stream processing tools,” in *Information Sciences and Systems 2014*, pp. 295–303, Springer International Publishing, Cham, 2014.

[68] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: cluster computing with working sets,” in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, p. 10, 2010.

[69] P. Carbone, S. Ewen, S. Haridi, A. Katsifodimos, V. Markl, and K. Tzoumas, “Apache flink: Stream and batch processing in a single engine,” *Data Engineering*, p. 28, 2015.

[70] R. Zuech, T. M. Khoshgoftaar, and R. Wald, “Intrusion detection and Big Heterogeneous Data: a Survey,” *Journal of Big Data*, vol. 2, no. 1, pp. 1–41, 2015.

[71] J. François, S. Wang, W. Bronzi, R. State, and T. Engel, “BotCloud: detecting botnets using MapReduce,” in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS ’11)*, pp. 1–6, Iguacu Falls, Brazil, December 2011.

[72] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, “Big data analytics framework for peer-to-peer botnet detection using random forests,” *Information Sciences*, vol. 278, pp. 488–497, 2014.

[73] T.-W. Chiou, S.-C. Tsai, and Y.-B. Lin, “Network security management with traffic pattern clustering,” *Soft Computing*, vol. 18, no. 9, pp. 1757–1770, 2014.

[74] Z. Luo, J. Shen, H. Jin, and D. Liu, “Research of Botnet Situation Awareness Based on Big Data,” in *Web Technologies and Applications*, vol. 9461 of *Lecture Notes in Computer Science*, pp. 71–78, Springer International Publishing, 2015.

[75] L. Invernizzi, S. Miskovic, R. Torres et al., “Nazca: Detecting Malware Distribution in Large-Scale Networks,” in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, Calif, USA, 2014.

[76] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, “Polonium: Tera-Scale Graph Mining and Inference for Malware Detection,” in *Proceedings of the SIAM International Conference on Data Mining (SDM) 2011*, Mesa, Ariz, USA, April 2011.

[77] S. T. Liu and Y. M. Chen, “Retrospective detection of malware attacks by cloud computing,” *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 3, pp. 280–296, 2011.

[78] C. R. Panigrahi, M. Tiwari, B. Pati, and R. Prasath, “Malware Detection in Big Data Using Fast Pattern Matching: A Hadoop Based Comparison on GPU,” in *Mining Intelligence and Knowledge Exploration*, vol. 8891 of *Lecture Notes in Computer Science*, pp. 407–416, Springer International Publishing, 2014.

[79] J. Jang, D. Brumley, and S. Venkataraman, “Bitshred: Fast, Scalable Malware Triage,” Tech. Rep., Cylab, Carnegie Mellon University, Pittsburgh, Pa, USA, 2010.

[80] Z. Hanif, C. Telvis, and J. Trost, “BinaryPig: Scalable Static Binary Analysis Over Hadoop,” in *Proceedings of the Blackhat USA*, Las Vegas, Nev, USA, 2013.

[81] M. Mizukoshi and M. Munetomo, “Distributed denial of services attack protection system with genetic algorithms on Hadoop cluster computing framework,” in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015*, pp. 1575–1580, May 2015.

[82] S. Tripathi, B. Gupta, A. Almomani, A. Mishra, and S. Veluru, “Hadoop Based Defense Solution to Handle Distributed Denial of Service (DDoS) Attacks,” *Journal of Information Security*, vol. 04, no. 03, pp. 150–164, 2013.

[83] Y. Lee and Y. Lee, “Detecting DDoS attacks with Hadoop,” in *Proceedings of the 2011 ACM CoNext Student Workshop, CoNEXT 2011*, ACM, December 2011.

[84] J. Choi, C. Choi, B. Ko, D. Choi, and P. Kim, “Detecting web based DDoS attack using MapReduce operations in cloud computing environment,” *Journal of Internet Services and Information Security*, vol. 3, pp. 28–37, Nov 2013.

[85] T. Zhao, D. C.-T. Lo, and K. Qian, “A neural-network based DDoS detection system using hadoop and HBase,” in *Proceedings of the 17th IEEE International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security and IEEE 12th International Conference on Embedded Software and Systems, HPCC-ICESS-CSS 2015*, pp. 1326–1331, August 2015.

[86] G. Caruana, M. Li, and H. Qi, “SpamCloud: A mapreduce based anti-spam architecture,” in *Proceedings of the 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010*, pp. 3003–3006, August 2010.

[87] G. Caruana, M. Li, and M. Qi, “A MapReduce based parallel SVM for large scale spam filtering,” in *Proceedings of the 2011 8th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011, Jointly with the 2011 7th International Conference on Natural Computation, ICNC’11*, pp. 2659–2662, July 2011.

[88] P. H. B. Las-Casas, V. S. Dias, W. Meira, and D. Guedes, “A Big Data Architecture for Security Data and Its Application to Phishing Characterization,” in *Proceedings of the 2nd IEEE International Conference on Big Data Security on Cloud, IEEE BigDataSecurity 2016, 2nd IEEE International Conference on High Performance and Smart Computing, IEEE HPSC 2016 and IEEE International Conference on Intelligent Data and Security, IEEE IDS 2016*, pp. 36–41, April 2016.

[89] M. Thomas, L. Metcalf, J. Spring, P. Krystosek, and K. Prevost, “SiLK: A tool suite for unsampled network flow analysis at scale,” in *Proceedings of the 3rd IEEE International Congress on Big Data, BigData Congress*, pp. 184–191, Anchorage, Alaska, USA, July 2014.

[90] Y. Lee, W. Kang, and H. Son, “An internet traffic analysis method with MapReduce,” in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium Workshops, NOMS 2010*, pp. 357–361, April 2010.

[91] M. Baker, D. Turnbull, and G. Kaszuba, “Finding Data Needles in Haystacks (the Size of Countries,” *in Prooceedings of Blackhat Europe*, 2012.

[92] Y. Lee, W. Kang, and Y. Lee, “A hadoop-based packet trace processing tool,” in *Traffic Monitoring and Analysis*, vol. 6613 of *Lecture Notes in Computer Science*, pp. 51–63, Springer, Berlin, Heidelberg, 2011.

[93] M. Kumar and M. Hanumanthappa, “Scalable intrusion detection systems log analysis using cloud computing infrastructure,” in *Proceedings of the 2013 4th IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*, December 2013.

[94] S.-F. Yang, W.-Y. Chen, and Y.-T. Wang, “ICAS: An inter-VM IDS log cloud analysis system,” in *Proceedings of the 2011 IEEE*

International Conference on Cloud Computing and Intelligence Systems, CCIS2011, pp. 285–289, September 2011.

[95] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Tech. Rep., Chalmers University of Technology, 2000.

[96] B. Zhu and S. Sastry, "SCADA-specific intrusion detection/prevention systems: a survey and taxonomy," in *Proceedings of the in Proceedings of the 1st Workshop on Secure Control Systems (SCS' 10)*, 2010.

[97] A. Thusoo, J. S. Sarma, and N. Jain, "Hive: a warehousing solution over a map-reduce framework," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1626–1629, 2009.

[98] J. Therdphapiyanak and K. Piromsopa, "Applying hadoop for log analysis toward distributed IDS," in *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, ICUIMC 2013*, January 2013.

[99] H. Hotelling, "Multivariate quality control," *Techniques of Statistical Analysis*, 1947.

[100] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[101] A. Ziviani, A. T. A. Gomes, M. L. Monsores, and P. S. S. Rodrigues, "Network anomaly detection using nonextensive entropy," *IEEE Communications Letters*, vol. 11, no. 12, pp. 1034–1036, 2007.

[102] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[103] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, 2012.

[104] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pp. 226–231, 1996.

[105] Z. Wang, J. Yang, and F. Li, "An on-line anomaly detection method based on LMS algorithm," in *Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium, APNOMS 2014*, September 2014.

[106] B. Schneier, "Attack trees," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21–29, 1999.

[107] E. J. Amoroso, *Fundamentals of Computer Security*, PTR Prentice Hall, Englewood Cliffs, NJ, USA, 1994.

[108] L. Anselin, "Local indicators of spatial association—LISA," *Geographical Analysis*, vol. 27, no. 2, pp. 93–115, 1995.

[109] P. G. Brown, "Overview of sciDB: large scale array storage, processing and analysis," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 963–968, ACM, 2010.

[110] E. Bompard, P. Cuccia, M. Masera, and I. N. Fovino, "Cyber vulnerability in power systems operation and control," in *Critical Infrastructure Protection*, vol. 7130, pp. 197–234, Springer, 2012.

[111] J. Camacho, R. Magán-Carrión, P. García-Teodoro, and J. J. Treinen, "Networkmetrics: multivariate big data analysis in the context of the internet," *Journal of Chemometrics*, vol. 30, no. 9, pp. 488–505, 2016.

[112] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 305–316, IEEE Computer Society, 2010.

[113] T. Morris and W. Gao, "Industrial Control System Traffic Data Sets for Intrusion Detection Research," in *Critical Infrastructure Protection VIII*, vol. 441 of *IFIP Advances in Information and Communication Technology*, pp. 65–78, Springer, Berlin, Heidelberg, 2014.

[114] E. Vasilomanolakis, S. Srinivasa, C. G. Cordero, and M. Mühlhäuser, "Multi-stage attack detection and signature generation with ICS honeypots," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium, NOMS 2016*, pp. 1227–1232, April 2016.

[115] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards high-interaction virtual ICS honeypots-in-a-box," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and PrivaCy, CPS-SPC 2016*, pp. 13–22.