Machine Learning for Automated Industrial IoT Attack Detection: An Efficiency-Complexity Trade-off

Article *in* ACM Transactions on Management Information Systems · October 2021

DOI: 10.1145/3460822

CITATIONS
16

READS
287

5 authors, including:

Saurav Chakraborty
University of Louisville
12 PUBLICATIONS **55** CITATIONS

SEE PROFILE

Sagar Samtani
Indiana University Bloomington
99 PUBLICATIONS **2,593** CITATIONS

SEE PROFILE

Machine Learning for Automated Industrial IoT Attack Detection: An Efficiency-Complexity Trade-off

SAURAV CHAKRABORTY, Information Systems, Analytics and Operations Department, University of Louisville, Louisville, Kentucky, USA
AGNIESZKA ONUCHOWSKA, School of Information Systems and Decision Sciences, University of South Florida, Tampa, Florida, USA
SAGAR SAMTANI, Operations and Decision Technologies, Indiana University, Bloomington, Indiana, USA
WOLFGANG JANK and BRANDON WOLFRAM, School of Information Systems and Decision Sciences, University of South Florida, Tampa, Florida, USA

Critical city infrastructures that depend on smart Industrial Internet of Things (IoT) devices have been increasingly becoming a target of cyberterrorist or hacker attacks. Although this has led to multiple studies in the recent past, there exists a paucity of literature concerning real-time Industrial IoT attack detection. The goal of this article is to build a machine-learning approach using Industrial IoT sensor readings for accurately tracking down Industrial IoT attacks in real time. We analyze IoT system behavior under a lab-controlled series of attacks on a Secure Water Treatment (SWaT) system. The system is analytically challenging in that it results in sensor readings that resemble waveforms. To that end, we develop a novel early detection method using functional shape analysis (FSA) to extract features from the data that can capture the profile of the waveform. Our results show an efficiency-complexity trade-off between functional and non-functional methods in predicting IoT attacks.

CCS Concepts: • **Security and privacy** → **Intrusion detection systems**; • **Computer systems organization** → Sensors and actuators; • **Networks** → *Sensor networks*;

Additional Key Words and Phrases: Industrial IoT, cybersecurity, machine learning, functional shape analysis (FSA)

ACM Reference format:
Saurav Chakraborty, Agnieszka Onuchowska, Sagar Samtani, Wolfgang Jank, and Brandon Wolfram. 2021. Machine Learning for Automated Industrial IoT Attack Detection: An Efficiency-Complexity Trade-off. *ACM Trans. Manage. Inf. Syst.* 12, 4, Article 37 (October 2021), 28 pages.
http://dx.doi.org/10.1145/3460822

This research did not receive any specific grant from funding agencies in the public, commercial, or non-profit sectors.
Authors' Addresses: S. Chakraborty, Information Systems, Analytics and Operations Department, University of Louisville, 2301 S 3rd Street Louisville, Kentucky 40292, USA; email: saurav.chakraborty@louisville.edu; A. Onuchowska, W. Jank, and B. Wolfram, School of Information Systems and Decision Sciences, University of South Florida, 4202 E Fowler Avenue Tampa, Florida 33620, USA; emails: {aonuchowska, wjank, bwolfram}@usf.edu; S. Samtani, Operations and Decision Technologies, Indiana University, 107 S Indiana Avenue Bloomington, Indiana 47405; email: ssamtani@iu.edu.

2158-656X/2021/10-ART37 $15.00
http://dx.doi.org/10.1145/3460822

1 INTRODUCTION

Recent years have seen an increased deployment of smart solutions relying on interconnected Industrial Internet of Things (IoT) sensors and devices. The levels of IoT systems' implementation have increased due to such systems' contributions to the quality of services provided to citizens [Kitchin & Dodge 2019, Thibodeaux 2017]. For example, real-time data collected over smart water management systems can indicate water quality levels or detect leaks [Cerrudo 2015], whereas sensors that collect information about the movement of people and vehicles can help streamline city traffic flows [Maddox 2016]. The deployment of IoT-based devices is considered one of the top four future **Information Systems (IS)**-related challenges, which are likely to cause significant security problems and generate multiple adverse implications [Ransbotham et al. 2016]. Core challenges related to IoT adoption in a large industrial setting include reliability of IoT devices and protection of private data [Thibaud et al. 2018].

Businesses increasingly deploy scalable tools and solutions to securely store, manage, and analyze data generated by IoT devices [Khatoun & Zeadally 2016]. However, unlike the norms of employees' compliance regulated by information security policies (ISPs) [Yazdanmehr & Wang 2016], secure management of IoT devices is often not mandated in organizations. For example, Industrial IoT networks often lack emergency plans for prevalent cyberattacks, for example, Denial of Service (DoS) [Khatoun & Zeadally 2016]. Consequently, critical city infrastructures, which depend on smart devices, have become a target of successful cyberterrorist or hacker attacks [Oliveira 2010, Sleptchenko & Johnson 2015]. So far, reported disruptions of Industrial IoT systems include cyberattacks on water dams [Sanger 2016b], pumping stations [Walton 2017], power grids [Sanger 2016a], and emergency warning systems [Rosenberg & Salam 2017]. Examples of attacks on city infrastructures include blackouts [Sanger 2016b], improper functioning of city traffic light systems [Jones 2016], or disrupted water supply and wastewater treatment [Walton 2017]. Past incidents related to IoT-supported water management systems took the form of attacks on water pumps [Nakashima 2011] or valves that control the flow of chemicals [Leyden 2016].

Smart systems are often exposed to vulnerabilities, the consequences of which are not yet fully understood [Kitchin & Dodge 2019, Ransbotham et al. 2016]. Furthermore, the problem of cyberattacks will intensify, as innovative solutions will bring new, more sophisticated vulnerabilities [Jalali et al. 2019]. Monitoring and management of industrial systems has become a serious issue that requires innovative and sustainable solutions to solve [Bauman et al. 2017, Ketter et al. 2018]. The lack of physical controls that monitor IoT-supported devices has already been causing security failures. More security shortcomings resulting in biological or chemical water contamination of water treatment plant infrastructures, as well as physical disruption of IoT-supported city infrastructures, are likely to follow [Sleptchenko & Johnson 2015, Ransbotham et al. 2016, Espelund 2016]. Organizations deploying state-of-the-art solutions to improve efficiency and effectiveness of their operations should realize that security processes and methods [McLeod & Dolezel 2018], including cyberattack detection mechanisms installed early in the process, must accompany technology implementations.

The challenge of effective Industrial IoT cyberattack detection stems from the monitoring system's capacity to capture, process, and analyze large volumes of data in real time [Pacheco & Hariri 2016]. This study is motivated by the dearth of literature examining real-time attack detection methods for Industrial IoT systems that balance efficiency, performance, and model complexity. As pattern recognition has successfully detected anomalies in related domains such as network-based environments [Planquart 2001] or to analyze cyberattack methods [Jang-Jaccard & Nepal 2014], we take this as encouraging evidence that security attacks on Industrial IoT networks can also be effectively identified via machine learning. Our dataset is generated

in a controlled laboratory environment that simulates cyberattacks on a water management system. The resulting data resemble waveforms that require novel feature engineering methods that are subsequently used inside automated machine learning algorithms. In order to capture these waveforms quantitatively, we propose a novel set of functional shape analysis (FSA) techniques [Foutz & Jank 2010]. FSA is a subclass of Functional Data Analysis, which focuses on the analysis of curves, surfaces, and other objects rather than data vectors, as is the case in classical statistics. Using the distinct shapes extracted by FSA, we propose a novel approach for early detection of attacks and compare the proposed techniques' performance against simpler non-functional classification models. We find that the proposed functional methods outperform simpler, non-functional ones, especially when the goal is to detect attacks as early as possible.

To address these challenges of effective detection of Industrial IoT cyberattacks, we develop a novel approach using functional data analysis to detect attacks quickly. We use data generated by a Secure Water Treatment (SWaT) system located in the Center for Research in Cyber Security at Singapore University of Technology and Design [Goh et al. 2016]. Our contributions can be articulated as follows:

- First, we design and apply novel feature extraction and representation techniques based on FSA that prove effective in improving the performance of traditional machine learning algorithms in predicting Industrial IoT attacks.
- Second, we demonstrate that the functional representation of the waveforms leads to the identification of a wider variety of Industrial IoT attacks.
- Third, we perform and present a comprehensive set of validation measures to ensure the robustness of our findings. We conduct a systematic experiment demonstrating the trade-off between efficiency and complexity analysis. The results provide unprecedented insight into the complexity of varying feature sets across multiple settings.

The article is organized as follows. First, we discuss previous research related to FSA, IoT network anomaly detection, feature extraction and engineering, and real-time dynamic models. Second, we describe our data and present a description of reported system attacks. Third, we derive our novel FSA-based classification method and compare its performance against baseline approaches. We then discuss the results of our study together with business implications. Finally, we elaborate on possible future directions for extended research.

2 RELATED WORK

2.1 Industrial IoT Network Anomaly Detection Methods

Cyberattacks are increasingly targeting industrial infrastructures and often cause significant threats to the safety of citizens. One key issue that critical infrastructures face is attack complexity and non-triviality in detection [Nader et al. 2016]. The diverse types of anomaly-generating attacks on Industrial IoT networks has resulted in a broad range of anomaly detection methods. For example, Hodo et al. [2016] proposed a neural network to classify normal and distributed denial of service (DDoS) attacks on IoT network traffic. Erfani et al. [2016] built a hybrid model, which connects deep belief networks (DBNs) and a one-class support vector machine (SVM) to investigate patterns on large-scale and high-dimensional domains generated by Industrial IoT networks. Rajasegarar et al. [2014] used a hyper-spherical, cluster-based algorithm to identify global anomalies at an individual node level on an IoT network. This reduced communication overhead generated by the centralized detection approach. Raza et al. [2013] focused on routed attacks on IoT networks and proposed an algorithm-based intrusion detection system (IDS).

Cervantes et al. [2015] investigated anomaly detection related to sinkhole attacks whose goal is to hinder communication between IoT devices in a given network. The researchers proposed an IDS that establishes dynamic clustering to support data transmission in IoT networks. Fan et al. [2004] proposed a distribution-based anomaly-generation algorithm to create anomaly detection models that capture known and unknown intrusions. Hansen et al. [2007] used a genetic programming algorithm to detect system attacks; Babaie et al. [2014] applied the Linear Dynamical System (LDS) to detect deviations from temporal and spatial correlations in the data, which ultimately leads to detection of a range of cyberattack types. Jiang and Papavassiliou [2004] analyzed normal network behavior patterns and proposed an anomaly-tolerant traffic prediction algorithm that detects network attacks using the analysis of abnormal behavior generated in the network. Thottan and Ji [1999] applied an algorithm that uses sequential Generalized Likelihood Ratio (GLR) tests to identify faulty network performance. In turn, Tian and Ding [2016] used dynamic threshold base detection methods and applied diffusion wavelets to the analysis of anomalies in network traffic. Dominic and Said [2014] used an outlier detection scheme in frequent pattern mining to distinguish anomalies in network traffic from normal network behavior.

Past research also focused on intrusion detection methods that deploy time series analysis or stream-based solutions. For example, Viinikka et al. [2009] deployed time series models to analyze intrusion alert notifications. Using the method, the researchers proposed a model that filters out low-level security alerts. Liu and Kim [2010] applied time series decomposition to detect DDoS attacks. The proposed approach allows real-time anomaly detection and reduces false positives and negatives among attack indicators. In turn, Anton et al. [2018] investigated attacks on Industrial IoT networks and compared the effectiveness of three time series–based algorithms (Matrix Profile, Long Short-Term Memory and Seasonal Autoregressive Integrated Moving Average) on Industrial IoT network attack identification. Viegas et al. [2019] proposed a stream learning classifier to detect network intrusions. The researchers employed a verification method that enhances reliability of the intrusion detection process.

Since our data resemble the shape of waveforms, we cannot apply these standard classifiers directly. Additionally, none of the aforementioned approaches has an associated time component. However, the time taken to detect an attack or breach can significantly affect the damage caused to the system. To that end, we have developed an approach that can capture complex shapes of waveforms and results in features that reflect dynamics in the data. As a result, our approach can detect IoT attacks in real time. To the best of our knowledge, none of the prior research in the area of IoT attack detection focuses on real-time decision-making.

2.2 Feature Extraction and Engineering in Industrial IoT Contexts

The data in our application resembles a complex waveform [Onuchowska et al. 2018]. Hence, as a first step of our methodology, we need to extract the important features of that waveform. Feature extraction [Maglaras & Jiang 2014]—along with data cleaning, fusion, and selection and transformation techniques—is part of the data pre-processing stage [Jiang & Yasakethu 2013] as presented in Table 1.

Tsang & Kwong [2005] proposed the Ant Colony Clustering Model (ACCM) supported by four feature extraction algorithms (PCA, Infomax ICA, Extended Infomax ICA, and FastICA) to detect known or unseen intrusion attacks on network infrastructure. Sayegh et al. [2014] extracted features such as timestamp recovery and field data from SCADA packets. Gai et al. [2017] proposed Monte Carlo Cyber Feature Extraction (MC2FE) to identify cyber incidents. In turn, Almalawi et al. [2014] deployed a fixed clustering technique to automatically identify consistent and inconsistent states of SCADA. Finally, Ahmadi et al. [2016] used N-gram, Metadata, Entropy, Image Representation, and String Length features for the selection and analysis of malware samples.

Table 1. Selected Literature on Feature Extraction in Industrial IoT Contexts

Year	Author	Context	Feature extraction method
2016	Ahmadi et al.	Malware samples' classification and analysis	N-gram, Metadata, Entropy, Image Representation and String Length
2017	Gai et al. [2017]	Creation of a secure cyber incident analytics framework	Monte Carlo Cyber Feature Extraction (MC2FE)
2014	Almalawi et al.	Automatic identification of consistent and inconsistent states of SCADA	Fixed-width clustering technique
2014	Sayegh et al.	Detection of multiple types of attacks using IDs on SCADA protocols	Timestamp recovery and field data from SCADA packets.
2005	Tsang & Kwong [2005]	Known or unseen intrusion attacks detection	PCA, Infomax ICA, Extended Infomax ICA and FastICA

In this research, we investigate a novel approach for feature extraction via FSA. FSA is a subset of the emerging field of functional data analysis (FDA) and has the appeal that it allows for quantification of the dynamics in the data.

2.3 Functional Shape Analysis (FSA)

FSA allows for the extraction of dynamic information in the waveform via dimension reduction techniques [Foutz & Jank 2010]. FSA first reduces noise in the data by rendering the waveforms into smooth continuous functions. Computing the first or second derivatives of the smooth continuous function allows us to quantify its dynamics. Functional principal component analysis can capture key elements of the smooth waveform (and its derivatives) as presented in Table 2. FSA is part of the broader area of FDA commonly applied in dynamic system contexts. For example, Wang et al. [2008] deployed FSA to analyze an auction's price evolution and price dynamics. The researchers proposed a model that predicts an "in progress" auction price and, depending on the new data arriving in the dataset, accommodates the new information so that the prediction can be amended accordingly. In turn, Bapna et al. [2008] used FSA to investigate characteristics of electronic markets and to explain the process of price formation. Yao et al. [2005] used FSA for irregularly spaced longitudinal data to predict individual smooth trajectories, for which a limited amount of measurements is available. Historically, this approach has been used effectively to capture the dynamic information about time series data and help draw inferences about the rate of change of such data [Jank & Zhang 2011] For example, Jank & Zhang [2011] proposed an automated and data-driven bidding strategy that consists of a dynamic forward-looking model for price in competing auctions and a bidding framework that looks to identify the best bid amount. In turn, Gao et al. [2012] developed a Markov chain model to capture the characteristics of tentative customer orders' attributes, which change dynamically over time. Golrezaei et al. [2014] investigated personalization of product assortments for arriving customers and proposed an algorithm for real-time optimization of personalized assortments. Choi et al. [2014] proposed an intelligent forecasting algorithm to analyze real-time sales levels for fast fashion items under the given time and data constraints.

Extracting features from functional data is a non-trivial task since functional data is typically based on continuous, high-dimensional objects. To overcome this challenge, one often employs functional principal component analysis (FPCA), which is a continuous version of traditional principal component analysis (PCA). FPCA is especially useful in drawing inferences from datasets with a high number of interrelated variables because it can help reduce data dimensionality while retaining a high amount of the variation [Ullah & Finch 2013]. Locantore et al. [1999] deployed PCA to extract ophthalmology-related feature vectors and explore corneal topography. Kneip &

Table 2. Selected Literature on Feature Extraction using FSA Techniques

Year	Author	Context	FSA method used
2019	Hasan et al.	Attack and anomaly detection on the Internet of Things (IoT) infrastructure	Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN)
2017	Shi et al.	Spoofing detection in WiFi networks	Support Vector Machine (SVM)
2017	Domb et al.	Real-time anomaly detection in IoT network intrusions	Random Forest (RF)
2017	Meidan et al.	Identification of IoT device connected to a network	Random Forest (RF), GBM, XGBoost
2006	Ferraty & Vieu	Classification of functional data	Functional PCA
2006	Vines et al.	Measuring Tension judgment in music	Functional linear models
2005	Croux & Ruiz-Gazen	Theoretical study	Projection pursuit (PP)–based PCA
2005	Viviani et al.	The analysis of functional magnetic resonance imaging (fMRI) data	Functional PCA
2002	Ratcliffe et al.	Analysis of fetal heart rates	Functional logistic regression model
2001	Kneip & Utikal	Analysis of the underlying population densities	Functional PCA
1999	Locantore et al.	Corneal topography measurement	PCA

Utikal [2001] used FPCA to characterize density families based on the yearly cross-sectional samples of British households. Croux & Ruiz-Gazen [2005] proposed a projection pursuit (PP)–based principal components estimator as a completion of a theoretical study. Viviani et al. [2005] used an FPCA method to extract and analyze data generated by functional magnetic resonance imaging (fMRI). Finally, Ferraty & Vieu [2006] discussed classification of functional data using FPCA.

3 RESEARCH GAPS AND QUESTIONS

Our literature review helped identify several research gaps. First, multiple studies have focused on building predictive algorithms to classify normal and attack behavior in Industrial IoT networks [Cervantes et al. 2015, Fan et al. 2004, Hodo et al. 2016, Rajasegarar et al. 2014]. However, we observed there has been little or no investigation on feature representation for IoT network data rooted in FSA. Consequently, we noticed that systematic comparison of prevailing feature representation in the IoT context has not been conducted in any of the past studies. Finally, in the past literature, little or no emphasis has been placed on the investigation of efficiency-complexity trade-off monitoring in the context of model selection. We further elaborate each of these research gaps in the following subsections.

FSA-based feature representation techniques have been employed extensively to augment the performance of predicting algorithms in the fields of health care and finance [Bapna et al. 2008, Gao et al. 2012]. FSA has been found to improve prediction performance of classification algorithms when applied to complex data [Ullah & Finch 2013, Locantore et al. 1999]. Given the high complexity of the data generated by sensor readings in Industrial IoT networks, FSA and related feature representation techniques may be helpful in detecting Industrial IoT system attacks.

To the best of our knowledge, a systematic comparison of feature representation techniques for detecting Industrial IoT attacks has not been performed. However, in fields such as image

processing [Yang et al. 2004, Ping Tian 2013], financial decision-making [Schumaker & Chen 2009, Hagenau et al. 2013, Zhang et al. 2019] and health sciences [Suk et al. 2014, Suk et al. 2015, Ahmad et al. 2008, Da Silva et al. 2011] such approaches have helped improve the performance of existing algorithms and lead to creation of new ones. Given the similarity in data complexity, an approach that systematically compares competing feature representation techniques can enhance Industrial IoT attack detection.

Past studies do not focus on investigating the efficiency-complexity trade-off in the context of Industrial IoT attack detection. In other contexts, researchers have studied the speed-accuracy trade-off [Huang et al. 2017, Uyttenhove & Steyaert 2002, MacKenzie & Isokoski 2008], which compares change in performance based on time dedicated for decision-making. Efficiency refers to the time allotted to the approach to detecting attacks [Huang et al. 2017], while complexity refers to the challenge of interpreting the approach's mechanisms [MacKenzie & Isokoski 2008].

As discussed by Maddox [2016] and McLeod and Dolezel [2018], the longer it takes to detect an Industrial IoT attack, the greater the total damage caused. Consequently, there is a need for an approach that can swiftly detect an attack on an Industrial IoT system. On the other hand, complexity of the detecting algorithm has critical managerial implications as inference difficulties may lead to complications in addressing the damage to the system.

In order to investigate these research problems, we propose the following research questions:

(1) How can we efficiently detect Industrial IoT attacks by employing features extracted via FSA from the IoT sensor readings?
(2) How can we develop an approach for systematic comparison of feature representation techniques in Industrial IoT attack detection?
(3) How can we develop an efficiency-complexity trade-off that can be used to compare and contrast feature representation techniques?

4 RESEARCH TESTBED AND DATA DESCRIPTION

To analyze the problem of anomaly detection in a network of IoT devices, we investigate the water quality data collected over the lab-controlled series of attacks. These attacks were performed on a SWaT system located in the Center for Research in Cyber Security at Singapore University of Technology and Design [Goh et al. 2016]. The SWaT system is a scaled-down version of an industrial water plant and is managed by a SCADA workstation supplied by a set of Programmable Logic Controllers and actuators as well as IoT sensors. The system is divided into six interconnected parts. Each is responsible for subsequent water filtration activities. The system can produce up to five gallons of filtered water using reverse osmosis and ultrafiltration techniques. The system consists of 54 IoT sensors, each of which collects the data from the environment on a second-by-second basis. The experiment was conducted over 11 days. The system was not attacked for the first seven days. We refer to the initial period as the "normal period of operation" and to the corresponding dataset of sensor readings as the "normal dataset." During the remaining days, simulated cyber-attacks were conducted; we refer to that data as the "attack dataset." Each attack lasted several minutes and affected some or all of the six parts of the system, as demonstrated in Figure 1. The normal dataset consists of 54 variables (each of which is an IoT sensor) and records their measurement readings for 496,800 seconds. The attack dataset records the readings of the same 54 IoT sensors for 449,919 seconds. Each row in the data captures sensor readings for all of the sensors. Of these 54 sensors (from different parts of the system), we focus on the four capturing water quality in the most advanced part of the system. The selected sensors in P5, referred to as analyzers, whose role is to monitor pre-defined water quality features, are as follows [Goh et al. 2016]:

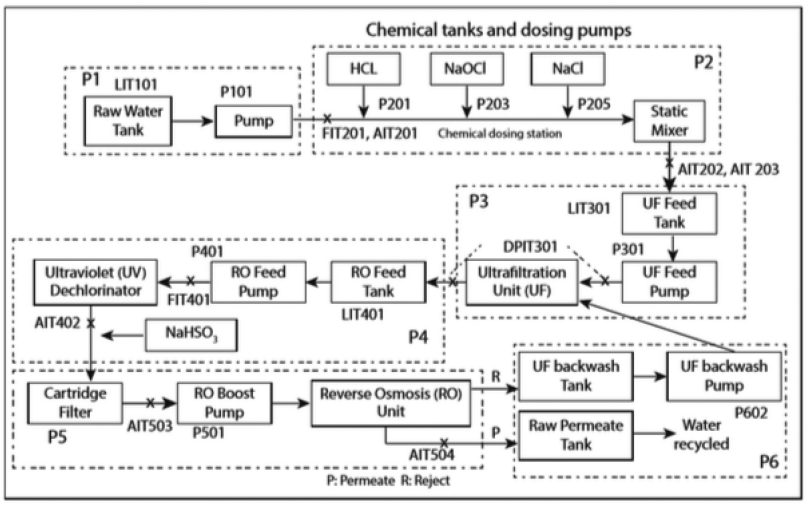

Fig. 1. SWaT testbed process overview [Goh et al. 2016].

- pH level analyzer (AIT501)—indicates the pH level of the inlet water and controls the level of HCl by managing HCl pump operation. Typical operating range for a pH level is defined as between 6 and 9.
- Sodium hypochlorite level analyzer (AIT502)—indicates residual chlorine level of inlet water, triggers the sodium bisulphite dosing pump operation.
- Sodium chloride level analyzers (AIT503 and AIT504)—provide conductivity indication of the inlet water, control the NaCl dosing pump operation.

4.1 Description of Attacks

A total of 36 attacks, on the six parts of the SWaT system, were conducted during the experiment. Attacks could either be physical or cyber. Physical attacks were typically simpler in nature, such as switching on or off a pump in the system. Cyberattacks had significantly higher complexity—for example, changing the input for a sensor by a subtle margin, leading to a decline in the water quality. The attacks were divided into four categories:

- Single-Stage Single-Point—attack on one point in the Cyber Physical System (CPS)
- Single-Stage Multi-Point—attack on multiple points on one stage
- Multi-Stage Single-Point—attack on single point on multiple stages
- Multi-Stage Multi-Point—attack on multiple points that span through multiple stages.

Each sensor presented in Table 3, was affected by some or all attacks. AIT503 was the target of eight different attacks, which is the most among all sensors. Other sensors experienced no more than four attacks. Table 4 summarizes the distribution of the number of attacks per category.

5 RESEARCH APPROACH

As stated earlier, past research has looked at employing different machine learning techniques (both unsupervised and supervised) for anomaly detection in Industrial IoT networks. Our research presents an alternative approach that augments the performance of machine learning techniques. To that end, we propose FSA to develop a novel, dynamic, early detection mechanism for IoT attacks. We employ FSA to extract functional and non-functional features from data, which can

Table 3. Sensor Functional Descriptions [adapted from **Goh et al. 2016**]

Analyzer (Sensor)	Description	Operating Range
AIT501	RO pH analyzer; measures HCl level	6–9
AIT502	RO feed Oxidation-Reduction Potential (ORP) analyzer; measures NaCl level	100–250 mV
AIT503	RO feed conductivity analyzer; measures NaCl level	200–300 μS/cm
AIT504	RO permeate conductivity analyzer; measures NaCl level	5–10 μS/cm

Table 4. Number of Attacks Per Category [Goh et al. 2016]

Category of Attack	Number of Attacks
Single-Stage Single-Point	26
Single-Stage Multi-Point	4
Multi-Stage Single-Point	2
Multi-Stage Multi-Point	4

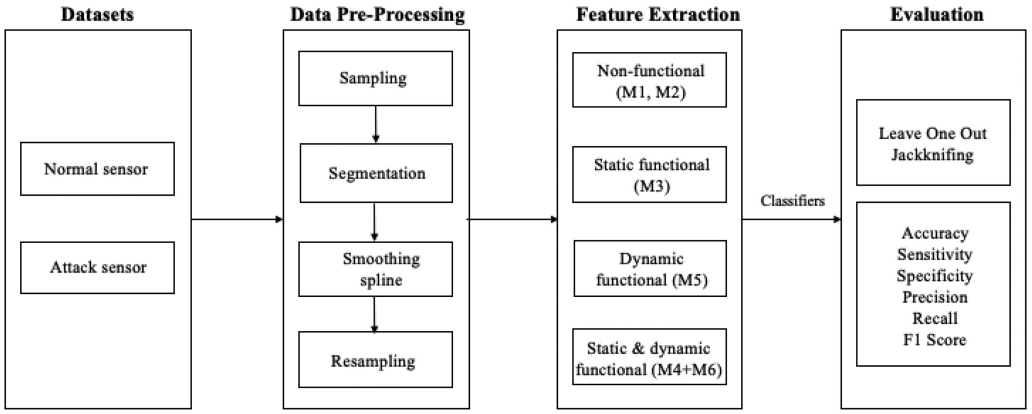

Fig. 2. The proposed FSA-based approach for attack detection.

help improve the performance of classifiers. Figure 2 illustrates the four major components of our research approach. Further details are provided below.

While developing such an approach, we face a two-fold issue. First, we need to establish whether the attacks have any impact on the sensor readings at all. Since a cyberattack manipulates the functioning of an Industrial IoT device and disrupts the dynamic equilibrium of the interconnected devices, we first need to check whether during an attack the waveform generated by a sensor is different from the waveform under normal conditions. To investigate this, we employ a sign-pair Welch t-test (Table 5). The difference in means (i.e., mean sensor reading under normal operations minus mean sensor reading under attack) is statistically significant at $\alpha = 0.05$ for all of the analyzers.

The results in Table 5 suggest that the waveform generated by a sensor while under attack is different from the waveform under normal conditions. While this is encouraging, our second challenge is to develop an approach that can identify such a difference as quickly as possible. This is especially challenging in our data, as the nature of the readings varies from one analyzer to another. Figure 3 presents an example of such variability.

Table 5. Results of Welch t-test for All Analyzers (Attack vs. Normal)

Analyzer (Sensor)	t-statistic	p-value	Mean of the differences
AIT501	56.059	<2.2e−16	0.08
AIT503	21.383	<2.2e−16	2.09
AIT504	−12.146	<2.2e−16	−1.99
AIT502	44.992	<2.2e−16	13.89

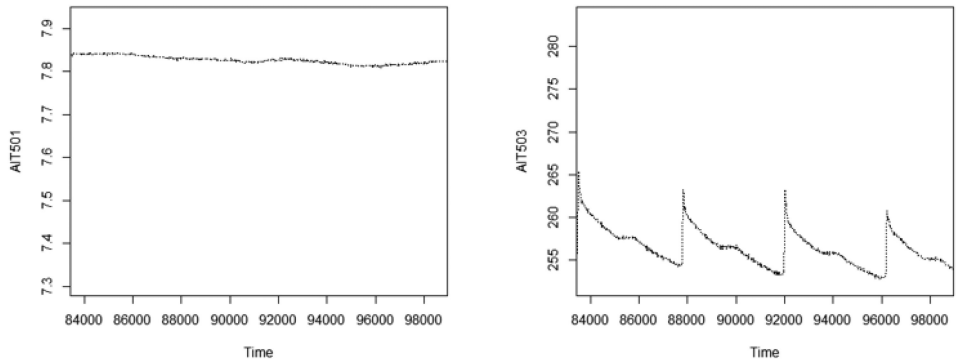

Fig. 3. Water quality indicators output: AIT501 (left) and AIT503 (right).

The left panel of Figure 3 shows that the data from one of the system's sensors, AIT501, resemble an almost straight-line, linear pattern. For such linear patterns, we observe a distinct disruption (i.e., a vertical shift) in reading values whenever an attack takes place. The sign pair *t*-test established a statistically significant difference in the mean sensor readings while under attack versus normal operation. This is different for other sensors. The right panel of Figure 3 shows data from another system sensor, AIT503. Here, we can see that the readings are non-linear, spiky, and resemble tall waves. As we will demonstrate later, this situation proves especially challenging to distinguish between sensor readings during normal operations versus attacks.

6 RESEARCH DESIGN

We divide the data into two parts. The first part captures sensor readings under normal operations (i.e., no attack). The second captures sensor readings under attack. Our goal is to decide as early as possible whether an attack has taken place. To investigate the impact of time on decision-making, we select sensor time series of different lengths from each part. Longer time series correspond to longer wait-times for decision-making. For instance, while a 200-second length allows for more information (potentially leading to more accurate predictions), it also requires waiting for at least 200 seconds before deciding about a potential attack. On the other hand, selecting a shorter time series (e.g., 20 seconds) enables faster decision-making. To this end, we vary the length (i.e., segments) of our time series from 20 to 200 seconds. The segments are created by slicing time periods of varying length (the length of the period determines segment size) from the overall data. We have used down-sampling mechanisms to capture these segments. Using these segments, we construct a testbed of data under both normal and attack operations that also allows for an assessment of time necessary for attack decisions. To ensure robustness, we went back and performed up-sampling mechanisms using bootstrapping and smote analysis to capture the time segments and observe that the results remain consistent.

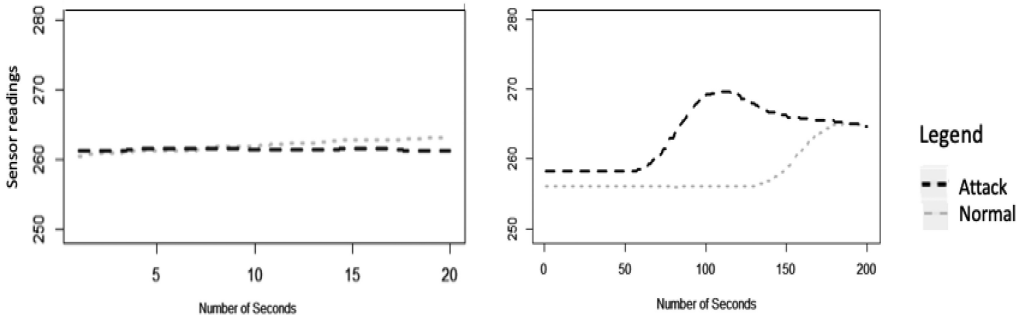

Fig. 4. AIT503 segments of size 20 seconds (left) and 200 seconds (right).

Figure 4 illustrates the different data segments. The gray dotted lines correspond to normal segments. The black dashed lines correspond to attack segments. The left panel shows an example of a 20-second segment. The right panel shows a 200-second segment example. Visually, it is harder to distinguish an attack from normal operation in the shorter, 20-second segment. However, if our goal is to make predictions as early as possible, then the 20-second segment is our target. Our results indicate that FSA can extract dynamic features from the shorter, 20-second segment, which allows for superior prediction accuracy compared with simpler, non-functional approaches.

7 A DYNAMIC FUNCTIONAL CLASSIFICATION MODEL FOR ATTACK PREDICTION

The first step in FSA is smoothing all data segments. Smoothing removes noise and allows us to compute dynamics in the form of the first or second derivatives of the smooth data segments[1] [Foutz & Jank 2010]. Next, we apply FPCA to each smooth segment as well as to its derivatives. This generates a list of features derived from principal components of the position (i.e., the main smooth curve) of the waveform, its velocity (i.e., first derivative), and acceleration (i.e., second derivative). We then select the most relevant features (using model selection techniques) for the classifier.

Figure 5 illustrates the result of FPCA for the position (i.e., the main smooth curve) of the waveform. It shows the first three functional principal component curves for all of the combined attack and normal 200-second segments. The principal component curves for other segments are qualitatively similar.

Figure 5 shows that PC1 is an almost constant straight line. This suggests that PC1 captures the difference in level between different waveforms. PC2 resembles an almost linearly decreasing line. This implies that PC2 captures the change in slope between different waveforms. On the other hand, PC3 captures the most non-linear element of the waveform. It resembles an inverse u-shape, which suggests that it measures the waves' peak (or valley).

Figure 6 illustrates the application of FPCA on the examples in Figure 4. In particular, it shows the functional principal component scores[2] (FPCS) for the normal and attack 20-second segments (first row) and the normal and attack 200-second segments (second row). In the top left graph of Figure 6, we can see the FPCS of the normal 20-second segment. The first bar of that graph corresponds to PC1 and shows that this segment has a positive score. However, it does not score as high on PC1 as the corresponding attack segment in the corresponding graph on the top right.

[1]We also refer to the smooth data segment as the main smooth curve in order to distinguish it from the curves of the first and second derivatives taken from the main smooth curve.

[2]The functional principal component score is computed as the inner product of the smoothed segment with the corresponding functional principal component curve.

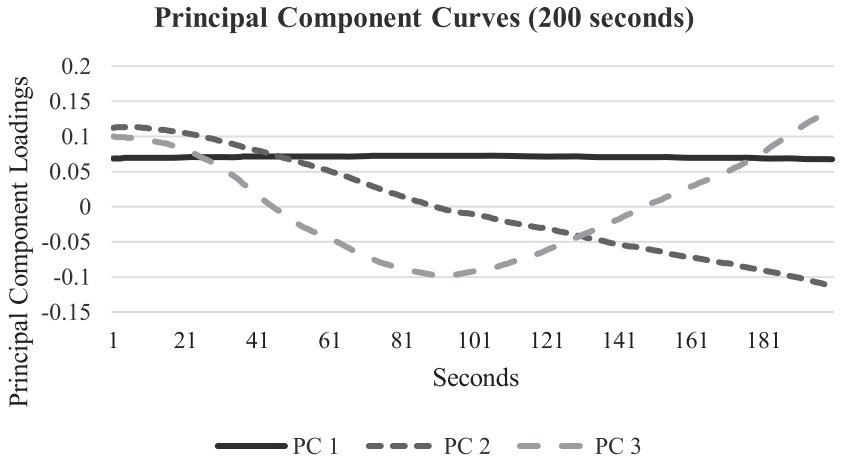

Fig. 5. The first 3 principal component curves for 200-second segments.

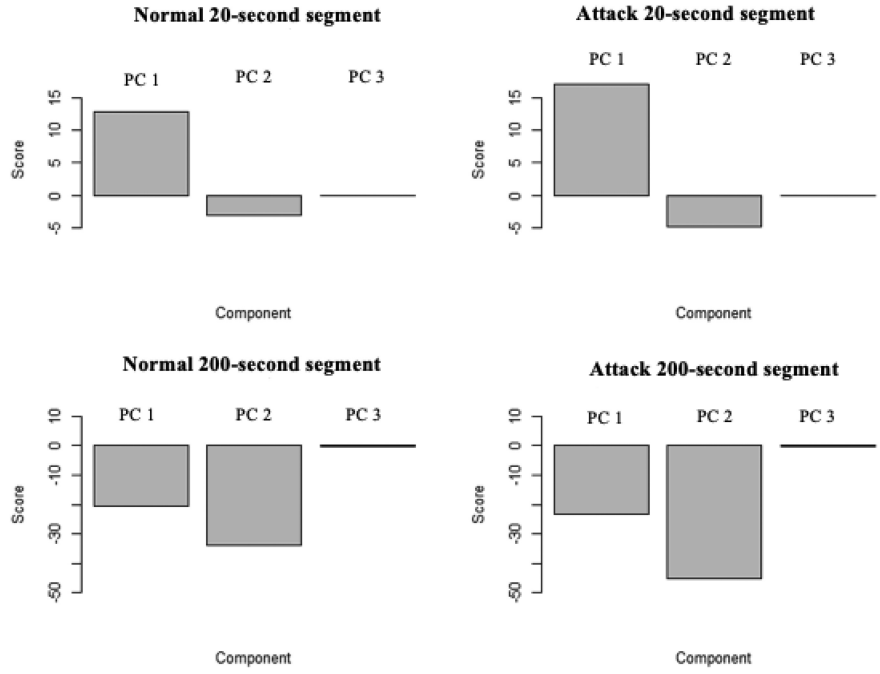

Fig. 6. The principal component scores of the normal and attack segment referenced in Figure 4.

We have seen in Figure 5 that PC1 captures the level of the curve; thus, we can conclude that the attack segment distinguishes itself from the normal segment by an upwards shift in the level. The second bar of the top left graph in Figure 6 corresponds to PC2; recall from Figure 5 that PC2 captures the slope of the waveform. The normal 20-second segment in the top left panel has a lower score (in absolute value) on PC2 compared with the corresponding attack segment in the top right panel. This suggests that the waveform under attack features a steeper slope. The graph's third bar corresponds to PC3, which, according to Figure 5, captures the peak of the waveform. The

attack segment in the top right panel scores slightly higher on PC3 (the difference in corresponding scores is always 1 or more) compared with the normal segment in the top left panel. This suggests that the waveform under attack has a higher peak compared with normal operations. All in all, the FPCS allows us to distinguish each waveform by its level, slope, and peak in a concise and data-driven fashion.

We apply FPCA to the main smooth curve (i.e., the position of the waveform) and its derivatives. We focus only on the first derivative because our investigations show that no additional predictive power can be gained from the second derivative. For each curve, we consider only the first three PCs, as they capture approximately 93% of the total variation across all segments, both for the main smooth curves and their first derivatives. Using these PC scores, we design four different dynamic functional classification models:

- **Functional Position Model:** This model uses only characterizations of the position of the waveform. That is, it uses PCs based only on the main smooth curve (and ignores all of the information related to the derivatives of that curve). In other words, similar to our findings from Figure 5, it uses only information about the level, the slope, and the peak of the waveform.
- **Functional Position and Velocity Model:** Similar to the functional position model, this model uses information about the level, slope, and peak of the waveform (i.e., the first three PCs of the main smooth curve). In addition, it uses the first three PCs of the velocity curve. As such, it not only captures the position of the waveform but also the speed at which it is changing.
- **Functional Velocity Model:** This model uses only the first three PCs of the velocity curve. That is, it ignores information about the position of the waveform and incorporates only the speed at which it is changing.
- **Lean Functional Position and Velocity Model:** This model is similar to the Functional Position and Velocity Model but considers only the first two PCs, both for the position and for the velocity. It is derived using variable selection and is done to ensure that there is no overfitting.

These four functional models differ only with reference to the features engineered from the waveforms. What they all have in common is that we will use all four models to predict a binary response. That is, we will use all four models to predict a binary response variable that has the value 1 if an attack is present (and 0 otherwise). This allows us to investigate different classification algorithms that link the inputs (i.e., features engineered from the waveforms) to the output (i.e., binary response). We employ a series of different classification algorithms, such as Logistic Regression [Ratcliffe et al. 2002, Hasan et al. 2019], Support Vector Classifiers [Shi et al. 2017, Hasan et al. 2019], K-Nearest Neighbors [Vines et al. 2006], and Random Forests [Domb et al. 2017, Meidan et al. 2017, and Hasan et al. 2019]. We compare our four functional models with two benchmark models that do not use any functional features. In addition, using runtime analysis, we compare and contrast the runtimes of functional and non-functional models. Using Welch's *t*-test, we observe that there is no statistically significant difference between the runtimes of the two models.

We train each of the resulting (4 + 2 = 6) models separately for each of the different data segments. Our data segments range from 20 seconds to 200 seconds in 20-second increments, resulting in 10 different data segments. After training each model on each segment, we use leave-one-out jack-knifing for validation. We evaluate the results using a confusion matrix of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Each is used to calculate performance metrics such as accuracy, sensitivity (i.e., recall), specificity (i.e., precision), and F1 score.

Table 6. Methods of Feature Extraction

Category	Method	Features Extracted	Complexity of Feature Extraction
Non-functional	M1	Slope and intercept of the segments	N2
	M2	The reading values from the segments	N
Functional	M3	The first three PCs of segments	N3
	M4	The first three PCs and first three velocity components of segments	N3
	M5	The first three velocity components of segments	N3
	M6	The first two PCs and first two velocity components of segments	N3

They are computed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \; Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN}, \; F_1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

8 RESULTS

We compare our four functional models with two models based on non-functional features. We refer to these six models as M1 though M6:

Slope and Intercept Model (M1): As each data segment represents a vector of analyzer readings, we fit a linear regression model with the readings as the Y- (or response) variable and time as the X- (or predictor) variable. The resulting classifier (which we refer to as M1) uses the estimated slope and intercept as feature variables. That is, it uses the slope and intercept as inputs in the chosen classifier algorithm (e.g., a logistic regression model) where the output is the binary variable of whether or not there is an attack. The rationale for this model is that it is similar in nature to the models based on the principal components but restricts itself to linear features. In other words, similar to the functional position model, it captures level (via the intercept) and slope. It does not capture any non-linear features such as the peak of the waveform. Therefore, M1 is a simple version of the more advanced models in that it restricts itself to linear features only. Vines et al. [2006] used a similar approach to understand tension judgment in music.

Classic Classification Model (M2): While M1 uses the data segments to derive (linear) features of the waveform, M2 uses the data segment directly (without extracting any features). That is, rather than using the analyzer readings to estimate their slope and intercept, we use the readings directly in a classifier. Ratcliffe et al. [2002] used a similar approach to study fetal heart rates.

Functional Position Model (M3): We refer to the functional position model (which only uses the first three PC scores of the main smooth curve) as M3.

Functional Position and Velocity Model (M4): We refer to the functional model that uses the first three PCs of the main smooth curve and the first three PCs of the velocity curve as M4.

Functional Velocity Model (M5): We refer to the model that uses only the first three PCs of the velocity curve as M5.

Lean Functional Position and Velocity Model (M6): The functional model that uses only the first two PCs (both of the main smooth curve and the velocity curve) is referred to as M6.

Table 6 summarizes each category and method. We also summarize the complexity of the feature extraction.

Table 7. Average F1 Score Across M1 to M6 by Segment Size (for AIT503)

Segment Size	20	40	60	80	100	120	140	160	180	200
RF	0.71	0.73	0.73	0.73	0.74	0.73	0.73	0.73	0.74	0.75
SVC	0.71	0.71	0.72	0.72	0.72	0.72	0.71	0.72	0.73	0.73
LR	0.7	0.7	0.71	0.71	0.72	0.72	0.72	0.73	0.72	0.73
KNN	0.69	0.7	0.71	0.7	0.71	0.72	0.72	0.71	0.72	0.74

As mentioned earlier, we train classifiers using each of the six models on data segments of varying length. Recall that shorter data segments contain less information about the waveform but enable faster decision-making. For each model and data segment, we perform leave-one-out jack-knifing and compute the prediction performance in terms of accuracy, precision, sensitivity, specificity, recall, and F1.

We compare the performance of all six models using the four classification algorithms (Logistic Regression [LR], Support Vector Machines [SVM], K-Nearest Neighbor [KNN], and Random Forest [RF]). We find that, on average, models using RF outperform the other three algorithms (Table 7 presents the average F1 score; the results are similar for other performance measures). Therefore, we present detailed results using RF only. Information about the performance of other algorithms (LR, SVM, and NN) can be found in Appendix A (Table 9, Table 10, and Table 11).

Table 8 shows detailed results for each individual model (using only RF). First, we observe that models perform better on the longer segments. For instance, F1 has the largest value for segments of 180 or 200 seconds across all models. This implies that attack prediction accuracy increases by allowing more time and, hence, more information for decision-making. As we reduce the time allotted for decision-making from 200 seconds to 20 seconds, prediction accuracy declines. While the prediction accuracy declines for all 6 models in Table 8, functional models outperform the non-functional ones. For instance, M4's F1 score reduces from 0.78 at 200 seconds to 0.72 at 20 seconds. On the other hand, the F1 score of M1 ranges from 0.73 when working with a 200 second segment size to 0.68 when allotted 20 second information to predict.

In order to give some more insight as to why functional methods may have the ability to outperform non-functional ones, see Figure 7. It shows four different examples of waveforms when the system was under attack. Notice that for the two examples in the top panel, both functional and non-functional methods correctly predicted the attack. However, for the two examples in the bottom panel, only the functional methods correctly predicted the attack. We can see that for the top two examples, the waveform is very simple, with mostly constant and linear features. This is in contrast to the bottom two examples, which have very complex features with many changes in level, slope, and peak. Recall that these changes in level, slope, and peak are captured by the principal component curves in Figure 5. This suggests that functional feature extraction methods are well capable of capturing not only simple, linear features but also that they thrive particularly in situations in which the waveform has complex, non-linear features with sudden changes in the dynamics.

In fact, the variation within the attack segments demonstrated earlier combined with findings from Table 8 suggest that these features become increasingly relevant for the shorter data segments—that is, for situations in which we allow little time for decision-making and, as a result, do not observe much information about the nature of the waveform.

We learn from Table 8 that F1 scores and accuracies increase (across all models) for longer segments. This implies that as decision-making time increases, models predict attacks more accurately. However, we also observe that there is a difference between the performance of the

Table 8. Performance of Functional and Non-functional Methods Across Varying Segment Sizes for AIT 503 Using a Random Forest Classifier

Segment size		20	40	60	80	100	120	140	160	180	200
M1	Accuracy	0.68	0.68	0.68	0.71	0.71	0.71	0.70	0.68	0.74	0.73
	Sensitivity	0.73	0.73	0.77	0.77	0.81	0.77	0.79	0.73	0.89	0.83
	Specificity	0.67	0.67	0.60	0.67	0.62	0.67	0.62	0.67	0.60	0.64
	F1	0.70	0.70	0.67	0.72	0.70	0.72	0.69	0.70	0.72	0.72
M2	Accuracy	0.70	0.71	0.74	0.68	0.71	0.73	0.74	0.73	0.74	0.74
	Sensitivity	0.79	0.77	0.87	0.77	0.83	0.87	0.89	0.85	0.89	0.89
	Specificity	0.62	0.67	0.62	0.60	0.60	0.60	0.60	0.62	0.60	0.60
	F1	0.69	0.72	0.72	0.67	0.69	0.71	0.72	0.72	0.72	0.72
M3	Accuracy	0.73	0.77	0.75	0.77	0.75	0.75	0.75	0.75	0.74	0.77
	Sensitivity	0.85	0.97	0.92	0.97	0.92	0.89	0.89	0.89	0.84	0.94
	Specificity	0.62	0.60	0.64	0.60	0.64	0.62	0.62	0.62	0.65	0.62
	F1	0.72	0.74	0.74	0.74	0.74	0.73	0.73	0.73	0.73	0.75
M4	Accuracy	0.73	0.78	0.75	0.77	0.77	0.76	0.76	0.76	0.76	0.78
	Sensitivity	0.83	1.00	0.88	0.97	0.90	0.90	0.88	0.88	0.84	0.87
	Specificity	0.64	0.60	0.64	0.60	0.65	0.64	0.65	0.65	0.69	0.71
	F1	0.72	0.75	0.74	0.74	0.76	0.74	0.75	0.75	0.76	0.78
M5	Accuracy	0.73	0.74	0.74	0.77	0.78	0.78	0.76	0.76	0.75	0.77
	Sensitivity	0.83	0.82	0.82	0.90	0.90	0.92	0.92	0.88	0.89	0.90
	Specificity	0.64	0.67	0.67	0.65	0.67	0.65	0.62	0.65	0.62	0.65
	F1	0.72	0.74	0.74	0.76	0.77	0.77	0.74	0.75	0.73	0.76
M6	Accuracy	0.72	0.73	0.74	0.73	0.77	0.73	0.76	0.76	0.76	0.76
	Sensitivity	0.85	0.83	0.84	0.83	0.97	0.83	0.90	0.88	0.88	0.86
	Specificity	0.60	0.64	0.65	0.64	0.60	0.64	0.64	0.65	0.65	0.67
	F1	0.70	0.72	0.73	0.72	0.74	0.72	0.74	0.75	0.75	0.76

functional methods and the non-functional ones. Figure 8 demonstrates the changing F1 scores from Table 7 against segment length to further investigate this difference.

Figure 8 shows the performance of our models in predicting attacks for the sensor AIT503. Recall that this sensor's resulting waveform is highly spiky and non-linear. Thus, feature extraction is non-trivial. We can see that the functional models (M3–M6) consistently outperform the non-functional ones (M1 and M2). This, again, re-emphasizes the added value of functional feature engineering methods, especially for challenging, spiky patterns. But, do functional methods also add value for simpler, more linear patterns? We investigate the model performance on sensors that result in less challenging data patterns. To that end, we combine the data from three additional sensors, AIT201, AIT502, and AIT504. Each sensor results in more linear readings. Figure 9 plots the performance of our six different models for these linear sensors.

For example, in Figure 8 for the longest segments (200 or 180 seconds), all six models have improved prediction performance. This, again, suggests that when we allow ourselves more time for decision-making, we obtain more accurate predictions regardless of the choice of model. Unlike Figure 8, however, for all six models in Figure 9, we observe that when segment size is reduced to less than 60 seconds, there is a significant drop in the performance of the non-functional methods. This implies that for data segments of 60 seconds or longer, the simple, linear nature of the data does not result in any prediction difference regardless of whether we use simple non-functional models (M1 or M2) or a more complex approach (functional models M3–M6). However, once we

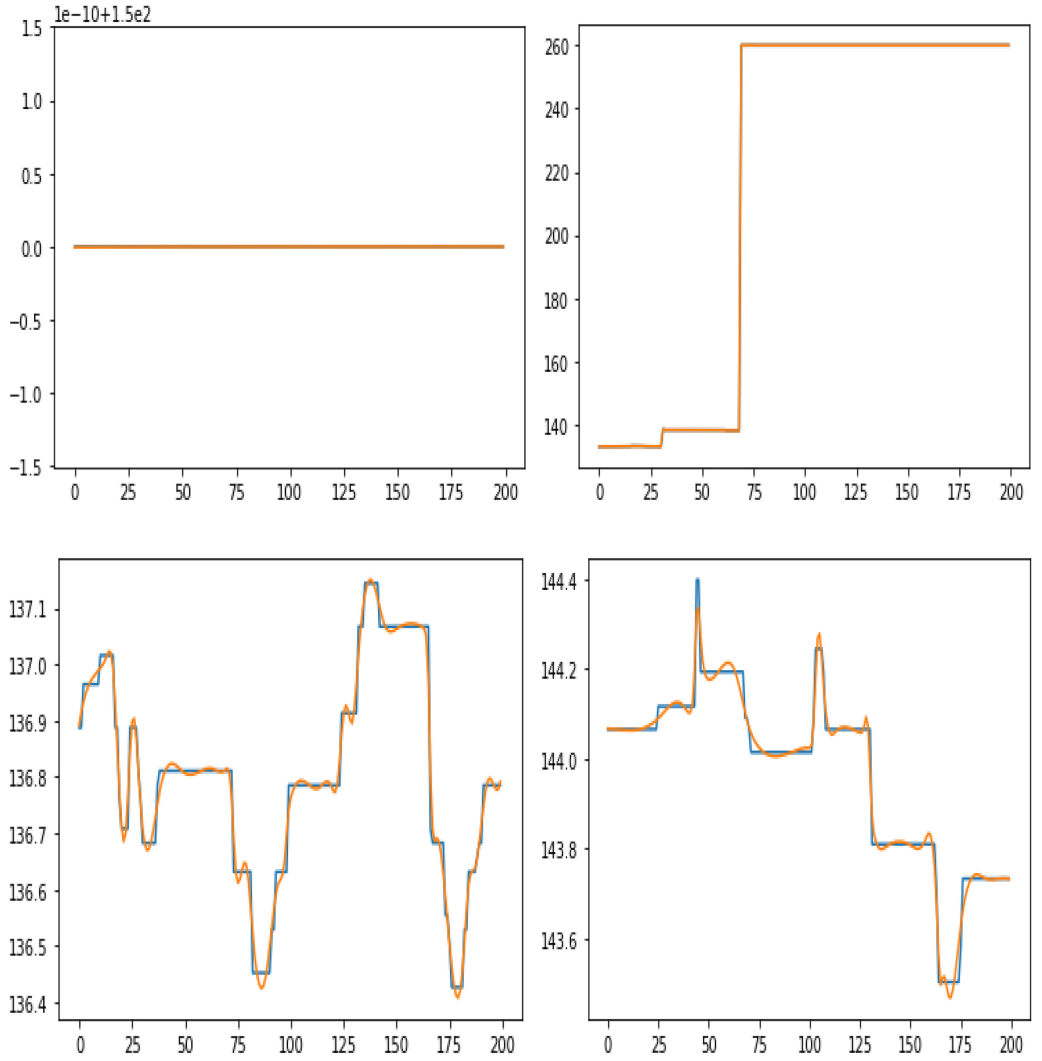

Fig. 7. Attack segments captured correctly by functional methods (M4, M6) but missed by non-functional methods (M1 and M2). The X-axis corresponds to the size of the time segment; the Y-axis corresponds to the sensor readings.

reduce the segment size below 60 seconds in order to make decisions in real (or almost real) time, the results change. In fact, for segments shorter than 60 seconds, non-functional models M1 and M2 exhibit a steeper performance decline than the functional models. This may be attributed to a similar reasoning as exhibited earlier in the section, where we demonstrated that the functional models are able to capture subtle changes in the sensor readings as compared with their non-functional counterparts. This suggests that there is value in using FSA for feature extraction, even for linear data. It also suggests that the dynamic and non-linear elements that FSA is able to capture are especially valuable for short data segments when the focus is on agile, real-time decision-making. Figure 9 suggests that in reduced time for decision-making, such as 20- or 40-second segment size, the functional methods perform nearly 1.5 times more accurately than the non-functional ones.

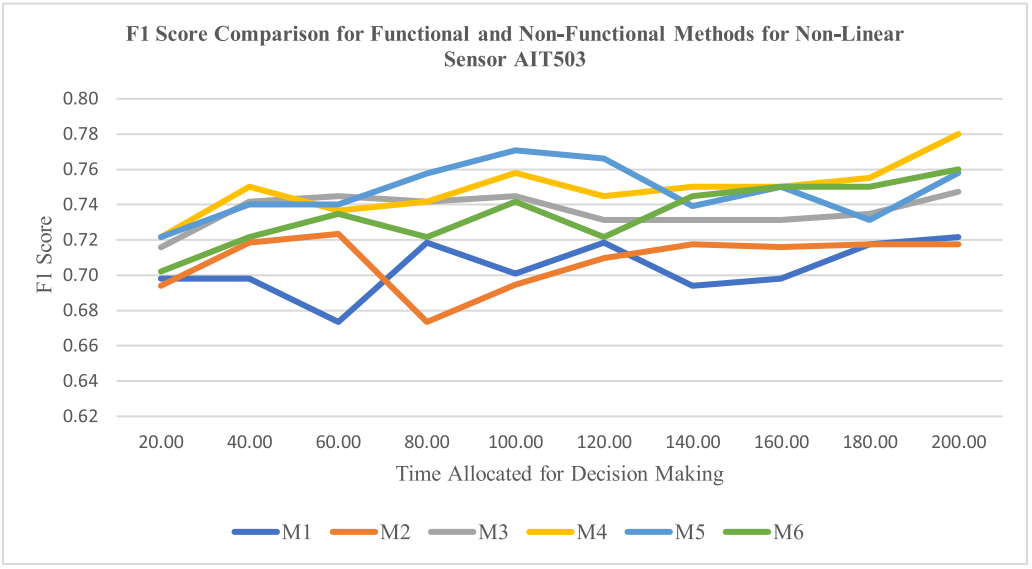

Fig. 8. Performance comparison for AIT503: functional vs. non-functional models.

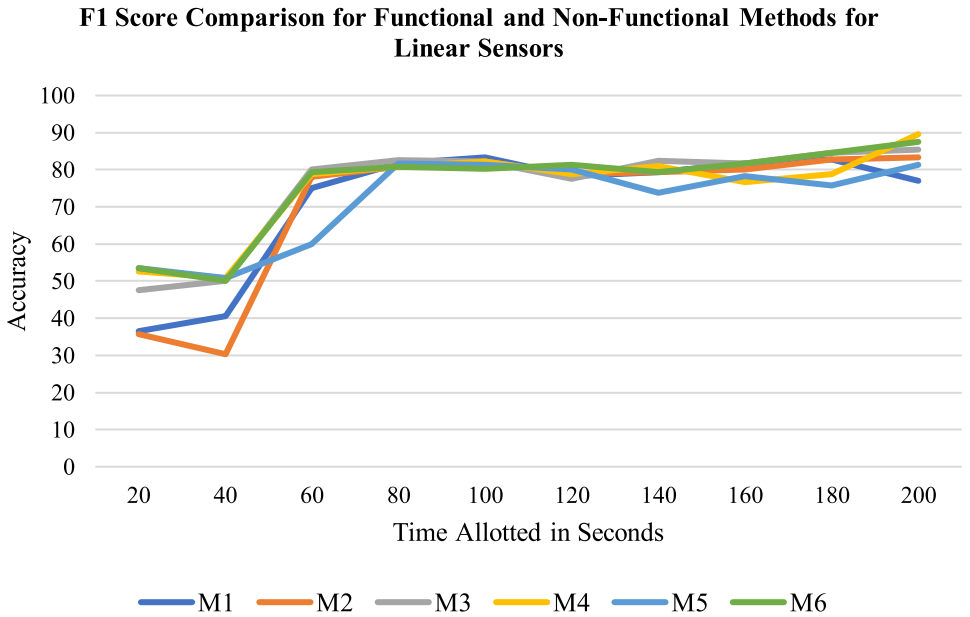

Fig. 9. Performance comparison for linear data sensors: functional versus non-functional methods.

9 EFFICIENCY COMPLEXITY TRADE-OFF

The previous results suggest that when segment size decreases, functional models outperform non-functional ones. Functional methods employ more mathematically complex computations as they extract non-linear features using smoothing and FPCA. We take a closer look at the trade-off between the complexity of the approach and the resulting performance gains by computing the

Fig. 10. Efficiency complexity trade-off for AIT503.

average accuracy of all functional methods and compare it to the average accuracy of all non-functional methods. The results can be seen in Figure 10 and Figure 11.

Figure 10 shows the average accuracy of the functional and non-functional models for the non-linear readings pertaining to sensor AIT503. Figure 11 shows the corresponding average accuracy for the models when accounting for more linear readings pertaining to sensors AIT501, AIT502, and AIT504. Overall, the difference in average accuracy between functional and non-functional models for linear sensor readings depicts a different scenario. When time allotted is 20 or 40 seconds, the functional methods perform better than the simple methods. However, once time allotted is increased, there is minimal benefit from FSA's extra complexity when allowing additional decision-making time. This benefit materializes only for the shortest data segments, that is, in situations for which agile, real-time decision-making is required.

10 BUSINESS AND MANAGERIAL IMPLICATIONS

The results presented earlier have several key implications for Industrial IoT managers and their respective businesses. Our results suggest that functional methods can be very effective in accurately tracking down an IoT attack in real time. They are especially valuable when sensor readings result in highly non-linear and spiky patterns. This suggests that functional methods can be applied to systems in which the measures are highly volatile and when there is little time to detect a deviation from the norm. Ultimately, such insights can enable the effective and efficient development of relevant security controls and protocols (e.g., firewall rules and antivirus updates). In particular, the results of this work can assist systems administrators, systems analysts, and security operations center professionals execute two tasks. First, the insights pertaining to the lead time necessary for attaining selected accuracy performances can hold significant value when

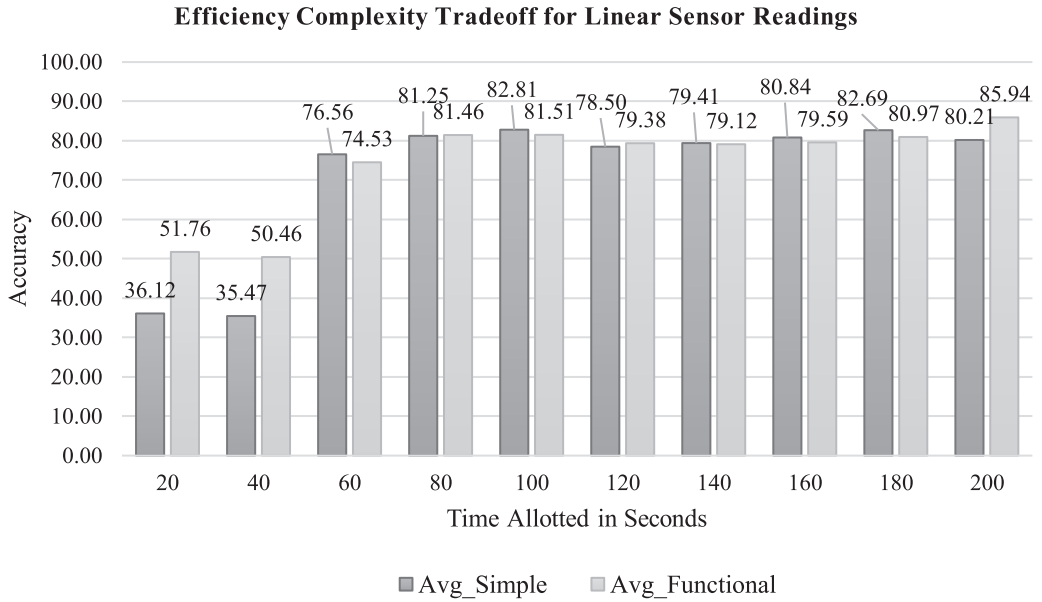

Fig. 11. Efficiency-complexity trade-off for linear sensor readings.

customizing or tuning the settings of the technical controls. Second, the analysis conducted across multiple testbeds can provide managers insight into how algorithms perform in selected variations (e.g., feature settings). This is essential when data contexts may vary across sensor types.

Beyond its application in cybersecurity, the proposed functional methods also have implications for other high-impact and emerging domains. For example, recent years have seen a sharp uptick in the demand for novel motion sensor–based health analytics for senior citizen care. Two emerging application areas in this regard are Activity of Daily Living Recognition (ADLR) and fall risk assessment (i.e., fall prediction and fall detection). The approach and its relevant constituent component (e.g., functional feature extraction) can enable scholars in the health informatics realm to remotely monitor patients recovering from illnesses or the well-being of the elderly with a higher degree of interpretability, efficiency, and effectiveness than those solutions deployed in extant literature.

11 CONCLUSIONS AND FUTURE WORK

Cybersecurity's rapid emergence as one of modern society's grand challenges has resulted in the compromise of numerous critical infrastructures and systems. In this work, we use functional shape analysis to a develop novel set of features for input into state-of-the-art classifiers for real-time cyberattack detection on Industrial IoT devices. We rigorously evaluate all proposed features against well-established benchmark approaches. Our results show three key insights. First, attacked Industrial IoT devices generate waveforms of sensor readings that are statistically significant from devices that are not under attack. Second, more complex functional features outperform those of the non-functional variety. Finally, the proposed functional methods outperform their counterparts when focused on agile, real-time decision-making scenarios. Taken together, these contributions help provide unprecedented attack detection capabilities and decision-making insights for managers of Industrial IoT devices and systems. While our method focuses on industrial IoT devices, it could be readily adapted to other IoT applications, such as smart homes, in order to prevent attacks on appliances such as thermostats or lighting fixtures, where a system overload

could lead to an appliance failure (e.g., compressor overload in an HVAC system or breaker failure in an electric system). In addition to smart homes, similar methodology could prove even more important in the context of health care or cybersecurity systems. In health care, for instance, an attack on a hospital's HVAC system could result in the severe loss of life. However, a hospital's IoT system could go well beyond its HVAC system. Smart hospitals feature wearables, smart pills, smart beds, biosensors, robots, glucose measurement devices, equipment monitoring devices, remote monitoring systems [Maltseva 2018], and much more. All of these interconnected devices are prone to malicious attacks that might be detected with methods similar to the ones developed in this article.

Although the current implementation of our approach is unable to detect the root cause behind process interruptions, there are several promising directions for future work. First, this work can be extended to include wavelet transformation for feature extraction. A wavelet transformation is a tool that uses functions localized in both real and Fourier space. The transformation allows changes only in time extension while keeping the shape constant. Second, future work can look to employing emerging deep learning algorithms to achieve higher attack detection performances. Deep learning is a class of machine learning methods that use multiple layers of non-linear activation functions. Such algorithms can help reduce feature-engineering efforts and draw out additional features beyond those demonstrated in this work. Finally, additional work can look at fusing related datasets (e.g., external IoT data) to provide finer-grained and more comprehensive intelligence about emerging threats toward Industrial IoT devices. Ultimately, each direction can help develop novel cybersecurity analytics to further improve the security posture of critical infrastructures.

APPENDIX A

Table 9. Performance of Functional and Non-functional Methods Across Differing Segment Sizes for AIT503 using k-Nearest Neighbor Classifier (k = 3 was found to be the best)

Segment size		20	40	60	80	100	120	140	160	180	200
M1	Accuracy	0.7	0.69	0.68	0.71	0.73	0.71	0.72	0.71	0.71	0.73
	Sensitivity	0.7778	0.7273	0.7674	0.7954	0.8333	0.8095	0.8293	0.7955	0.7955	0.8043
	Specificity	0.6364	0.6182	0.6	0.6364	0.6364	0.6182	0.6182	0.6364	0.6364	0.6727
	F1	0.7	0.6869	0.6735	0.7071	0.7216	0.701	0.7083	0.7071	0.7071	0.7328
M2	Accuracy	0.7	0.73	0.73	0.71	0.7	0.71	0.72	0.71	0.71	0.72
	Sensitivity	0.7778	0.8043	0.8333	0.8095	0.7907	0.825	0.8293	0.7955	0.7955	0.814
	Specificity	0.6364	0.6727	0.6364	0.6182	0.6182	0.6	0.6182	0.6364	0.6364	0.6364
	F1	0.7	0.7328	0.7216	0.701	0.6939	0.6947	0.7083	0.7071	0.7071	0.7143
M3	Accuracy	0.7	0.71	0.7	0.69	0.7	0.73	0.71	0.72	0.72	0.74
	Sensitivity	0.7778	0.7955	0.7778	0.7273	0.7907	0.8409	0.8095	0.8	0.8	0.8537
	Specificity	0.6364	0.6364	0.6364	0.6182	0.6182	0.6727	0.6182	0.6545	0.6545	0.6364
	F1	0.7	0.7071	0.7	0.6869	0.6939	0.7475	0.701	0.72	0.72	0.7292
M4	Accuracy	0.69	0.7	0.71	0.69	0.71	0.7	0.7	0.74	0.74	0.78
	Sensitivity	0.7609	0.7907	0.7955	0.7608	0.76	0.7451	0.8085	0.8372	0.8537	0.8667
	Specificity	0.6364	0.6182	0.6364	0.6364	0.6909	0.6909	0.6909	0.6455	0.6364	0.7091
	F1	0.6931	0.6939	0.7071	0.6931	0.7238	0.717	0.7451	0.7347	0.7292	0.78
M5	Accuracy	0.7	0.71	0.73	0.73	0.75	0.75	0.74	0.75	0.75	0.76
	Sensitivity	0.7778	0.825	0.85	0.85	0.8571	0.875	0.8537	0.7955	0.8571	0.9189
	Specificity	0.6364	0.6	0.6182	0.6182	0.6545	0.6364	0.6364	0.6364	0.6545	0.6182
	F1	0.7	0.6947	0.7158	0.7158	0.7071	0.7368	0.7292	0.7071	0.7423	0.7391
M6	Accuracy	0.68	0.72	0.73	0.72	0.72	0.75	0.72	0.73	0.75	0.76
	Sensitivity	0.7556	0.814	0.8333	0.814	0.814	0.8409	0.814	0.8182	0.875	0.8604
	Specificity	0.6182	0.6364	0.6364	0.6364	0.6364	0.6727	0.6364	0.6545	0.6364	0.6727
	F1	0.68	0.7143	0.7216	0.7143	0.7143	0.7475	0.7143	0.7273	0.7368	0.7551

Table 10. Performance of Functional and Non-functional Methods across Differing Segment Sizes for AIT503 Using Logistic Regression Classifier

Segment size		20	40	60	80	100	120	140	160	180	200
M1	Accuracy	0.68	0.67	0.665	0.68	0.67	0.7	0.7	0.7	0.7	0.7
	Sensitivity	0.9	0.8	0.9	0.9	0.9	0.9	0.9	1	1	1
	Specificity	0.5636	0.6018	0.5818	0.5818	0.5818	0.5818	0.5818	0.5818	0.5818	0.5818
	F1	0.66	0.6656	0.67	0.67	0.67	0.67	0.67	0.68	0.68	0.68
M2	Accuracy	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.76	0.76	0.76
	Sensitivity	1	1	1	1	1	1	1	1	1	1
	Specificity	0.5455	0.5455	0.5455	0.5455	0.5455	0.5636	0.5636	0.5636	0.5636	0.5636
	F1	0.7059	0.7059	0.7059	0.7059	0.7059	0.7209	0.7209	0.7209	0.7209	0.7209
M3	Accuracy	0.75	0.75	0.75	0.75	0.76	0.77	0.77	0.77	0.77	0.77
	Sensitivity	1	1	1	1	1	1	1	1	1	1
	Specificity	0.5455	0.5455	0.5455	0.5455	0.5636	0.5818	0.5818	0.5818	0.5818	0.5818
	F1	0.7059	0.7059	0.7059	0.7059	0.7209	0.7356	0.7356	0.7356	0.7356	0.7356
M4	Accuracy	0.75	0.75	0.78	0.78	0.77	0.78	0.78	0.78	0.77	0.77
	Sensitivity	1	1	0.9459	1	0.9706	1	1	1	1	1
	Specificity	0.5455	0.5455	0.6	0.6	0.6	0.6	0.6	0.61	0.6	0.6364
	F1	0.7059	0.7059	0.75	0.75	0.7416	0.75	0.75	0.7609	0.74	0.75
M5	Accuracy	0.73	0.74	0.75	0.76	0.77	0.77	0.77	0.77	0.77	0.77
	Sensitivity	0.8889	0.8537	1	1	1	1	1	1	1	1
	Specificity	0.5364	0.5818	0.5455	0.5636	0.5818	0.5818	0.5818	0.5818	0.5818	0.5818
	F1	0.7033	0.7292	0.7059	0.7209	0.7356	0.7356	0.7356	0.7356	0.7356	0.7356
M6	Accuracy	0.76	0.76	0.76	0.76	0.76	0.77	0.77	0.76	0.77	0.77
	Sensitivity	1	1	1	1	1	1	1	1	1	1
	Specificity	0.5636	0.5636	0.5636	0.5636	0.5636	0.5818	0.5818	0.5636	0.5818	0.5818
	F1	0.7209	0.7209	0.7209	0.7209	0.7209	0.7356	0.7356	0.7209	0.7356	0.7356

Table 11. Performance of Functional and Non-functional Methods Across Differing Segment Sizes for AIT503 Using Support Vector Classifier

Segment size		20	40	60	80	100	120	140	160	180	200
M1	Accuracy	0.7	0.71	0.68	0.68	0.68	0.68	0.67	0.68	0.71	0.68
	Sensitivity	0.7907	0.7955	0.7255	0.7255	0.7255	0.7255	0.8	0.9	0.7955	0.7255
	Specificity	0.6182	0.6364	0.6727	0.6727	0.6727	0.6727	0.6018	0.5818	0.6364	0.6727
	F1	0.6939	0.7071	0.6981	0.6981	0.6981	0.6981	0.6656	0.67	0.7071	0.6981
M2	Accuracy	0.7	0.7	0.71	0.72	0.71	0.71	0.71	0.71	0.73	0.73
	Sensitivity	0.7778	0.7907	0.825	0.8293	0.7955	0.7955	0.7955	0.7955	0.8043	0.8043
	Specificity	0.6364	0.6182	0.6	0.6182	0.6364	0.6364	0.6364	0.6364	0.6727	0.6727
	F1	0.7	0.6939	0.6947	0.7083	0.7071	0.7071	0.7071	0.7071	0.7328	0.7328
M3	Accuracy	0.74	0.74	0.74	0.73	0.74	0.74	0.74	0.76	0.76	0.76
	Sensitivity	0.8718	0.8718	0.8718	0.8043	0.8718	0.8718	0.8718	1	1	1
	Specificity	0.6162	0.6162	0.6162	0.6727	0.6162	0.6162	0.6162	0.5636	0.5636	0.5636
	F1	0.7234	0.7234	0.7234	0.7328	0.7234	0.7234	0.7234	0.7209	0.7209	0.7209
M4	Accuracy	0.73	0.73	0.73	0.73	0.74	0.75	0.77	0.75	0.77	0.75
	Sensitivity	0.85	0.85	0.8182	0.8182	0.8718	0.9167	1	0.9167	1	0.9167
	Specificity	0.6182	0.6182	0.6545	0.6545	0.6162	0.6364	0.5818	0.6364	0.5818	0.6364
	F1	0.7158	0.7158	0.7273	0.7273	0.7234	0.7447	0.7356	0.7447	0.7356	0.7447
M5	Accuracy	0.73	0.73	0.75	0.75	0.76	0.73	0.76	0.76	0.76	0.76
	Sensitivity	0.8889	0.8889	0.9167	0.9167	1	0.8182	1	1	1	1
	Specificity	0.5364	0.5364	0.6364	0.6364	0.5636	0.6545	0.5636	0.5636	0.5636	0.5636
	F1	0.7033	0.7033	0.7447	0.7447	0.7209	0.7273	0.7209	0.7209	0.7209	0.7209
M6	Accuracy	0.76	0.75	0.77	0.77	0.77	0.77	0.75	0.77	0.75	0.76
	Sensitivity	1	0.9167	1	1	1	1	0.9167	1	0.9167	1
	Specificity	0.5636	0.6364	0.5818	0.5818	0.5818	0.5818	0.6364	0.5818	0.6364	0.5636
	F1	0.7209	0.7447	0.7356	0.7356	0.7356	0.7356	0.7447	0.7356	0.7447	0.7209

ACKNOWLEDGMENTS

Data for this research has been kindly provided by the iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design.

REFERENCES

Wan Siti Ahmad, Halimatul Munirah Wan, and Mohammad Faizal Ahmad Fauzi. 2008. Comparison of different feature extraction techniques in content-based image retrieval for CT brain images. In *IEEE 10th Workshop on Multimedia Signal Processing, Cairns*. IEEE, 503–508.

Mansour Ahmadi, Dmitry Ulyanov, Stanislav Semenov, Mikhail Trofimov, and Giorgio Giacinto. 2016. Novel feature extraction, selection and fusion for effective malware family classification. In *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy New Orleans*. ACM, 183–194.

Abdulmohsen Almalawi, Xinghuo Yu, Zahir Tari, Adil Fahad, and Ibrahim Khalil. 2014. An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Computers & Security* 46 (2014), 94–110.

Simon Duque Anton, Lia Ahrens, Daniel Fraunholz, and Hans Dieter Schotten. 2018. Time is of the essence: Machine learning-based intrusion detection in industrial time series data. In *IEEE International Conference on Data Mining Workshops (ICDMW'18), Singapore*. IEEE, 1–6.

Tahereh Babaie, Sanjay Chawla, Sebastien Ardon, and Yue Yu. 2014. A unified approach to network anomaly detection. In *IEEE International Conference on Big Data (Big Data'14), Washington DC*. IEEE, 650–655.

Ravi Bapna, Wolfgang Jank, and Galit Shmueli. 2008. Price formation and its dynamics in online auctions. *Decision Support Systems* 44, 3 (2008), 641–656.

Konstantin Bauman, Alexander Tuzhilin, and Ryan Zaczynski. 2017. Using social sensors for detecting emergency events: A case of power outages in the electrical utility industry. *ACM Transactions on Management Information Systems* 8, 2–3 (2017), 1–20.

Cesar Cerrudo. 2015. An emerging US (and world) threat: Cities wide open to cyber-attacks. *Securing Smart Cities* 17 (2015), 137–151.

Christian Cervantes, Diego Poplade, Michele Nogueira, and Aldri Santos. 2015. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In *IFIP/IEEE International Symposium on Integrated Network Management (IM'15), Ottawa*. IEEE, 606–611.

Tsan-Ming Choi, Chi-Leung Hui, Na Liu, Sau-Fun Ng, and Yong Yu. 2014. Fast fashion sales forecasting with limited data and time. *Decision Support Systems* 59 (2014), 84–92.

Christophe Croux and Anne Ruiz-Gazen. 2005. High breakdown estimators for principal components: The projection-pursuit approach revisited. *Journal of Multivariate Analysis* 95, 1 (2005), 206–226.

Sérgio Francisco Da Silva, Marcela Xavier Ribeiro, João do ES Batista Neto, Caetano Traina-Jr, and Agma J. M. Traina. 2011. Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision Support Systems* 51, 4 (2011), 810–820.

Menachem Domb, Elisheva Bonchek-Dokow, and Guy Leshem. 2017. Lightweight adaptive random-forest for IoT rule generation and execution. *Journal of Information Security and Applications* 34 (2017), 218–224.

Dhanapal Durai Dominic, and Aiman Moyaid Said. 2014. Network anomaly detection approach based on frequent pattern mining technique. In *International Conference on Computational Science and Technology (ICCST'14), Malaysia*. IEEE, 1–6.

Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition* 58 (2016), 121–134.

Görrel Espelund. 2016. How vulnerable are water utilities to traditional and cyber threats? *Environmental Science & Engineering Magazine*. Retrieved July 18, 2021 from https://esemag.com/featured/how-vulnerable-are-water-utilities-to-cyber-threats/.

Wei Fan, Matthew Miller, Sal Stolfo, Wenke Lee, and Phil Chan. 2004. Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems* 6, 5 (2004), 507–527.

Frédéric Ferraty and Philippe Vieu. 2006. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Natasha Zhang Foutz and Wolfgang Jank. 2010. Research note—prerelease demand forecasting for motion pictures using functional shape analysis of virtual stock markets. *Marketing Science* 29, 3 (2010), 568–579.

Keke Gai, Meikang Qiu, and Houcine Hassan. 2017. Secure cyber incident analytics framework using Monte Carlo simulations for financial cybersecurity insurance in cloud computing. *Concurrency and Computation: Practice and Experience* 29, 7 (2017), e3856.

Long Gao, Susan H. Xu, and Michael O. Ball. 2012. Managing an available-to-promise assembly system with dynamic short-term pseudo-order forecast. *Management Science* 58, 4 (2012), 770–790.

Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2016. A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security* 88–99. Paris Springer, Cham.

Negin Golrezaei, Hamid Nazerzadeh, and Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* 60, 6 (2014), 1532–1551.

Michael Hagenau, Michael Liebmann, and Dirk Neumann. 2013. Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems* 55, 3 (2013), 685–697.

James V. Hansen, Paul Benjamin Lowry, Rayman D. Meservy, and Daniel M. McDonald. 2007. Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. *Decision Support Systems* 43, 4 (2007), 1362–1374.

Mahmudul Hasan, Md Milon Islam, Md Ishrak Islam Zarif, and M. M. A. Hashem. 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things* 7 (2019), 100059.

Elike Hodo, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis, and Robert Atkinson. 2016. Threat analysis of IoT networks using artificial neural network intrusion detection system. In *International Symposium on Networks, Computers and Communications (ISNCC'16), Tunisia*. IEEE, 1–6.

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, et al. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu*. IEEE, 7310–7311.

Mohammad S. Jalali, Michael Siegel, and Stuart Madnick. 2019. Decision-making and biases in cybersecurity capability development: Evidence from a simulation game experiment. *The Journal of Strategic Information Systems* 28, 1 (2019), 66–82.

Julian Jang-Jaccard and Surya Nepal. 2014. A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences* 80, 5 (2014), 973–993.

Wolfgang Jank and Shu Zhang. 2011. An automated and data-driven bidding strategy for online auctions. *INFORMS Journal on Computing* 23, 2 (2011), 238–253.

Jun Jiang, and Symeon Papavassiliou. 2004. Detecting network attacks in the Internet via statistical network traffic normality prediction. *Journal of Network and Systems Management* 12, 1 (2004), 51–72.

Jianmin Jiang and Lasith Yasakethu. 2013. Anomaly detection via one class SVM for protection of SCADA systems. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing*. IEEE, 82–88.

Lawrie Jones. 2016. Securing the smart city. Retrieved July 18, 2021 from https://eandt.theiet.org/content/articles/2016/05/securing-the-smart-city/.

Wolfgang Ketter, John Collins, Maytal Saar-Tsechansky, and Ori Marom. 2018. Information systems for a smart electricity grid: Emerging challenges and opportunities. *ACM Transactions on Management Information Systems (TMIS)* 9, 3 (2018), 1–22.

Rida Khatoun and Sherali Zeadally. 2016. Smart cities: Concepts, architectures, research opportunities. *Communications of the ACM* 59, 8 (2016), 46–57.

Rob Kitchin and Martin Dodge. 2019. The (in) security of smart cities: Vulnerabilities, risks, mitigation, and prevention. *Journal of Urban Technology* 26, 2 (2019), 47–65.

Alois Kneip and Klaus J. Utikal. 2001. Inference for density families using functional principal component analysis. *Journal of the American Statistical Association* 96, 454 (2001), 519–542.

John Leyden. 2016. Water treatment plant hacked, chemical mix changed for tap supplies. Retrieved July 18, 2021 from https://www.theregister.co.uk/2016/03/24/water_utility_hacked/.

Haiqin Liu and Min Sik Kim. 2010. Real-time detection of stealthy DDoS attacks using time-series decomposition. In *IEEE International Conference on Communications, Cape Town*. IEEE, 1–6.

N. Locantore, J. S. Marron, D. G. Simpson, N. Tripoli, J. T. Zhang, K. L. Cohen, Graciela Boente, et al. 1999. Robust principal component analysis for functional data. *Test* 8, 1 (1999), 1–73.

Scott I. MacKenzie and Poika Isokoski. 2008. Fitts' throughput and the speed-accuracy tradeoff. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Florence*. 1633–1636.

Teena Maddox. 2016. Smart cities: 6 essential technologies. Retrieved July 18, 2021 from https://www.techrepublic.com/article/smart-cities-6-essential-technologies/.

Leandros A. Maglaras and Jianmin Jiang. 2014. Intrusion detection in SCADA systems using machine learning techniques. In *Science and Information Conference, London*. IEEE, 626–631.

Diana Maltseva. 2018. IoT technology in Healthcare. *The Creation of Smart Hospitals*. Retrieved July 18, 2021 from https://medium.com/@Diana.Maltseva/iot-technology-in-healthcare-the-creation-of-smart-hospitals-926d41bf239.

Alexander McLeod and Diane Dolezel. 2018. Cyber-analytics: Modeling factors associated with healthcare data breaches. *Decision Support Systems* 108 (2018), 57–68.

Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. 2017. ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis. In *Proceedings of the Symposium on Applied Computing, Marakkesh*. 506–509.

Patric Nader, Paul Honeine, and Pierre Beauseroy. 2016. Detection of cyberattacks in a water distribution system using machine learning techniques. In *6th International Conference on Digital Information Processing and Communications (ICDIPC'16), Beirut*. IEEE, 25–30.

Ellen Nakashima. 2011. Foreign hackers targeted U.S. water plant in apparent malicious cyber attack, expert says. Retrieved July 18, 2021 from https://www.washingtonpost.com/blogs/checkpoint-washington/post/foreign-hackers-broke-into-illinois-water-plant-control-system-industry-expert-says/2011/11/18/gIQAgmTZYN_blog.html?utm_term=.e531bba76430.

Daniela Oliveira. 2010. Cyber-terrorism & critical energy infrastructure vulnerability to cyber-attacks. *Environmental & Energy Law & Policy Journal* 5 (2010), 519.

Jesus Pacheco and Salim Hariri. 2016. IoT security framework for smart cyber infrastructures. In *IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W'16), Augsburg*. IEEE, 242–247.

Dong Ping Tian. 2013. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering* 8, 4 (2013), 385–396.

J.-P. Planquart. 2001. Application of neural networks to intrusion detection. *Sans Institute* Retrieved July 18, 2021 from http://rr.sans.org/intrusion/neural.php.

Sutharshan Rajasegarar, Christopher Leckie, and Marimuthu Palaniswami. 2014. Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. *Journal of Parallel and Distributed Computing* 74, 1 (2014), 1833–1847.

Sam Ransbotham, Robert G. Fichman, Ram Gopal, and Alok Gupta. 2016. Special section introduction—Ubiquitous IT and digital vulnerabilities. *Information Systems Research* 27, 4 (2016), 834–847.

Sarah J. Ratcliffe, Leo R. Leader, and Gillian Z. Heller. 2002. Functional data analysis with application to periodically stimulated foetal heart rate data. I: Functional regression. *Statistics in Medicine* 21, 8 (2002), 1103–1114.

Shahid Raza, Linus Wallgren, and Thiemo Voigt. 2013. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Networks* 11, 8 (2013), 2661–2674.

Eli Rosenberg and Maya Salam. 2017. Hacking attack woke up Dallas with emergency sirens, Officials Say. Retrieved July 18, 2021 from https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html?_r=1&utm_source=MIT+Technology+Review&utm_campaign=056ffab32c-The_Download_2017-04-07&utm_medium=email&utm_term=0_997ed6f472-056ffab32c-154352697&mtrref=undefined.

David E. Sanger. 2016a. Utilities cautioned about potential for a cyberattack after Ukraine's. Retrieved July 18, 2021 from https://www.nytimes.com/2016/03/01/us/politics/utilities-cautioned-about-potential-for-a-cyberattack-after-ukraines.html. (2016).

David E. Sanger. 2016b. U.S. Indicts 7 Iranians in cyberattacks on banks and a dam. Retrieved July 18, 2021 from https://www.nytimes.com/2016/03/25/world/middleeast/us-indicts-iranians-in-cyberattacks-on-banks-and-a-dam.html.

Naoum Sayegh, Imad H. Elhajj, Ayman Kayssi, and Ali Chehab. 2014. SCADA intrusion detection system based on temporal behavior of frequent patterns. In *17th IEEE Mediterranean Electrotechnical Conference (MELECON'14), Porto Vallerta.* IEEE, 432–438.

Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. 2017. Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Los Angeles.* ACM, 1–10.

Robert P. Schumaker and Hsinchun Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)* 27, 2 (2009), 1–19.

Andrei Sleptchenko and M. Eric Johnson. 2015. Maintaining secure and reliable distributed control systems. *INFORMS Journal on Computing* 27, 1 (2015), 103–117.

Heung-Il Suk, Seong-Whan Lee, Dinggang Shen, and Alzheimer's Disease Neuroimaging Initiative. 2014. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage* 101 (2014), 569–582.

Heung-Il Suk, Seong-Whan Lee, Dinggang Shen, and Alzheimer's Disease Neuroimaging Initiative. 2015. Latent feature representation with stacked auto-encoder for AD/MCI diagnosis. *Brain Structure and Function* 220, 2 (2015), 841–859.

Montbel Thibaud, Huihui Chi, Wei Zhou, and Selwyn Piramuthu. 2018. Internet of Things (IoT) in high-risk environment, health and safety (EHS) industries: A comprehensive review. *Decision Support Systems* 108 (2018), 79–95.

Todd Thibodeaux. 2017. Smart cities are going to be a security nightmare. Retrieved July 18, 2021 from https://hbr.org/2017/04/smart-cities-are-going-to-be-a-security-nightmare.

Marina Thottan and Chuanyi Ji. 1999. Statistical detection of enterprise network problems. *Journal of Network and Systems Management* 7, 1 (1999), 27–45.

Hui Tian and Meimei Ding. 2016. Diffusion wavelet-based anomaly detection in networks. In *17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'16), Guangzhou.* IEEE, 382–386.

Chi-Ho Tsang and Sam Kwong. 2005. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. In *IEEE International Conference on Industrial Technology.* IEEE, 51–56.

A. Onuchowska, S. Chakraborty, W. Jank, and U. Shrivastava. 2018. Detection and classification of attacks on IoT networks. In *24th Americas Conference on Information Systems AMCIS 2018, New Orleans. Association for Information Systems.*

Shahid Ullah and Caroline F. Finch. 2013. Applications of functional data analysis: A systematic review. *BMC Medical Research Methodology* 13, 1 (2013), 43.

Koen Uyttenhove and Michel SJ Steyaert. 2002. Speed-power-accuracy tradeoff in high-speed CMOS ADCs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49, 4 (2002), 280–287.

Jouni Viinikka, Debar Hervé, Mé Ludovic, Anssi Lehikoinen, and Mika Tarvainen. 2009. Processing intrusion detection alert aggregates with time series modeling. *Information Fusion* 10, 4 (2009), 312–324.

Eduardo Viegas, Altair Santin, Alysson Bessani, and Nuno Neves. 2019. BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems* 93 (2019), 473–485.

Bradley W. Vines, Carol L. Krumhansl, Marcelo M. Wanderley, and Daniel J. Levitin. 2006. Cross-modal interactions in the perception of musical performance. *Cognition* 101, 1 (2006), 80–113.

Roberto Viviani, Georg Grön, and Manfred Spitzer. 2005. Functional principal component analysis of fMRI data. *Human Brain Mapping* 24, 2 (2005), 109–129.

Brett Walton. 2017. Water utility cyberattack rings up hefty data charges—circle of blue. Retrieved July 18, 2021 from http://www.circleofblue.org/2017/water-management/water-utility-cyberattack-rings-hefty-data-charges/.

Shanshan Wang, Wolfgang Jank, and Galit Shmueli. 2008. Explaining and forecasting online auction prices and their dynamics using functional data analysis. *Journal of Business & Economic Statistics* 26, 2 (2008), 144–160.

Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-yu Yang. 2004. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1 (2004), 131–137.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. 2005. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association* 100, 470 (2005), 577–590.

Adel Yazdanmehr and Jingguo Wang. 2016. Employees' information security policy compliance: A norm activation perspective. *Decision Support Systems* 92 (2016), 36–46.

Yishi Zhang, Qi Zhang, Zhijun Chen, Jennifer Shang, and Haiying Wei. 2019. Feature assessment and ranking for classification with nonlinear sparse representation and approximate dependence analysis. *Decision Support Systems* 122 (2019), 113064.

Received May 2020; revised December 2020; accepted April 2021