CASPER: Context-Aware IoT Anomaly Detection System for Industrial Robotic Arms

HAKAN KAYAN, Cardi University, United Kingdom
RYAN HEARTFIELD, Exalens, United Kingdom
OMER RANA, Cardi University, United Kingdom
PETE BURNAP, Cardi University, United Kingdom
CHARITH PERERA, Cardi University, United Kingdom

Industrial cyber-physical systems (ICPS) are widely employed in supervising and controlling critical infrastructures (CIs), with manufacturing systems that incorporate industrial robotic arms being a prominent example. The increasing adoption of ubiquitous computing technologies in these systems has led to bene ts such as real-time monitoring, reduced maintenance costs, and high interconnectivity. This adoption has also brought cybersecurity vulnerabilities exploited by adversaries disrupting manufacturing processes via manipulating actuator behaviors. Previous incidents in the industrial cyber domain prove that adversaries launch sophisticated attacks rendering cyber-only anomaly detection mechanisms insu cient as the "physics" involved in the process is overlooked. To address this issue, we propose an IoT-based cyber-physical anomaly detection system that can detect motion-based behavioral changes in an industrial robotic arm. We apply both statistical and state-of-the-art machine learning (ML) methods to real-time Inertial Measurement Unit (IMU) data collected from an edge development board attached to an arm doing a pick-and-place operation. To generate anomalies, we gradually modify the joint velocity of the arm. Our goal is to create an air-gapped secondary protection layer to detect "physical" anomalies without depending on the integrity of network data, thus augmenting overall anomaly detection capability. Our empirical results show that the proposed system, which utilizes 1D-CNNs, can successfully detect motion-based anomalies on a real-world industrial robotic arm. The signi cance of our work lies in its contribution to developing a comprehensive solution for ICPS security, which goes beyond conventional cyber-only anomaly detection methods. By incorporating physical measurements into the anomaly detection process, the proposed system can detect movement-based anomalies occurred on a real-world industrial robotic arm testbed.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Hardware** → *Sensor applications and deployments*; • **Computing methodologies** → *Anomaly detection*.

Additional Key Words and Phrases: neural networks, anomaly detection, industrial robotic arms, cyber-physical systems, ubiquitous computing

ACM Reference Format:
Hakan Kayan, Ryan Heart eld, Omer Rana, Pete Burnap, and Charith Perera. 2023. CASPER: Context-Aware IoT Anomaly Detection System for Industrial Robotic Arms. 1, 1 (September 2023), 30 pages. https://doi.org/10.1111/1111111.1111111

Authors' addresses: Hakan Kayan, kayanh@cardi .ac.uk, Cardi University, Senghennydd Rd, Cardi , United Kingdom, CF24 4AX; Ryan Heart eld, ryan.heart eld@exalens.com, Exalens, Plexal, Here East, London, United Kingdom, E20 3BS; Omer Rana, RanaOF@cardi .ac.uk, Cardi University, Senghennydd Rd, Cardi , United Kingdom, CF24 4AX; Pete Burnap, BurnapP@cardi .ac.uk, Cardi University, Senghennydd Rd, Cardi , United Kingdom, CF24 4AX; Charith Perera, pererac@cardi .ac.uk, Cardi University, Senghennydd Rd, Cardi , United Kingdom, CF24 4AX.

XXXX-XXXX/2023/9-ART $15.00
https://doi.org/10.1111/1111111.1111111

1 INTRODUCTION

Industrial cyber-physical systems (ICPS) [22], which is the backbone of Industry 4.0 [59], are the result of adapting emerging information communication technologies (ICT) to the industrial control systems (ICS). Implementing advanced ubiquitous computing resources enables interconnecting the cyber and physical assets of ICPS. This provides the ability to supervise sophisticated industrial systems where each layer (e.g., production, corporate) contains interdependent operations. Hence, a broad range of domains that manage critical infrastructures (CIs), including manufacturing, transportation, and healthcare employs ICPS. Academia and industry refer to these domains as "smart" [55] as the assets of ICPS can self-supervise. In smart systems, actuators operate according to information generated from corresponding sensors. The heterogeneity of the industrial environment may require an adaptive actuation that is directed by multiple sensor data. An autonomous robotic arm[1] executing repetitive patterns to assemble car parts, a conveyor belt that rotates based on the specific product carried, and a furnace that decreases or increases gas supply to heating elements according to processed material and temperature are such examples of cyber-physical systems.

The International Federation of Robotics (IFR) report published in 2022 [85] shows that collaborative robots (cobots) will lead the robotics industry after 2025. The rapid development of these autonomous robots that can perform repetitive tasks accelerates the utilization of highly interconnected industrial infrastructures. However, high interconnectivity means increased attack surface which mainly occurs due to the integration of information technologies (IT) to operational technologies (OT). Thus, ICPS are exposed to attacks that were not an issue for legacy ICS. These attacks become successful when inadequate cybersecurity measures are present causing disasters [97, 123] as ICPS supervise CIs. The majority of attack detection solutions rely on intrusion detection systems (IDS) [65] which only perform network traffic analysis (NTA). As industrial systems have different security requirements, the characteristics of industrial IDS differ from their peers [42]. These IDS operate in the "cyber" domain of ICPS where sophisticated attacks (e.g., stealthy attacks, advanced persistent threats (APT)) can penetrate through to disturb the physical processes. Physics-based attack detection mechanisms [125] observe these processes to detect any kind of abnormal behaviors hence monitoring the "physical" side of ICPS.

We consider attack detection as a sub-group of anomaly detection [17] (also known as outlier detection) as the anomalies in ICPS may occur due to three main reasons: attack, failure due to degradation, and misconfiguration. These anomalies can be either cyber or physical while both can occur either at once or at independent times. An example where both occur due to an attack would be a successful distributed denial-of-service (DDoS) [82] attack that causes the stoppage of the robotic arm (physical anomaly) due to missing network packets (cyber anomaly). We consider such an attack as a cyber-physical attack [81] as the attack causes physical alterations. An example where only a physical anomaly occurs due to degradation would be a change in the acceleration of the robotic arm due to corrosion on the bearings. IDS fail to detect such deviation either when the affected asset is not monitored or when the data are spoofed by an adversary. One other precaution against cyber-physical attacks is to set thresholds for physical characteristics (e.g., setting the joint speed limit for an industrial robotic arm, and setting the heat limit for an oven). As these thresholds mostly determine upper and lower limits they fail to identify time-sensitive anomalies within these limits. Hence, these kinds of events require contextual physics-based monitoring mechanisms.

Fault diagnosis [45] an early discipline that examines unwanted physical deviations of system characteristics, has similarities with anomaly detection. However, the primary difference is that fault diagnosis aims to identify the reason for the anomaly. There are two main types of fault diagnosis: model-based [46], and signal-based [32]. Model-based approaches attempt to generate an

[1]From now on, an arm refers to an industrial robotic arm.

explicit model of system behavior to predict the output while signal-based approaches process raw sensor measurements to predict the healthy state of the system. Anomaly detection also has two similar approaches: model-based [116], and data-driven [117]. The two signi□cant drawbacks of model-based approaches are: (I) They require expert knowledge, which is hard to obtain due to the high complexity of industrial cyber-physical systems, making this task laborious and error-prone for humans. (II) They depend on the integrity of components, which must be trusted. This dependence on components' integrity raises concerns about the cybersecurity of these parameters, as they can be spoofed through integrity attacks [119]. The Stuxnet malware [57] attack on Iran's nuclear centrifuges is a real-world example of such an integrity attack, where attackers modi□ed the gas centrifuge parameters. To address these drawbacks, data-driven approaches [86, 92] have become increasingly popular due to the rapid development of data technologies. These approaches utilize machine learning models, which can be grouped into three based on supervision [17]: supervised, semi-supervised, and unsupervised. The supervised models use labeled data for training, while the unsupervised models either do not require any training data [66] or use non-labeled data for training [54]. Semi-supervised models combine these two.

Neural networks [39] are a type of machine learning method that mimics the structure of the human brain, utilizing connected neurons and activation functions to learn from data. Neural networks are typically categorized based on network structure [58]: shallow neural networks (SNN), and deep neural networks (DNN). Bianchini and Scarselli [13] propose a detailed comparison regarding the complexity of these two neural network types. The □exibility and scalability of neural networks make them desirable for industrial applications. In recent years, academia presented many DNN-based research papers [35, 44, 54, 75], which o□er promising results, within the context of detecting physical anomalies in ICPS.

Computing infrastructures can be grouped into three based on computing location [135]: edge, fog, and central/cloud. In short, we de□ne "edge" as the location where real-world data are present, "cloud" as the servers that are accessed via the internet, and "fog" as anything between the edge and cloud. If we imagine an assembly line, we consider the distributed embedded devices on arms that interfere with the sensor data as edge devices, and a local device that manages several edge devices while forwarding data (either raw or preprocessed) to the cloud as a fog device. Central (local) servers might be preferred if cloud systems are undesired or unreachable. As Internet of Things (IoT) devices enable access to the cloud, they are heavily utilized in both edge and fog.

Training neural networks is a resource-intensive task, requiring substantial computational resources. Cloud computing platforms such as Amazon Web Services (AWS) [21], Google Cloud [14], and Microsoft Azure [80] are attractive options as they o□er machine learning as a service (MLaaS) [100]. These platforms can be integrated into local builds to establish an automated ML pipeline as such a pipeline requires edge devices to generate raw data, and an internet connection to access cloud services, IoT-based solutions become desirable choices. Local data science workstations are alternatives to these services. If the domain is industrial, the industrial internet of things (IIoT) [114] is utilized. We consider IIoT as one of the requirements for advanced/smart manufacturing. While the initial IIoT solutions [56, 127] focus on increasing production e□ciency, the use of IIoT to detect anomalies [90, 111] is gaining popularity thanks to rapid developments in ubiquitous computing technologies.

In this work, we propose an anomaly detection system that detects movement-based physical anomalies which occur in an industrial robotic arm. We utilize statistical and ML based methods including neural network model utilizing 1D convolutional neural networks (1D-CNN) layers. To the best of our knowledge, we are the □rst to propose a context-aware anomaly detection system (CASPER) that detects movement-based anomalies by applying the 1D-CNN model on IMU data. CASPER is also segregated hence ensuring the integrity of data generated via a cyber-physical

edge resource as data are transmitted over Bluetooth Low Energy (BLE). We summarize our key contributions as:

- We propose an anomaly detection model that utilizes 1D-CNN to detect anomalies occurring due to deviation of joint velocities of an industrial robotic arm while o□ering an IoT-based edge monitoring system. We demonstrate the performance of the proposed model on a real-world testbed. We present the work to the public on a well-documented GitHub repository[2].
- We publish a real-world dataset that contains four □les in total: (I) A □le that consists of accelerometer, gyroscope, and magnetometer data of an arm that accomplishes a repetitive task, (II) two □les (one per industrial arm) that consist of built-in arm parameters such as joint current, and velocity values, (III) one pcap □le which contains all the network tra□c between the local PC and the industrial robotic arms.
- We analyze the recent real-world industrial cyber-physical incidents.
- We present a thorough correlation analysis between the raw IMU data and the quaternion representation of orientation to show how proposed model perform when data are correlated.

2 BACKGROUND

Our work focuses on applying cyber-physical anomaly detection to arms that operate in manufacturing systems. Cyberattacks that disrupt the behavior of such actuators may cause devastating events. In this section, we summarize recent real-world cyber incidents based on the attack scope, domain, and result while identifying the common points that motivate the proposed work.

In 2013, the maximum-security prison Turner Guilford Knight Correctional Center in Florida, USA had been subjected to two cyber incidents in one month [103]. The prison control system was recently upgraded for a cost of $1.4 by a □rm named Black Creek Integrated Systems. All cell gates in the prison were automatically opened, thus leading to chaos within the prison. Even though the director named the incidents a glitch, a surveillance video had shown that some prisoners were acting as if they knew the gates were about to be opened. Hence, cybersecurity researchers suspected that the □rst event was done to test the response of the guards, and the second was carried out for a more speci□c reason as 2 prison members tried to attack another prisoner. These incidents have shown that even air-gapped systems can be programmed to glitch to cause a cyber incident, hence air-gapping only is not adequate to secure the systems.

On February 8, 2021, an adversary tried to poison Oldsmar, a city in Florida, USA [97]. The adversary accessed the computer that hosts the water treatment control software via a remote access program, then increased the amount of sodium hydroxide above the normal level. The water concentration change was seen by an operator and immediately reversed. Then, the remote access was disabled. How computer credentials were captured is still unknown. In this incident, having 24/7 IT sta□ (which is not the case for most industrial systems) to supervise the system prevented the possible disaster from happening. Also, the adversary did not fake the sensor readings hence the unexpected change was detected.

In May 2021, the US Colonial Pipeline was hit by ransomware that is developed by a group known as DarkSide [123]. The attack was directed at a pipeline not to damage but to extort money from the owner company. All the activities of the pipeline had to shut down due to being connected to a central system. The pipeline was equipped with the newest digital sensors including a smart pipeline inspection gauge. However, due to being connected to a central system, all access to sensors was blocked. Hence the operators shut down the pipeline. How the attackers deployed the ransomware is unknown but assumed to be done via phishing e-mails. This incident is an example of the downside of being highly interconnected.

[2]https://github.com/hkayann/1D-CNN-Anomaly-Detection-via-CASPER

Table 1. The Evaluation of Recent Cyber Incidents

Year	Incident Subject	Location	Sector	Attack Scope	IT	OT	Result
2013	Prison	USA	Utility	Cyber-Physical	●	●	Prison gates were wrongfully opened
2019	Altran Technologies	France	Service Provider	Cyber	●	●	IT network was shut down.
2019	Healt Facilities	Australia	Healthcare	Cyber	●	○	Health operations were delayed.
2020	HUBER+SUHNER	Switzerland	Manufacturing	Cyber	●	◐	All network was shut down.
2021	Bombardier	Canada	Manufacturing	Cyber	●	○	Customer data was stolen.
2021	Colonial Pipeline	USA	Utility	Cyber-Physical	●	●	Pipeline was shut down.
2021	Water Plant	USA	Utility	Cyber-Physical	●	●	Water is poisoned.
2021	Ca□taly	Italy	Manufacturing	Cyber	●	◐	Production was stopped.
2021	MND Group	France	Manufacturing	Cyber	●	◐	Production was stopped.
2021	Sierra Wireless	Canada	Manufacturing	Cyber	●	◐	Production was stopped.

Legend: ● : The domain is directly a□ected, ◐ : The domain is indirectly a□ected, ○ : The domain is not a□ected.

In March 2021, Canadian IoT as a service provider Sierra Wireless was subjected to a ransomware attack [15]. The IT systems of the company were locked down. The company announced that there was no damage done to any production units and the con□dential customer data was not a□ected thanks to being stored on an independent platform. However, the company halted production for over two weeks until the systems were cleared. This incident shows the importance of reaction time and having independent domains.

On February 23, 2021, the Canadian plane manufacturer Bombardier announced that it was subjected to a cyberattack [77]. The attack was done by exploiting a vulnerability belonging to a third-party □le transfer application hosted on a separate server. The attackers in□ltrated the con□dential data related to customers and suppliers. The internal IT and OT systems of the company were not a□ected there was no network connection between the systems. Thus, the company did not halt production. This incident shows the importance of keeping third-party applications up-to-date and having independent servers/systems.

On January 24, 2019, French engineering consultancy company Altran Technologies was subjected to a ransomware attack based on a crypto locker even though their systems were protected via □rewalls and several IT defense mechanisms [37]. The company had to shut down all of its IT network and applications across Europe. They acquired cybersecurity services from third-party providers to bring their systems back to normal. This incident shows that having up-to-date cybersecurity defense mechanisms does not provide 100% security, hence the companies should have ready-to-deploy mitigation/recovery plans.

On December 14, 2020, HUBER+SUHNER, a □ber optic cable manufacturing company located in Switzerland, was subjected to a cyberattack [93]. When the internal IT monitoring system detected an unknown activity, the company shut down all of its operations to prevent possible damage from happening at production sites due to having a highly interconnected network. As a result, no physical damage occurred. The company contacted third-party security providers to analyze the attack, then gradually resumed its operations. In this incident, the physical damage was prevented thanks to the rapid reaction, however, the con□dential data was stolen.

In February 2021, the Italian co□ee capsule/machine manufacturer Ca□taly System was subjected to a cyberattack [23]. The company was outsourcing the IT services to a third-party provider, which was exploited by adversaries. The production was halted to prevent further damage as the IT and OT systems were interconnected. The reason/motivation behind the attack is unknown as the company did not share the details of the incident. While outsourcing IT/Cybersecurity services to third parties is considered a compact solution by many cybersecurity providers, this incident was caused via such a provider.

On March 22, 2021, the French artificial snow manufacturer the MND Group detected malware on its servers located in France and Austria [129]. The company shut down its all IT network to prevent a further breach. The OT systems were not heavily affected by the attacks thanks to being disconnected from IT systems, hence the company halted production for only a few days as a precaution. The company put a business recovery plan into practice to recover from the attack within a week. The details of the attack were not shared with the public. Having a ready-to-deploy recovery plan was the key feature to mitigate the result of this cyber incident.

In September 2019, Eastern Health facilities in Victoria, Australia were subjected to a ransomware attack [98]. Several servers that hosted financial, booking, and management data were shut down due to being captured, hence the hospitals had to delay operations including not critical surgeries. The authorities and cybersecurity experts were contacted to resolve the issue. In this incident, the attacked domain was purely cyber but, there was an indirect physical impact that occurred due to the lack of data availability.

Most private entities subjected to cyber incidents do not publish official statements. The information is made available via cybersecurity journals/bloggers which beclouds verifying the incident details such as the cause, response, and already deployed security mechanisms. We observe the following from the aforementioned cyber incidents: (I) The example attacks demonstrate that integration of IT to OT systems clearly exposes OT systems to new threats. (II) We can safely assume that the companies have at least one intrusion detection/prevention tool (e.g., default defender, antivirus software) in place during the incident thus proving the inefficiency of these tools. (III) Additional security measures that observe the targeted infrastructure can detect the undesired changes. We see this both in the Iranian nuclear program [57] and Florida water poisoning [97] incidents where attacks were detected via the supervisory staff. The recent industrial cyber incidents prove the necessity of security measures which observe the physical properties from an air-gapped/segregated network which can ensure the integrity of industrial processes.

3 RELATED WORK

3.1 Anomaly Detection in Industrial Systems

Anomaly detection in industrial systems is a topic where an extensive number of studies are present [17, 30, 51, 124]. Detecting anomalies based on physical behavioral changes via data-driven approaches is one of the hot sub-branches. These changes differ according to the monitored asset. If this asset is an industrial robotic arm, data-driven approaches are applied where the data are sound [9, 27], IMU [87], joint current [18, 91], electromagnetic side-channel signal [52], tension [99], vibration [92], or visual [133] data. In addition to these, we can utilize temperature data [120] to detect anomalies as malfunctioning industrial assets tend to generate unusual heat. As we can remotely measure environmental sensing data such as temperature, humidity, barometric pressure, and CO_2 level, we can deploy mobile physical anomaly detection units [34], which provide flexible real-time physical anomaly detection, in industrial sites. Unlike model-based anomaly detection approaches, data-driven approaches can be scaled into heterogeneous environments. SWaT [78] is a water treatment testbed that contains around 68 sensors and actuators. Hence, the SWaT dataset contains both discrete and continuous sensor data. In addition, the sensors have different sampling rates. This kind of environment is challenging due to its high diversity. Recent research [54, 95, 130] shows that data-driven approaches do well even in such environments.

3.2 Role of IoT within Anomaly Detection

Time series data generated by sensors in IoT applications often exhibit temporal correlations resulting in contextual anomalies where the context is time. Detection of such anomalies can be

challenging as compared to point anomalies, making available solutions computationally complex [49, 92]. This proposes no issue if the detection is done o□ine (see Section 3.4). Real-world industrial applications are mostly time-sensitive (e.g., manufacturing, fuel extraction). In this case, the common approach is to use IoT sensors/devices to enable cloud access where high computing power is available [76]. However, the occurrence of delay causes researchers to pursue alternative approaches [88, 109]. This delay can also be eliminated by applying anomaly detection on edge devices. The available methods are pretty limited but expanding [24] thanks to the rapid development of ubiquitous technologies. IoT devices are also used for real-time monitoring [94, 96] which might be critical (see Section 2) when the other security mechanisms in place fail. We utilize IoT for edge data monitoring while considering edge anomaly detection implementation as future work.

3.3 Applying Machine Learning on Multimodal Sensor Data

In an ideal scenario, multiple sensor data sources are employed to monitor/supervise systems as each sensing modality provide unique/more context combined to produce an accurate representation of the environment. This approach is common in human activity recognition (HAR) applications [84, 102]. For example, the Apple Watch [7] tracks a user's sleep by combining heart rate and accelerometer data or calculates the number of steps taken based on geolocation and acceleration data. The features extracted from these modalities are either combined into a single feature vector (feature concatenation) [38, 89, 137] or utilized individually (ensemble classi□ers) [6, 40, 118, 128]. Traditional machine learning (ML) methods use a single modality for each stage of the ML application [104]. Multimodal fusion approaches employ all modalities at each stage [11, 25]. Cross-modality learning approaches [41, 136] utilize all modalities during feature learning while training and testing are performed with the same single modality, which di□ers from shared representation learning [79, 134], where di□erent modalities are used for testing and training.

3.4 Sensor Data Analysis with ML-based Approaches

Data-driven ML methods are grouped into three [33] based on the: (I) supervision, (II) time, and (III) working principle. *Supervision*. ML methods are *supervised* if labels (e.g., anomaly, normal) are fed during training. Supervised methods are common in human activity recognition (HAR) [10]. However, labeled data might be hard to obtain. In this case, the *semi-supervised* method, which is a mix of supervised and unsupervised, is applied. Generating labels from unlabeled data for training is an example use case. Pipe damage detection [110] is one of the areas where semi-supervised learning is preferred. *Unsupervised* learning is applied if the model is expected to learn without any human interference. These methods are popular in anomaly detection [50] where normal data are fed during training and then the model is expected to recognize unknown/novel data. The learning also might depend on a policy where the model learns by its actions. *Reinforcement learning* is such an example that can be seen in game-playing robots [113]. The learning might be online or o□ine. *Online* algorithms learn on the □y while *batch/o ine* learning makes use of pre-gathered data to train the model. Adaptive ML models [83] require online learning algorithms due to novel streaming data. O□ine learning is more common in classi□cation tasks such as natural language processing (NLP) [68] where the capacity of the model depends on the size/content of the training data. ML models can also be classi□ed into two according to working principles: instance-based, and model-based. The instance-based ones analyze the correlation between the known points and new points while the model-based algorithm tries to understand the behavior of data patterns. Instance-based methods are popular in image classi□cation [20] while model-based methods are seen in predictive analytics/forecasting [107]. Figure 1 demonstrates the aforementioned ML categories.

CNNs o□er several advantages over their counterparts: are widely used in various machine learning applications due to their advantages over traditional models while one of them is to extract

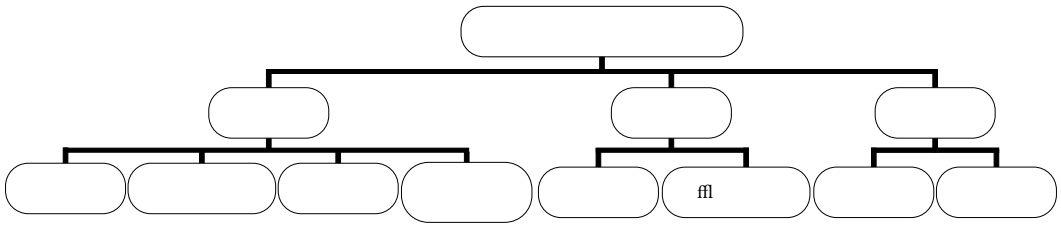


Fig. 1. We group machine learning types based on supervision, time, and working principle.

features automatically eliminating the need for manual feature extraction, a labor-intensive task. CNNs have a lower computational complexity than fully connected models, as local neurons are only connected to a certain group of layers, and feedback loops, as seen in Recurrent Neural Networks (RNN), are not required [108]. CNNs can be either 1D, 2D, or multi-dimensional. While 2D-CNNs are the de facto choice for input data with a strong 2D structure that correlates spatially (e.g., images, and speech) [64], 1D-CNNs are useful for time series data as such data are expected to have strong temporal correlations [60]. 1D-CNNs are also computationally less expensive as they require exponentially fewer operations making them desirable for real-time sensing applications. The various recent applications of 1D-CNNs include ball bearing fault detection [43], water treatment system anomaly detection [54], HAR [19], seizure detection [48], and music genre classi□cation [4].

4 ANOMALIES

In the □eld of data science, anomalies or outliers are data that deviate from the expected patterns of behavior. In other disciplines, such anomalies may also be referred to as "abnormalities" , though this term is also used to de□ne a behavior. This section provides an overview of di□erent types of anomalies, decision-making methods, and techniques for generating anomalous data.

4.1 Anomaly Types

Anomalies are classi□ed into three categories [17]: (I) point anomalies, (II) contextual anomalies, and (III) collective anomalies. *Point anomalies* di□er from the rest of the data. Being the most common ones, if the anomaly type is not mentioned, it usually refers to point anomalies [70, 105, 132]. *Contextual anomalies* are harder to detect as such detection requires context (e.g., time, location) analysis where de□ning one might be challenging. The application that generates time series data tends to contain contextual anomalies where the context is the time [16, 67]. *Collective anomalies* is a group of data that di□ers from the rest being relatively rare due to their nature. Triggering certain malicious network actions in order can cause a collective anomaly that can be identi□ed via network anomaly detection methods [2, 3]. Figure 2 demonstrates each type of anomaly that can occur on an industrial robotic arm that operate in manufacturing plants.

4.2 Anomaly Decision Methods for Sensor Data

Anomalies are de□ned as either binary (e.g., 0 for normals and 1 for anomalies) or via anomaly score which mostly scales between 0 and 1. Then these scores might be converted into binary labels by using a certain threshold. While boundary-de□ning methods such as SVMs [86] tend to utilize binary de□nitions, decision tree-based approaches such as Isolation forest [66] utilizes anomaly scores. On the other hand, regression methods (e.g., gradient boosting, logistic regression) estimate a value. Then statistical methods are applied to the residuals which are the absolute di□erence between the predicted and actual values.

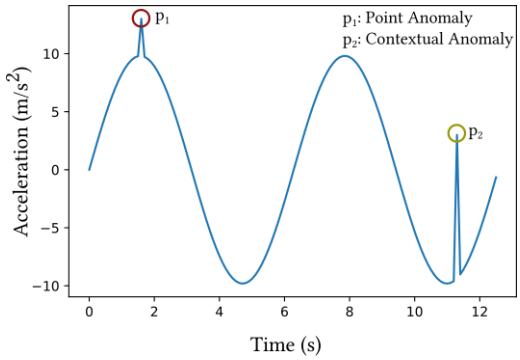


Fig. 2. Demonstrates an example acceleration data of one industrial robotic arm joint. While the point anomaly p_1 does not appear across the data (or appears very less in numbers), the contextual p_2 does. While p_1 can be detected via simple thresholding, more sophisticated methods are required to detect p_2. Collective anomaly is the event where point/contextual anomaly occurs simultaneously across all joints.

Table 2. Anomaly Creation Methods

Reference	Testbed	Attack	Anomaly Creation Method
Narayanan and Bobba [86]	Industrial Robotic Arm	-	Set industrial arm to follow a di□erent trajectory.
Chen et al. [18]	Industrial Robotic Arm	-	Manually injecting faults.
Khan et al. [52]	Robotic Arm Syringe Pump	✓	Implementing control-□ow hijack and □rmware modi□cation attacks.
Riazi et al. [99]	Belt-driven Robotic Arm	-	Loosening and tightening the belt.
Park et al. [92]	Robot Manipulator	-	Adjusting the amount of air injected into vacuum ejector.
Angle et al. [5]	High Voltage Motor Development Kit	-	Modifying the □rmware to allow to damage the kit.
Vuong et al. [126]	Robotic Vehicle	✓	Conducting DoS attack.
Wu et al. [131]	3D Printer	-	Injecting faulty □les to 3D printer to print a damaged product.
Gao et al. [31]	3D Printer	-	Modifying the □rmware to change printer features such as printing velocity.
Li et al. [63]	Rotor Kit	-	Adding weights to a mass load.
Bezemskij et al. [12]	Robotic Vehicle	✓	Conducting replay attack, creating rogue node, manipulating compass, and breaking wheel.
Sonntag et al. [115]	Industrial Robotic Arm	-	Hitting to an industrial arm.
Sisinni et al. [114]	Robotic Vehicle	✓	Conducting DoS, command injection, and malware attack.
CASPER	Industrial Robotic Arm	-	Manually manipulating the joint velocity of the arm.

4.3 Generating Anomalies on Cyber-Physical Systems

While the use of public datasets [26, 35, 61, 62, 78] enables benchmarking similar works, having no control over anomaly creation beclouds the recreation of desired challenging scenarios. This also applies to simulation-only studies [29, 101]. Thus, real-world testbeds are required to assess practicality. Generating anomalies on such a testbed that replicates the original industrial process (e.g., manufacturing) is challenging due to the risk of damaging high-cost equipment. For instance, conducting a cyber attack on a controller unit [5, 31, 69, 131] carries such a risk. In this work, we introduce controlled anomalies as seen in [12, 63, 86, 92, 99, 115] via manually modifying the joint velocities. Table 2 demonstrates the anomaly creation processes of related work.

5 CASPER - SYSTEM OVERVIEW

The CASPER consists of edge, fog, and central components that o□er an open-source low-cost IoT-based monitoring system. In this section, we present each component of CASPER while justifying our design choices.

5.1 Edge Components

In this work, we use edge development boards that contain 32-bit microcontroller units (MCUs) for the following reasons: (I) These boards are easy to deploy (attachable), low-cost, and power-e□cient

Table 3. Edge Development Boards Tech Specifications

Name	Arduino Nano 33 BLE Sense	Adafruit Feather nRF52840 Sense	Nicla Sense ME
SoC (Microprocessor)	nRF52840 (ARM Cortex M4)	nRF52840 (ARM Cortex M4)	nRF52832 (ARM Cortex M4)
Memory	256 KB SRAM, 1MB □ash	256 KB SRAM, 1MB □ash	64 KB SRAM, 512 KB □ash
Connectivity	BLE 5.0	BLE 5.0	BLE 4.2
Sensor (Module Name)	IMU (LSM9DS) Microphone (MP34DT05) Gesture, Light, Proximity (APDS9960) Barometric Pressure (LPS22HB) Temperature, Humidity (HTS221)	IMU (LSM6DS33 & LIS3MDL) Microphone (PDM MEMS) Gesture, Light, Proximity (APDS9960) Barometric Pressure (BMP280) Temperature, Humidity (SHT-30)	IMU (BHI260AP & BMM150) Gas, Pressure, Temperature, Humidity (BME688) Pressure (BMP390)

Table 4. Cloud/Central/Fog Tech Specifications

	Google Colab Pro	**Data Science Workstation**	**Raspberry Pi 4B**
GPU	Tesla P100-PCIE-16GB	NVIDIA RTX A6000-48GB	None
CPU	Intel Xeon @2.20GHz	Intel Xeon W-2245 @3.90GHz	Broadcom BCM2711, Quad core Cortex-A72 64-bit SoC @ 1.5GHz
RAM	24 GB	128 GB	4 GB

devices. The IoT environments are dynamic, heterogeneous, and resource-constrained. Thus, we need the aforementioned characteristics to have a sustainable model. (II) They can support BLE, which is a wireless personal area network (WPAN) technology, that enables low-power encrypted wireless communication. (III) They either allow the integration of third-party sensors or come with built-in ones. The boards with built-in sensors remove the need for additional attachments thus o□ering accessible deployment. We compare three edge development boards based on the aforementioned requirements: (I) Arduino Nano 33 BLE Sense [121], (II) Adafruit Feather nRF52840 Sense [1], (III) Nicla Sense ME [122][3]. Table 3 compares tech speci□cations of the utilized edge devices. As we focus on detecting motion-related anomalies of an arm where corresponding data generated on the edge, we consider the following:

- The edge development board should have built-in inertial measurement unit (IMU) sensors. These sensors measure linear acceleration, magnetic direction, and angular velocity to de□ne an orientation.
- The edge development board must provide BLE [112] connectivity. We observed in our previous work [50] that BLE o□ers low power usage and □exibility thus favored in resource-constrained environments. In addition, most system-on-chips (SoC) provide BLE, hence we do not need any additional modules/devices as seen in Zigbee [28] networks.

5.2 Fog Components

The fog device manages several edge devices while acting as a bridge between the edge and the cloud. As the edge devices are resource-constrained, in an IoT environment, connecting internet via the fog device is an optimal solution in most cases. However, as ICPS supervise CIs, one might prefer not to have a cloud connection due to security challenges [106]. In this case, the fog device is also expected to have enough capacity to perform precon□gured tasks (e.g., data monitoring, edge device supervision, data preprocessing). Low cost is another deciding factor as they might be required in great numbers depending on the capacity of industrial area. Based on these, we use an embedded single board computer (SBC) as a fog device in this work. We consider the following as

[3]From now on, we may mention these boards with their initial names only.

key characteristics: (I) It must be portable, small, and low-cost, (II) must be able to connect to the internet, (III) must support BLE as we send edge data over BLE to SBC, (IV) must be able to run an operating system (OS) that supports software tools such as Node-RED (nodered.org) and Grafana (grafana.com). We explain details regarding these tools in the following section.

In this work, we utilize Raspberry Pi 4 (RPi4) as SBC as previous research [8, 36] o□er promising benchmarking results [72]. RPi4 runs on DietPi OS [53], that minimizes resource usage when running Node-RED and Grafana. A more cost-e□cient option would be using an edge development board as fog device, however, due to a lack of on-device training and visualizing support, currently they are not feasible.

5.3 Cloud/Central Components

As ML model training is a resource-intensive task, a cloud or central device with high computing power is required. In an ideal scenario where ML models are deployed for real-world applications, online learning is implemented to prevent the fade of model's e□ciency due to undesired events such as concept drift. However, in this work, we do o□ine learning as our primary target is to investigate the e□ciency of statistical and ML based methods for anomaly detection while o□ering IoT-based monitoring on a realistic environment. We use local data science workstation as central component for resource-intensive operations (e.g., training, development of alternative ML algorithms for comparison) while utilizing fog device to supervise edge data. Table 4 demonstrates the key speci□cations of central, fog device, and an example of Google Colab Pro instance to give an insight about the capability of utilized workstation.

6 EVALUATION

This section presents a detailed description of the experimental setup utilized in this study, including the essential components of the testbed and the use case scenario. We conduct a comparative analysis of three di□erent edge development boards in terms of the generated IMU data and introduce the CASPER dataset. We assess the e□ectiveness of various statistical and machine learning-based methods in detecting movement-based anomalies of an industrial robotic arm. We conduct a comprehensive evaluation of the proposed approach on a real-world industrial robotic arm testbed.

6.1 Experimental Setup

6.1.1 Testbed Components. We utilize a real-world industrial testbed that simulates a pick-and-place task seen in manufacturing systems. Table 5 and Table 6 present the testbed components while explaining their key features and tasks. Figure 3 visualizes each component, demonstrates how each component communicates, de□nes the purpose of each joint of the arm and shows rotations, presents the use case scenario step-by-step, and proposes the real testbed image where the control boxes are not visible due to being located under the desk. The frame and mounting plates of the custom platform are made of aluminum while the legs are made of steel.

6.1.2 Use Case Scenario. 9-DOF multi-jointed industrial robotic arms are used in various industrial applications. These applications include manufacturing-related tasks such as welding, soldering, screw driving, brazing, placing, casting, and painting. The trajectory of the arm depends on the task. For instance, pick-and-place applications mostly require a horizontal trajectory while screw-driving ones require both. The arms repeat the same high-precision tasks which are completed within the certain time intervals. In this work, we examine a pick-and-place scenario (see Figure 3c) while considering the following assumptions:

- The movement is repetitive, has a certain frequency, and continuous.

Table 5. Hardware Components

Component Name	Key Features	Purpose	Location
UR3e 6-DoF Industrial Grade Arm	5kg payload, 500mm reach	Pick and place.	Edge
2FG7 OnRobot Parallel Gripper	37mm maximum width 140N maximum gripping force	Gripping, and releasing the steel ball.	Edge
Controller Box	Built-in ethernet port Input/output (IO) sockets	Main control unit of the arm. Enables remote controlling via urp scripts.	Edge
Custom Platform	~2.5 meter width, ~1 meter height ~1.5 meter length, mostly steel	Base for the arms. Contains two inclined parts that allows ball to roll.	Edge
Steel Ball	25.40mm diameter, 66.84g weigth	It is passed from one arm to another via inclined platform.	Edge
Nicla Sense ME	BLE connectivity IMU sensors	Generates IMU data and forward to fog over BLE.	Edge
Pi-HMI	Touchpad Screen ML capable BLE & Wi-Fi connectivity	Supervises the IMU data and resource usage.	Fog
Network Switch	Power over ethernet (PoE)	Provides TCP/IP communication between PC and arms. Powers Pi-HMI.	Fog
Laptop	Runs Ubuntu, RTDE compatible	Runs Python script to control arms. Generates dataset.	Central
Data Science Workstation	High computing power	Does the training/evaluation of proposed/compared ML models	Central

Table 6. So□ware Components

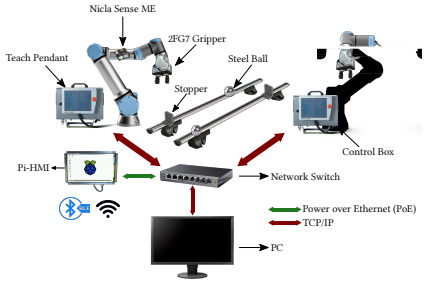
Software Name	Purpose	Version
Grafana	Provides interactive visualization of IMU data.	9.0.9
In□uxDB	Stores the IMU data.	1.8
DietPi OS	Manages Pi-HMI. Power e□cient OS for Pi.	8.0
Ubuntu	Manages the central PC.	20.04
Python	Enables programming of the simulation.	3.8
Universal Robot Scripts (urp)	Communicate with python script to execute commands.	5.11
Arduino Sketch	Runs on Nicla Sense ME. Generates and transmits the IMU data.	1.6.10
Node-RED	Sets up the BLE connection between Pi-HMI and Nano BLE Sense.	3.0

- The arm is autonomous hence does not require any human interaction aside from the initialization phase where no adversarial behaviors are in place.
- The adversary aims to disrupt the physical process. Thus, the behavior of the arm deviates as a result of an attack. The deviation from the behavior might occur as a result of accidental events (e.g., bumping into an industrial arm) as well.
- The integrity of the built-in data is compromised as the adversary has complete control over the communication between the central laptop and the robotic arms.

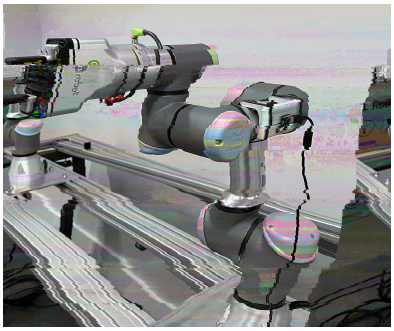
6.2 Sensor Fusion & Edge Development Board Comparison

Micro-electro-mechanical systems (MEMS) sensors that generate IMU[4] data are: (I) accelerometer and (II) gyroscope, and (III) magnetometer. The accelerometer measures the linear acceleration which de□nes the velocity change in units of either gravitational force (g) or meters per second squared ($m\,s^{-2}$). The gyroscope measures the angular velocity which de□nes the rotational change in motion in units of degrees per second (dps). The magnetometer measures local magnetic □eld

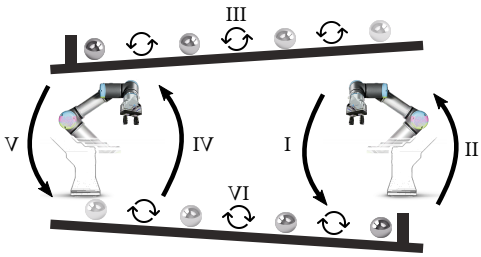
[4]Sometimes IMU is de□ned as magnetic and inertial measurement unit (MIMU) due to the presence of magnetometer.



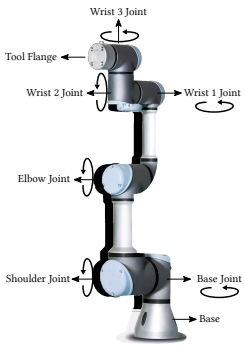
(a) Demonstrates the testbed. We transfer edge data to Raspberry Pi via BLE. The local PC controls two industrial robotic arms over TCP/IP.



(b) The real image of the industrial robotic arm.



(c) The demonstration of the pick-and-place use case scenario. Both arms are in a home position at the beginning. The steel balls stand near the stopper. The scenario steps are as follows: **(I)** First, the arms grab the steel ball from the inclined platform. **(II)** Then, they drop the steel ball to the other inclined platform. **(III)** Steel balls roll down until the stopper. Each arm completes the process around 20 seconds (see §6.1.2 for further details).



(d) The arm joints and their rotations. While base, shoulder, and elbow joints provide larger movements, wrist joints provider finer movements. Tool flange is the part where we a□ach 2FG7 parallel gripper.

Fig. 3. Testbed and use case scenario.

strength in units of Tesla (T). These three sensors are used in attitude heading reference systems (AHRS) (also known as magnetic, angular rate, and gravity (MARG)) to de□ne an accurate 3D orientation [47]. Sensor fusion algorithms are applied to come up with accurate orientation representation. Euler angles and quaternions are two common parameters in this context. Euler angles su□er from gimbal lock which causes the loss of one degree of freedom. Thus, quaternion representations are preferred. Mahony [74] and Madgwick [73] are two popular AHRS □lters that de□ne orientation via quaternions. Madgwick □lter generates less root mean squared error (RMSE) while being computationally expensive in a negligible matter [71] in Adafruit and Arduino boards where we utilize open-source libraries[5,6]. We use proprietary libraries[7] developed by Bosch for the Nicla Sense ME where quaternions are generated via the Mahony algorithm. We compare the quality of the IMU data while also observing the quaternion generation to visually observe the stability of sensors (see Figure 4). We observe the following:

[5]github.com/adafruit/Adafruit_AHRS
[6]github.com/arduino-libraries/Arduino_LSM9DS1
[7]github.com/arduino/nicla-sense-me-fw

Table 7. Edge Development Board Testing

Edge Development Board	Arduino Nano 33 BLE Sense	Adafruit Feather nRF52840 Sense	Nicla Sense ME
Power Consumption (mAh) [Quaternion, *Raw Data]	[24.1, 24.2]	[12.9, 12.6]	[14.9, 15.7]
Sensor Type (Range & Sensitivity)	Acc. ([-4, 4] g & 0.122 mg) Gyro. ([-2000, +2000] dps & 70 mdps) Mag. ([-400, +400] μT & 0.014 μT)	Acc. ([-4, 4] g & 0.732 mg) Gyro. ([-2000, +2000] dps & 1 mpds) Mag. ([-400, +400] μT & 0.014 μT)	Acc. ([-4, 4] g & 0.239 mg) Gyro. ([-2000, +2000] & 30 mdps) Mag. ([±1300 (x, y), ±2500(z)] μT & 0.02 μT)

* By "Raw", we mean accelerometer, gyroscope, and magnetometer data. T: Tesla, dps: degrees per second, g: G-force. Acc: Accelerometer, Gyro: Gyroscope, Mag: Magnetometer. Ranges are the default ones.

- *Adafruit consumes less power overall*. Out of three edge development boards, the power consumption of Adafruit is signi□cantly lower than Arduino while being closer to Nicla. If we supply these boards with 9 Volts 250 mAh battery, we would expect the Adafruit to run around 20 hours, Nicla to run around 16 hours, and Arduino to run around 10 hours.
- *Nicla provides the most stable data.* As Adafruit and Arduino generate a higher noise, it is hard to judge if the resolution re□ects the actual change. However, analysis of gyroscope data revealed the existence of random spikes, which may introduce potential outliers to the data.

6.3 Dataset Generation and Characteristics

In this work, we change the arm's motion by modifying the joint velocity to create anomalies. We apply changes at di□erent magnitudes to evaluate the sensitivity of the proposed anomaly detection system. Thus, we have two states: *normal state* where the arm joints move at default velocity (1.05 rad/s), *anomalous state* where the arm joints move at various velocities. The anomalous state also has two phases: the □rst phase where the joint velocities are higher than the default, and the second phase where the opposite applies. The Table 8 demonstrates the anomalies with respect to time.

Table 8. The Generated Anomalies

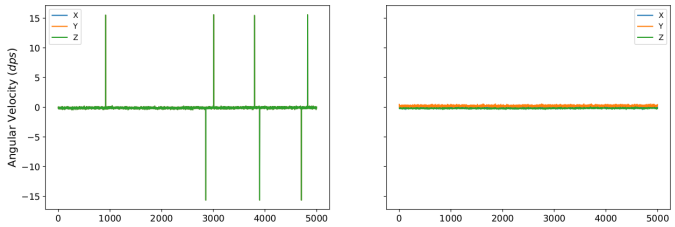
Time Interval (minutes*)	900-936	972-1008	1044-1080	1116-1152	1188-1224	1260-1296	1332-1368	1404-1440
Velocity Change	10% Increase	35% Increase	65% Increase	100% Increase	50% Decrease	5% Decrease	20% Increase	25% Decrease

*Whole test is 1460 minutes. The arm joints runs at normal velocity during non-mentioned time intervals.

In total, the CASPER dataset is a time series dataset containing four □les generated from a pick-and-place operation lasting around 24 hours: The □rst Comma Separated Values (CSV) □le consists of IMU data. We gather data via Nicla attached to one of the arms (see Figure 3b). The data include accelerometer, gyroscope, and magnetometer data. The second and the third □les (one □le per arm) contain built-in arm parameters (e.g., joint positions, velocities, and currents). We gather both data at 20Hz which corresponds 50 ms di□erence between two consecutive data points. The □nal □le is a PCAP containing the network tra□c between the local controller PC and the arms. Table 9 demonstrates the datasets while providing the feature names and characteristics. In this study, our focus is solely on the data generated by Nicla, as our objective is to investigate the e□ectiveness of an air-gapped IoT anomaly detection system. We share the built-in and network data for researchers who are working in related □elds.

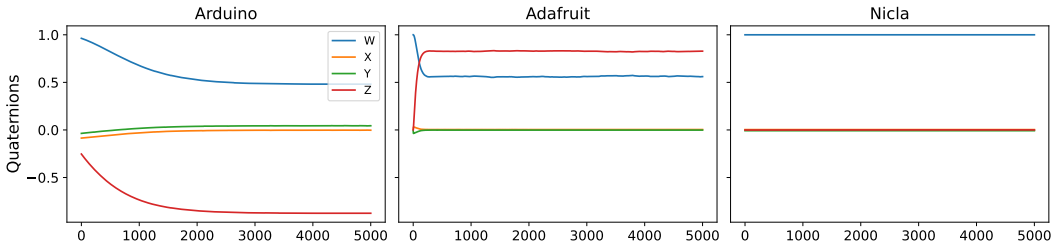
6.4 Anomaly Detection

Anomaly detection application on IMU data obtained from an edge development board attached to an industrial robotic arm that performs repetitive tasks contains the following challenges: (I) Each arm is idle for a certain period causing data to contain a high number of near-zero data points. This beclouds the use of common feature extraction methods for time series data, such as applying



(a) The gyroscope from Arduino generates random spikes when we query with magnetometer data. Thus, we applied a smoothing filter (moving median with a window length of three) to eliminate these. The graph on the le is without the filter.

(b) We generated three sample datasets with 5000 data points at 20Hz to observe the behavior of IMU sensors of each edge board. We applied the available calibration methods (the methods provided in open-source code repositories) and have not tweaked the source codes. Our findings show that Nicla generates less noisy data overall.



(c) We generated quaternion data from each edge development board. The comparison shows that Nicla generates the most stable quaternion data while Adafruit and Arduino are subjected to initial dri.

Fig. 4. Edge data generation comparison.

Table 9. The CASPER Dataset

Data	Features	Number of Data Points/Packets	Size
Nicla - IMU	Accelerometer (x, y, z) Gyroscope (x, y, z) Magnetometer (x, y, z)	1750932	138.9 MB
Arm Parameters*	Timestamp Joint Positions Joint Velocities Joint Currents Joint Voltages Cartesian Coordinates Generalized Forces Joint Temperatures Execution Time Safety Status Norm of Cartesian Linear Momentum Robot Current Tool Acceleration Tool Current Tool Temperature Tool velocity Elbow Position Elbow Velocity TCP Force Anomaly State	1762650	2.0 GB
Network	267**	14582826	3.7 GB

*: This is for only one single arm, we have two arms in total. **: This is the number of common TCP features that can be extracted from the pcap file. The total number of available features (wireshark.org/docs/dfref) are a lot more.

rolling mean/median to input windows. (II) IMU data by nature contain highly correlated features, which can lead to unstable predictions generated by less reliable models due to multicollinearity. (III) There is a possibility of label mismatching. We modify the joint velocity of the arms via a controller PC. However, the data that we apply anomaly detection to is generated via a different source (an edge development board). Hence, we also utilize one of the features (X-axis of a gyroscope) where anomalies are obvious to generate accurate anomaly labels. Figure 5 presents the IMU data generated by Nicla where we can spot the anomalies on the aforementioned feature. The anomaly detection methodology as follows: The dataset is divided into two sets, non-anomalous and anomalous, and the optimization of anomaly detection algorithms is done on the non-anomalous set where we target the minimized loss (RMSE) without overfitting the models. Then, anomalous windows are inputted into these optimized models where window labeling is performed through thresholding where thresholds are determined via grid search. The performance of these models is then evaluated using the confusion matrix, and relevant performance metrics (accuracy, recall, F1 score, and precision) are generated.

6.4.1 Feature Processing. We employ several feature processing techniques. First, we remove some of the noise by applying rolling median filter (see Fig. 6). The optimal window length for the filter is found via grid search considering the trade-off between information loss and noise reduction. We apply z-score normalization to the data-driven models only, by fitting the models exclusively with the training data to prevent the validation/test data from having access to any training data characteristics.

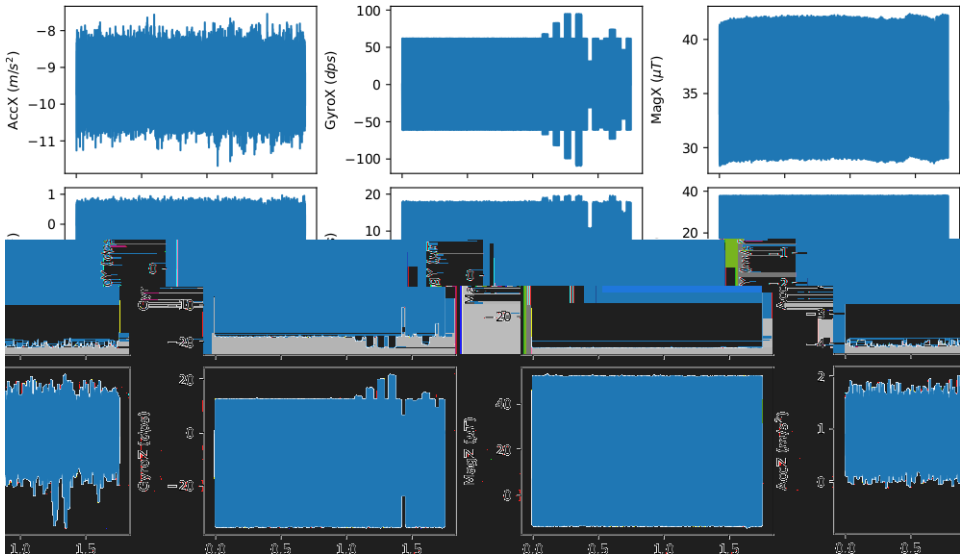


Fig. 5. Demonstrates IMU data generated via an edge development board a□ached to an industrial robotic arm. We can easily see that the anomalies reflect on the X-axis of gyroscope data.

Table 10. Autocorrelation Analysis

Feature	AccX	AccY	AccZ	GyroX	GyroY	GyroZ	MagX	MagY	MagZ
$r*$, $w**$	0.995, 755	0.998, 755	0.977, 755	0.997, 755	0.996, 755	0.995, 755	0.998, 755	0.999, 755	0.999, 755

r*: The Pearson correlation coe□cient. w**: Window length.

6.4.2 Correlation Analysis. We apply autocorrelation to □nd the highest time-dependent Pearson correlation coe□cient (r) denoted as ρ where E is the expected value, μ is the mean and σ is the standard deviation (see Equation 1) to □nd the periodicity. We utilize the period as an input window length for baseline and data-driven approaches. All input features autocorrelate most when the window length is set to 755 data points (see Table 10). We also analyze how features (sets of features) correlate with each other due to the aforementioned reasons. We make the following observations from the feature correlation heatmap (see Figure 7), and canonical-correlation analysis (CCA) (see Table 11): (I) The X and Y-axes of the accelerometer are the most correlated features followed by the Y-axes of accelerometer and magnetometer. (II) Gyroscope features do not correlate with others. (III) The accelerometer and gyroscope features are the least correlated features. (IV) CCA shows that the overall, accelerometer and magnetometer features correlate. As correlated input features are undesired, we also investigate the correlation of the quaternion representation of IMU data. We see two main advantages of utilizing quaternions over raw IMU: (I) The transformation reduces the number of input features from 9 to 4, (II) the quaternions generated via the Madgwick algorithm do not show any collinearity on the contrary of Mahony algorithm. Figure 8 compares the correlation heatmap of quaternions generated by both algorithms.

$$\rho_{XX}(t_1, t_2) = \frac{E[(X_{t_1}) - (\mu_{t_1})\overline{(X_{t_2}) - (\mu_{t_2})}]}{\sigma_{t_1}\sigma_{t_2}} \tag{1}$$

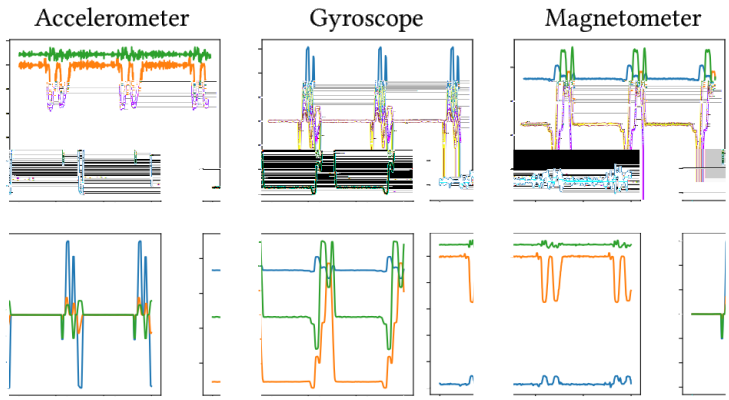


Fig. 6. Demonstrates the e□ect of noise removal on all features. The bo□om three figures are the noise-removed data.

Table 11. Canonical-correlation Analysis

Accelerometer - Gyroscope	Accelerometer - Magnetometer	Gyroscope - Magnetometer
[0.48561, 0.07371, 0.02834]	[0.96962, 0.58022, 0.27068]	[0.41173, 0.30430, 0.07603]

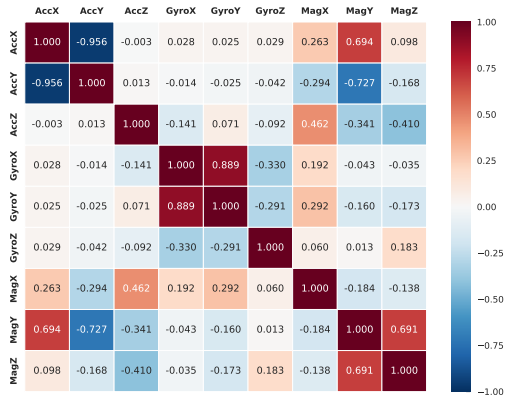
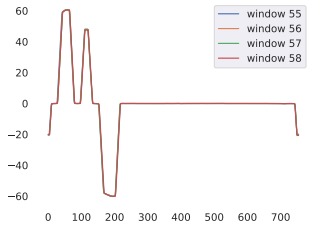


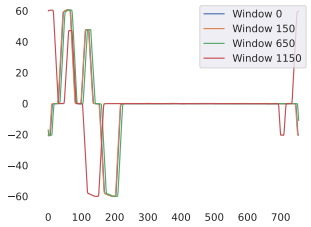
Fig. 7. The correlation of input features. We see that several features are highly correlated (e.g., X and Y-axes of accelerometer. This is expected due to the nature of IMU data.

(a) Correlation heatmap of Madgwick □ aternions. (b) Correlation heatmap of Mahony □ aternions.

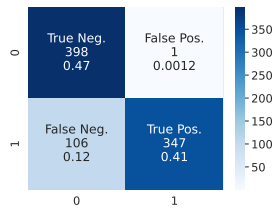
Fig. 8. A comparison of correlation heatmaps of two common quaternion algorithms.



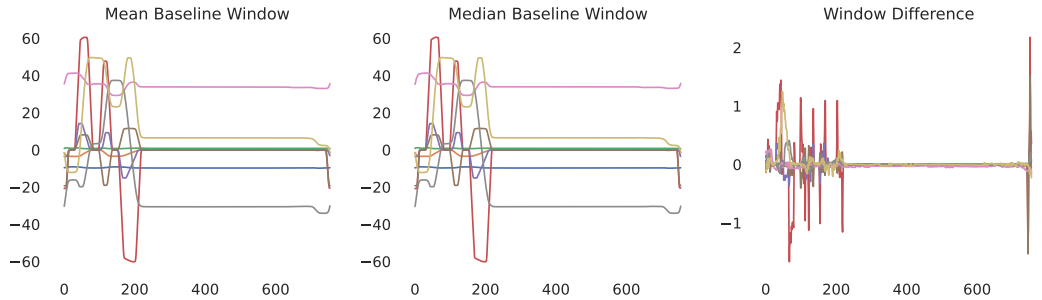
(a) Consecutive windows.



(b) Random windows.



(c) Confusion matrix.



(d) Comparison of baseline input windows.

Fig. 9. The lag is obvious as the gap between the window increases. Mean baseline RMSE is 0.3909, while the median one is 0.3999. Hence, mean baseline performs be□er than the medium baseline with metrics of 84.6% accuracy and 83.4% F1 score.

6.4.3 Baseline. We utilize a statistical baseline to establish a minimum level of con□dence to be beaten by the data-driven approaches to justify their use. We divide the data into input windows (generated from non-anomalous data only) where the window length is the period found via the aforementioned correlation analysis. The comparison of input windows reveals the random unknown lag proving the deviations on the sampling rate set to 20Hz during the experiment. The lag is minimal in consecutive windows (ranging from -3 to 3 data points) but expands over time causing a delay of around a quarter of the window length. To come up with a stronger baseline, we eliminate the lag by considering the □rst window (755 data points) as the base window. We generate the mean and median windows. Then, we calculate the overall RMSE (see Equation 2).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

for both where y_i is the actual and $\hat{y}_i^2$ is the predicted value. The mean baseline beats the median one hence used to detect anomalies via thresholding based on RMSE. We measure the performance of anomaly detection methods via a confusion matrix consisting of four main parameters: (I) True positives (TP)-when an anomaly is detected as an anomaly, false positives (FP)-when normal is detected as an anomaly, true negatives (TN)-when normal is detected as normal, false negatives (FN)-when normal is detected as an anomaly. We calculate performance metrics which are accuracy, precision, F1-score, and recall via these parameters as shown below. Figure 9 demonstrates the lag, mean, and median baselines and their di□erence, and confusion matrix of baseline.

Algorithm 1 Anomaly Detection Algorithm for Data-driven Approaches

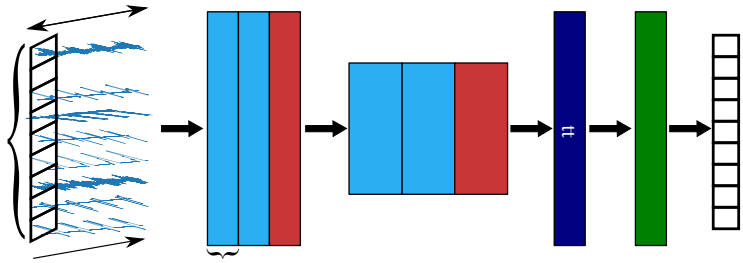
Input: Test data $X \in \mathbb{R}^{=\times 9}$, $\mu_{CA08=8=6} \in \mathbb{R}^{1\times 9}$, $\sigma_{CA08=8=6} \in \mathbb{R}^{1\times 9}$, threshold list $T \in \mathbb{R}^{:}$
Output: List $P \in 0, 1^{:}$, where $l = m - 755 + 1$, where $m = n - 755$, where $n = |X|$,

$\hat{X} = \frac{- - ` CA08=8=6}{f_{CA08=8=6}}$ ▷ Normalize test data via training parameters
$W \in \mathbb{R}^{755\times 9}$ ▷ Initialize a sliding window with size 755
$R = [\,],\ S = [\,],\ P = [\,]$ ▷ Initialize empty lists for RMSE values, RMSE rolling sums and final labels
for $i = 1$ to $n - 755$ **do**
  $W = \hat{X} i : i + 754, :$ ▷ Select the i^{C} window of test data
  $\hat{y} = f_{1D\text{-}CNN}(W) \in \mathbb{R}^{1\times 9}$ ▷ Predict the next point using 1D-CNN model
  $y = \hat{y} \cdot \sigma + \mu \in \mathbb{R}^{1\times 9}$ ▷ Inverse normalize the predicted value
  $r_8 = \sqrt{\frac{1}{9} \sum_{9=1}^{9} (y_{8\bullet 9_{\text{target}}} - y_{8\bullet 9})^2}$ ▷ Calculate RMSE per time step
  $R \leftarrow [r_8]$ ▷ Append to RMSE list
end for
$S_8 = \sum_{9=8-,\ +1}^{8} R_9$ for $i = W, W + 1, \ldots, |R|$ ▷ Apply rolling sum for RMSEs with window length W
for $j \leftarrow 1$ to $|T|$ **do** ▷ Generate a prediction label list via thresholding
  $P \leftarrow [\,]$
  for $i \leftarrow 1$ to $|S| - W + 1$ **do**
    if $S_8 > T_9$ **then**
      $P \leftarrow P + [1]$
    else
      $P \leftarrow P + [0]$
    end if
  end for
end for

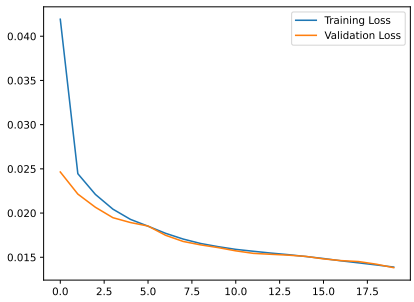
6.4.4 Partial Least Squares regression. Due to the high correlation of input features, we investigate the feasibility of using Partial Least Squares regression as an anomaly detection method. PLS reduces the number of predictors to 7 capturing around 99% of the variation of the data where the correlations between the predictors are near-zero. The computational complexity of PLS is far less than data-driven approaches. While the loss (RMSE) is similar to data-driven approaches, the PLS fails to generate relatively high RMSEs when the input consists of anomalous points.

6.4.5 1D convolutional neural network. We design a 1D-CNN-based ML algorithm to detect anomalies. We expand the receptive field by stacking two 1d-CNN layers to extract deeper local/temporal features. These layers are followed by a max pooling layer that makes the model more robust to overfitting. Finally, we output our features via the fully connected layer. We are implementing a sliding window approach in which the input window consists of 755-time steps (window length), while the output window consists of only 1-time step, then we shift by 1-time step. We do not manually eliminate any lags as we have done for the baseline. Rectified Linear Unit (ReLU) is used as an activation function. The best hyperparameters are found via grid search. We follow the same approach for the anomaly labels. The sliding windows with more anomaly points are accepted as anomalous (see Algorithm 1). We see that the 1D-CNN beats the baseline by a high margin. Figure 10 demonstrates the model architecture, hyperparameters tried during the grid search, and the related confusion matrix.

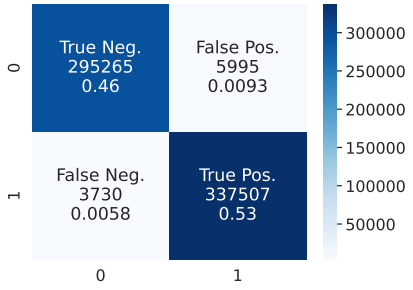
6.4.6 Long Short-Term Memory recurrent neural network. As LSTMs are de facto choice for many application where time series data are present due to their ability to "remember" past inputs, we compare the proposed approach to a LSTM-based approach. The LSTM model comprises a single LSTM layer followed by two dense layers with rectified linear unit (ReLU) activation. In order to address overfitting, which is commonly observed with LSTMs, we introduce a dropout layer and L1 regularization on the LSTM layer. Despite these measures, we observe overfitting,



(a) The architecture of the 1D-CNN model and the utilized hyperparameters.



(b) The loss graph of the model.



(c) The confusion matrix for the 1D-CNN-based anomaly detection model.

Fig. 10. Demonstrates the neural network architecture, loss graph, and the confusion matrix. One epoch takes around 4 minutes for the final chain of cross-validation.

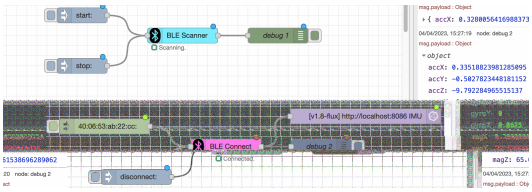
which we attribute to the simplicity of the input data. The loss graph, along with corresponding hyperparameters, is available in our GitHub repository.

6.4.7 XGBoost. Among decision tree regressors, we adopt the XGBoost which is a state-of-the-art boosting algorithm. We specify the mean squared error loss function and train the algorithm through the use of 10-fold forward chaining cross-validation. Experimental results reveal that XGBoost is capable of achieving comparable performance, even when trained on just 10% of the data corresponding to the first fold of cross-validation, while also boasting greater computational efficiency than its neural network counterparts. Notably, we implement Algorithm 1 with a singular modification, wherein we shift data with window length generating only two windows (input, and target which is the window length shifted version of input) instead of traditional sliding windowing that we implemented on 1D-CNN. This is necessary as tree-based algorithms rely on 2-dimensional inputs. Optimal hyperparameters, including the number of estimators and the maximum depth, are selected via grid search. We do not manually eliminate the lag as we have done for the baseline.

6.4.8 Comparison of anomaly detection methods. Table 12 compares the performances of implemented anomaly detection systems on IMU data. The statistical baseline, where we manually eliminate the lag to make it stronger, performs well, indicating its usefulness for applications where

Table 12. Comparison of Anomaly Detection Approaches

Approach	RMSE	Accuracy	Recall	Precision	F1 Score
Baseline	0.3909	0.8744	0.7660	0.9971	0.8664
PLS	0.1586	0.49	0.49	0.52	0.51
1D-CNN	0.1175	0.9848	0.9890	0.9825	0.9857
XGBoost	0.1301	0.9782	0.9900	0.9695	0.9797
LSTM	0.0947	0.8661	0.7510	0.9960	0.8563



(a) A Node-RED setup enabling BLE communication.

(b) Real-time IMU data visualized via Grafana dashboard.

Fig. 11. The demonstration of an IoT supervision system.

the accuracy of anomaly detection is not signi□cantly critical as seen in the industrial domain. All other approaches beat the statistical baseline in terms of predicting normal data. PLS, which is the computationally least expensive approach, fails at anomaly detection as the anomaly inputs generate similar outputs causing small losses and rendering thresholding approaches ine□ective in distinguishing between normal and anomalous data. This approach is better suited for forecasting non-anomalous data only. The LSTM model performs worse (fails to detect most of the 5% decrease in joint velocity anomalies) than its counterparts due to over□tting despite the introduced counter-measures such as the dropout layer and L1 regularization. The 1D-CNN and XGBoost perform the best while 1D-CNN slightly provides better results.

6.5 IoT Supervision System

Our analysis of recent cyber incidents (see Section 2) demonstrates the signi□cance of having supervisory systems. In this work, as we assume that the integrity of built-in data is corrupted, we provide an IoT supervision system for the IMU data generated via the edge development board. We transfer edge data to a fog device over BLE (Nicla Sense ME contains an nRF52832 microcontroller o□ering BLE 4.2 connectivity.) that provides encrypted transmission. We utilize Node-RED, which is an open-source □ow-based programming tool. We developed a Node-RED package[8] that provides nodes (one for scanning, one for transmission) to establish a BLE communication between the edge and the fog device. We utilize Arduino IDE to upload an Arduino sketch (provided in the GitHub repository) which enables BLE transmission from an edge device. The transmitted data are registered to In□uxDB. We query this data via the Grafana server that runs on the fog device and

[8]https://www.npmjs.com/package/node-red-contrib-ble-sense

demonstrate on the Grafana dashboard hence providing a real-time live stream of IMU data. Figure 11 demonstrates the Node-RED setup and the Grafana dashboard.

6.6 Discussions

Undesired delay due to lack of control. We utilized two UR3e industrial robotic arms classified as collaborative robots equipped with a control box and an HMI (known as a teach pendant). The intended use of the manufacturer for this arm involves control through the teach pendant limiting synchronization with other industrial edge components such as additional robotic arms or conveyor belts. To address this issue, the manufacturers developed a custom protocol, known as Real Time Data Exchange (RTDE), which enables remote control. This protocol relies on the Python socket library[9], which provides TCP/IP communication. However, due to the limited control over delay offered by the library, the local PC and both robotic arms were not entirely synchronized during the experiment, which resulted in undesired delays.

Matching anomaly labels from a different data source. The anomalies are created via the local controller PC which also generates the built-in data. The anomaly detection is done on the data generated from an attached edge development board. Both data-generating processes (fixed at 20Hz) are independent of each other. Due to mismatching lengths of these two data occurring due to the edge development board not running at 20Hz exactly, we utilize one of the features where the anomalies are obvious to generate correct anomaly labels. This requires manual identification of the drift and the obvious presence of anomalous behavior on a certain feature which might not be the case for all scenarios.

Correlated input features due to nature of an IMU data. The correlation of IMU features is expected as they define the aspects of motion. Our correlation analysis demonstrates that the accelerometer and magnetometer features exhibit a high correlation for the pick-and-place use case scenario. This finding highlights the effectiveness of the proposed 1D-CNN-based model even in the presence of highly correlated input features. As our future work aims to run this model on an edge development board, we have analyzed the feature correlation of quaternion representations which consists of only four features allowing us to reduce computational complexity. Our analysis shows that Madgwick quaternions are less correlated than Mahony quaternions making them more promising for our research work with the current dataset.

Realistic data with high number of zeros. In industrial environments, it is common for edge actuators to remain idle during periods of cooperation. In our investigation, we simulated an environment where two industrial robotic arms operated consecutively, resulting in a dataset with a large number of near-zero values. Disregarding these values is not feasible, as anomalies can be identified through variations in idle time. However, the presence of a high number of near-zero values presents two significant challenges: (I) Traditional feature extraction methods for time series data (e.g., mean, median, kurtosis, and skewness) lose their validity. (II) Window sampling based on the highest Pearson correlation coefficient can produce unaligned windows, necessitating manual lag elimination for approaches that require aligned windows.

Grid search to find optimal hyperparameters and thresholds. Grid search is a commonly used approach for identifying optimal hyperparameters in data-driven methods. However, the computational complexity of this technique increases exponentially with each additional parameter, rendering the process time-consuming. Since grid search is often conducted manually, there is a possibility of human error. Despite guidelines for conducting grid search effectively, there remains a need for a more optimized methodology for initializing and accurately estimating the best parameters. This issue is also relevant when determining the most appropriate threshold for anomaly

[9]https://docs.python.org/3/library/socket.html

detection implemented via forecasting. Therefore, it is crucial to explore novel methodologies that enable more e□cient and reliable hyperparameter optimization and anomaly threshold estimation.

Cause independent cyber-physical detection. The proposed 1D-CNN model demonstrates the ability to detect the smallest anomaly introduced in the experiment, a 5% reduction in joint velocities. 1D-CNN layers trained on non-anomalous data can extract discriminative features that capture the precise pattern of the time series data in a way that when the input (predictor) consists of anomalies the output (response) is disrupted enough to be detected through thresholding. As a result, the proposed approach's performance is independent of the cause of an anomaly, whether it be due to a cyberattack, aging, power failure, or a physical accident. This approach is vulnerable to adversarial attacks if the adversary gains control over the industrial robotic arm during the training process which is unrealistic, given the accuracy requirements of industrial applications, any unexpected physical deviation would likely have been detected by the relevant sta□, leading to a halt in training/production.

Continuous anomalous runs longer than the input window. The proposed baseline approach relies on a static sample window generated through averaging non-anomalous windows. This approach neglects the temporal correlations present in the data. Instead, a stronger baseline approach that accounts for these correlations involves averaging the root-mean-square errors (RMSEs) of consecutive windows. This method can e□ectively detect the beginning of an anomalous run but is prone to failure when the input window contains anomalous points. Similarly, linear regression methods are sensitive to anomalous data, as such data can skew the regression line. Data-driven approaches that are able to learn non-anomalous feature representations of sequence data are more robust to anomalous inputs. These models may struggle to predict anomalous data accurately since it deviates from the learned pattern during training, leading to higher RMSE enabling the detection of anomalous windows via thresholding. As an example of such a data-driven approach for anomaly detection, the proposed 1D-CNN model o□ers promising results.

7 CONCLUSIONS AND FUTURE WORK

The transition of IT to OT is increasingly continuing thus allowing the development of smart manufacturing systems where ubiquitous networking technologies are utilized. This causes an increased attack surface thus bringing new vulnerabilities which are exploitable as we can see from the recent industrial cyber incidents. The cyber-only solutions are inadequate as attackers that target industrial domains are capable of performing sophisticated attacks that can deceive monitoring (e.g., network-based IDSs) systems. As the motivation of these attacks is to cause the highest damage, attackers try to manipulate the physical processes via cyber-only access. To prevent such incidents, we do require cyber-physical defense mechanisms utilizing a segregated network. In this work, we proposed a ubiquitous cyber-physical anomaly detection system that detects movement-based anomalies of an industrial robotic arm. According to our experimental results, a 1D-CNN-based deep learning model is capable of accurately learning the behavior of time series (sequential) data even when the input features are highly correlated as the proposed model can even detect a 5% decrease in the joint velocities which is the minimal applied deviation. We also propose an open-source IoT monitoring system that utilizes BLE to transmit edge data via the developed Node-RED package. We expect our work to encourage the exploration of 1D-CNNs for time series data as they are computationally more bene□cial than their counterpart RNNs. Future work includes several improvements to CASPER: increasing the scope of the work via introducing new anomalies (e.g., adding additional weight, touching the arm, shaking the testbed), online learning via cloud, use of quaternions, and detecting anomalies at the edge (on the edge development board) to prevent delay.

REFERENCES

[1] Adafruit. 2021. *Adafruit Feather nRF52840 Sense*. Adafruit Industries. Retrieved Nov 12, 2022 from https://learn.adafruit.com/adafruit-feather-sense

[2] Mohiuddin Ahmed and Abdun Naser Mahmood. 2014. Network traffic analysis based on collective anomaly detection. In *2014 9th IEEE Conference on Industrial Electronics and Applications*. IEEE, Hangzhou, China, 1141–1146.

[3] Mohiuddin Ahmed and Abdun Naser Mahmood. 2015. Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection. *Annals of Data Science* 2, 1 (2015), 111–130.

[4] Safaa Allamy and Alessandro Lameiras Koerich. 2021. 1D CNN architectures for music genre classification. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Orlando, FL, USA, 01–07.

[5] Matthew G Angle, Stuart Madnick, James L Kirtley, and Shaharyar Khan. 2019. Identifying and anticipating cyberattacks that could cause physical damage to industrial control systems. *IEEE Power and Energy Technology Systems Journal* 6, 4 (2019), 172–182.

[6] R Ani, S Krishna, N Anju, M Sona Aslam, and OS Deepa. 2017. Iot based patient monitoring and diagnostic prediction tool using ensemble classifier. In *2017 International conference on advances in computing, communications and informatics (ICACCI)*. IEEE, Udupi, 1588–1593.

[7] Apple. 2022. *Track your sleep with Apple Watch*. Apple Inc. Retrieved January 14, 2022 from https://support.apple.com/en-gb/guide/watch/apd830528336/watchos

[8] R Ganesh Babu, P Karthika, and V Aravinda Rajan. 2019. Secure IoT systems using raspberry Pi machine learning artificial intelligence. In *International conference on computer networks and inventive communication technologies* (Coimbatore, India). Springer, Cham, Switzerland, 797–805.

[9] Barış Bayram, Taha Berkay Duman, and Gökhan Ince. 2021. Real time detection of acoustic anomalies in industrial processes using sequential autoencoders. *Expert Systems* 38, 1 (2021), e12564.

[10] Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit: using wearable sensors to detect eating episodes in unconstrained environments. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1, 3 (2017), 1–20.

[11] Edgar A Bernal, Xitong Yang, Qun Li, Jayant Kumar, Sriganesh Madhvanath, Palghat Ramesh, and Raja Bala. 2017. Deep temporal multimodal fusion for medical procedure monitoring using wearable sensors. *IEEE Transactions on Multimedia* 20, 1 (2017), 107–118.

[12] Anatolij Bezemskij, George Loukas, Richard J Anthony, and Diane Gan. 2016. Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle. In *2016 15th international conference on ubiquitous computing and communications and 2016 international symposium on cyberspace and security (IUCC-CSS)*. IEEE, Granada, Spain, 61–68.

[13] Monica Bianchini and Franco Scarselli. 2014. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems* 25, 8 (2014), 1553–1565.

[14] Ekaba Bisong. 2019. *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. Apress, OTTAWA, Canada.

[15] BleepingComputer. 2021. *Sierra Wireless resumes production after ransomware attack*. BleepingComputer. Retrieved Nov 12, 2022 from https://www.bleepingcomputer.com/news/security/sierra-wireless-resumes-production-after-ransomware-attack/

[16] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. 2021. Neural contextual anomaly detection for time series. arXiv:2107.07702 [cs.LG]

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

[18] Tingting Chen, Xueping Liu, Bizhong Xia, Wei Wang, and Yongzhi Lai. 2020. Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. *IEEE Access* 8 (2020), 47072–47081.

[19] Heeryon Cho and Sang Min Yoon. 2018. Divide and conquer-based 1D CNN human activity recognition using test data sharpening. *Sensors* 18, 4 (2018), 1055.

[20] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, Providence, RI, USA, 3642–3649.

[21] Amazon Elastic Compute Cloud. 2011. Amazon web services. *Retrieved November* 9, 2011 (2011), 2011.

[22] Armando W Colombo, Stamatis Karnouskos, Okyay Kaynak, Yang Shi, and Shen Yin. 2017. Industrial cyberphysical systems: A backbone of the fourth industrial revolution. *IEEE Industrial Electronics Magazine* 11, 1 (2017), 6–16.

[23] Comunicaffè. 2021. *Caffitaly, gli hacker all'assalto delle capsule di Gaggio*. Caffitaly. Retrieved 2021-05-30 from https://www.comunicaffe.it/caffitaly-gli-haker-allassalto-delle-caspule-di-gaggio-montano/

[24] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, et al. 2020. Tensorflow lite micro: Embedded machine learning on tinyml systems. *arXiv*

preprint arXiv:2010.08678 (2020).

[25] Essam Debie, Raul Fernandez Rojas, Justin Fidock, Michael Barlow, Kathryn Kasmarik, Sreenatha Anavatti, Matt Garratt, and Hussein A Abbass. 2019. Multimodal fusion for objective assessment of cognitive workload: a review. *IEEE transactions on cybernetics* 51, 3 (2019), 1542–1555.

[26] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Arti cial Intelligence*, Vol. 35. AAAI Press, Virtual, 4027–4035.

[27] Taha Berkay Duman, Barış Bayram, and Gökhan İnce. 2019. Acoustic anomaly detection using convolutional autoencoders in industrial processes. In *International Workshop on Soft Computing Models in Industrial and Environmental Applications* (Seville, Spain). Springer, Cham, Switzerland, 432–442.

[28] Sinem Coleri Ergen. 2004. ZigBee/IEEE 802.15. 4 Summary. *UC Berkeley, September* 10, 17 (2004), 11.

[29] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. 2016. Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. arXiv:1612.06676 [cs.LG]

[30] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. 2005. An Approach to Spacecraft Anomaly Detection Problem Using Kernel Feature Space. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (Chicago, Illinois, USA) *(KDD '05)*. Association for Computing Machinery, New York, NY, USA, 401–410. https://doi.org/10.1145/1081870.1081917

[31] Yang Gao, Borui Li, Wei Wang, Wenyao Xu, Chi Zhou, and Zhanpeng Jin. 2018. Watching and safeguarding your 3D printer: Online process monitoring against cyber-physical attacks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–27.

[32] Zhiwei Gao, Carlo Cecati, and Steven X Ding. 2015. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE transactions on industrial electronics* 62, 6 (2015), 3757–3767.

[33] Aurélien Géron. 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Canada.

[34] Mohammed Ghazal, Tasnim Basmaji, Maha Yaghi, Mohammad Alkhedher, Mohamed Mahmoud, and Ayman S El-Baz. 2020. Cloud-Based Monitoring of Thermal Anomalies in Industrial Environments Using AI and the Internet of Robotic Things. *Sensors* 20, 21 (2020), 6348.

[35] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. 2017. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, Singapore, 140–145.

[36] Victor Gonzalez-Huitron, José A León-Borges, AE Rodriguez-Mata, Leonel Ernesto Amabilis-Sosa, Blenda Ramírez-Pereda, and Hector Rodriguez. 2021. Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Computers and Electronics in Agriculture* 181 (2021), 105951.

[37] Naveen Goud. 2001. *Altran hit by a Cyber Attack and Ransomware is suspect*. Cybersecurity Insiders. Retrieved Nov 12, 2022 from https://www.cybersecurity-insiders.com/altran-hit-by-a-cyber-attack-and-ransomware-is-suspect/

[38] Haodong Guo, Ling Chen, Liangying Peng, and Gencai Chen. 2016. Wearable sensor based multimodal human activity recognition exploiting the diversity of classi□er ensemble. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, New York, NY, USA, 1112–1123.

[39] Kevin Gurney. 2018. *An introduction to neural networks*. CRC press, London, UK.

[40] Juan Haladjian, Daniel Schlabbers, Sajjad Taheri, Max Tharr, and Bernd Bruegge. 2020. Sensor-Based Detection and Classi□cation of Soccer Goalkeeper Training Exercises. *ACM Transactions on Internet of Things* 1, 2, Article 12 (apr 2020), 20 pages. https://doi.org/10.1145/3372342

[41] Danfeng Hong, Naoto Yokoya, Gui-Song Xia, Jocelyn Chanussot, and Xiao Xiang Zhu. 2020. X-ModalNet: A semi-supervised deep cross-modal network for classi□cation of remote sensing data. *ISPRS Journal of Photogrammetry and Remote Sensing* 167 (2020), 12–23.

[42] Yan Hu, An Yang, Hong Li, Yuyan Sun, and Limin Sun. 2018. A survey of intrusion detection on industrial control systems. *International Journal of Distributed Sensor Networks* 14, 8 (2018), 1550147718794615.

[43] Turker Ince, Serkan Kiranyaz, Levent Eren, Murat Askar, and Moncef Gabbouj. 2016. Real-time motor fault detection by 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics* 63, 11 (2016), 7067–7075.

[44] Jun Inoue, Yoriyuki Yamagata, Yuqi Chen, Christopher M Poskitt, and Jun Sun. 2017. Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, New Orleans, LA, USA, 1058–1065.

[45] Rolf Isermann. 1997. Supervision, fault-detection and fault-diagnosis methods—an introduction. *Control engineering practice* 5, 5 (1997), 639–652.

[46] Rolf Isermann. 2005. Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in control* 29, 1 (2005), 71–85.

[47] Tariqul Islam, Md Saiful Islam, Md Shajid-Ul-Mahmud, and Md Hossam-E-Haider. 2017. Comparison of complementary and Kalman Filter based data fusion for attitude heading reference system. In *AIP Conference Proceedings* (Dhaka, Bangladesh), Vol. 1919. AIP Publishing LLC, New York, NY, USA, 020002.
[48] Gopal Chandra Jana, Ratna Sharma, and Anupam Agrawal. 2020. A 1D-CNN-spectrogram based approach for seizure detection from EEG signal. *Procedia Computer Science* 167 (2020), 403–412.
[49] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. 2019. Multivariate LSTM-FCNs for time series classification. *Neural networks* 116 (2019), 237–245.
[50] Hakan Kayan, Yasar Majib, Wael Alsafery, Mahmoud Barhamgi, and Charith Perera. 2021. AnoML-IoT: An end to end re-configurable multi-protocol anomaly detection pipeline for Internet of Things. *Internet of Things* 16 (2021), 100437.
[51] Hakan Kayan, Matthew Nunes, Omer Rana, Pete Burnap, and Charith Perera. 2022. Cybersecurity of Industrial Cyber-Physical Systems: A Review. *ACM Comput. Surv.* 54, 11s, Article 229 (sep 2022), 35 pages.
[52] Haider Adnan Khan, Nader Sehatbakhsh, Luong N Nguyen, Robert L Callan, Arie Yeredor, Milos Prvulovic, and Alenka Zajić. 2019. IDEA: Intrusion detection through electromagnetic-signal analysis for critical embedded and cyber-physical systems. *IEEE Transactions on Dependable and Secure Computing* 18, 3 (2019), 1150–1163.
[53] Daniel Knight. 2021. *DietPi OS*. DietPi. Retrieved Nov 12, 2022 from https://dietpi.com/
[54] Moshe Kravchik and Asaf Shabtai. 2018. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*. ACM, New York, NY, USA, 72–83.
[55] Andrew Kusiak. 2018. Smart manufacturing. *International Journal of Production Research* 56, 1-2 (2018), 508–517.
[56] Prasanth Lade, Rumi Ghosh, and Soundar Srinivasan. 2017. Manufacturing analytics and industrial internet of things. *IEEE Intelligent Systems* 32, 3 (2017), 74–79.
[57] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy* 9, 3 (2011), 49–51.
[58] Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. 2009. Exploring strategies for training deep neural networks. *Journal of machine learning research* 10, 1 (2009), 1–40.
[59] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. 2014. Industry 4.0. *Business & information systems engineering* 6, 4 (2014), 239–242.
[60] Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.
[61] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks* (Munich, Germany). Springer, Cham, Switzerland, 703–716.
[62] Guangxia Li, Yulong Shen, Peilin Zhao, Xiao Lu, Jia Liu, Yangyang Liu, and Steven CH Hoi. 2019. Detecting cyberattacks in industrial control systems using online learning algorithms. *Neurocomputing* 364 (2019), 338–348.
[63] Zhe Li, Jingyue Li, Yi Wang, and Kesheng Wang. 2019. A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment. *The International Journal of Advanced Manufacturing Technology* 103, 1 (2019), 499–510.
[64] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–21.
[65] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications* 36, 1 (2013), 16–24.
[66] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. IEEE, Pisa, Italy, 413–422.
[67] Qi Liu, Rudy Klucik, Chao Chen, Glenn Grant, David Gallaher, Qin Lv, and Li Shang. 2017. Unsupervised detection of contextual anomaly in remotely sensed data. *Remote Sensing of Environment* 202 (2017), 75–87.
[68] Marc Moreno Lopez and Jugal Kalita. 2017. Deep Learning applied to NLP. arXiv:1703.03091 [cs.CL]
[69] George Loukas, Tuan Vuong, Ryan Heartfield, Georgia Sakellari, Yongpil Yoon, and Diane Gan. 2018. Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning. *IEEE Access* 6 (2018), 3491–3508.
[70] Huimin Lu, Yujie Li, Shenglin Mu, Dong Wang, Hyoungseop Kim, and Seiichi Serikawa. 2017. Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE internet of things journal* 5, 4 (2017), 2315–2322.
[71] Simone A Ludwig, Kaleb D Burnham, Antonio R Jiménez, and Pierre A Touma. 2018. Comparison of attitude and heading reference systems using foot mounted MIMU sensor data: Basic, Madgwick, and Mahony. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018* (Denver, Colorado, United States), Vol. 10598. SPIE, Washington, DC, USA, 644–650.
[72] Chunjie Luo, Fan Zhang, Cheng Huang, Xingwang Xiong, Jianan Chen, Lei Wang, Wanling Gao, Hainan Ye, Tong Wu, Runsong Zhou, et al. 2018. AIoT bench: towards comprehensive benchmarking mobile and embedded device intelligence. In *International Symposium on Benchmarking, Measuring and Optimization* (Seattle, WA, USA). Springer,

Cham, Switzerland, 31–35.
[73] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. 2011. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE international conference on rehabilitation robotics*. IEEE, Zurich, Switzerland, 1–7.
[74] Robert Mahony, Tarek Hamel, and Jean-Michel P□imlin. 2008. Nonlinear complementary □lters on the special orthogonal group. *IEEE Transactions on automatic control* 53, 5 (2008), 1203–1218.
[75] Pankaj Malhotra, Lovekesh Vig, Gautam Shro□, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. In *European Symposium on Arti cial Neural Networks, Computational Intelligence and Machine Learning*, Vol. 89. IEEE, Bruges, Belgium, 89–94.
[76] S Manimurugan. 2021. IoT-Fog-Cloud model for anomaly detection using improved Naïve Bayes and principal component analysis. *Journal of Ambient Intelligence and Humanized Computing* (2021), 1–10.
[77] Mark Masluch. 2021. *Bombardier Statement on Cybersecurity Breach*. Bombardier. Retrieved May 29, 2022 from https://bombardier.com/en/media/news/bombardier-statement-cybersecurity-breach
[78] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, Porto, Portugal, 31–36.
[79] Siamak Mehrkanoon. 2019. Deep shared representation learning for weather elements forecasting. *Knowledge-Based Systems* 179 (2019), 120–128.
[80] Microsoft. 2022. *AZURE*. Microsoft Corporation. Retrieved Nov 2, 2022 from https://azure.microsoft.com/en-gb/
[81] Charlie Miller and Chris Valasek. 2014. A survey of remote automotive attack surfaces. *black hat USA* 2014 (2014), 94.
[82] Jelena Mirkovic and Peter Reiher. 2004. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review* 34, 2 (2004), 39–53.
[83] Ali Moin, Andy Zhou, Abbas Rahimi, Alisha Menon, Simone Benatti, George Alexandrov, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, et al. 2021. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics* 4, 1 (2021), 54–63.
[84] Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. 2017. CNN-based sensor fusion techniques for multimodal human activity recognition. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers* (Maui, Hawaii). ACM, New York, NY, USA, 158–165.
[85] Andrew Murphy. 2022. *Industrial: Robotics Outlook 2025*. Loup Funds, LLC. Retrieved February 23, 2022 from https://loupfunds.com/industrial-robotics-outlook-2025/
[86] Vedanth Narayanan and Rakesh B. Bobba. 2018. Learning Based Anomaly Detection for Industrial Arm Applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy* (Toronto, Canada) *(CPS-SPC '18)*. Association for Computing Machinery, New York, NY, USA, 13–23. https://doi.org/10.1145/3264888.3264894
[87] Vedanth Narayanan and Rakesh B. Bobba. 2018. Learning Based Anomaly Detection for Industrial Arm Applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*. ACM, Toronto Canada, 13–23. https://doi.org/10.1145/3264888.3264894
[88] Mao V. Ngo, Tie Luo, and Tony Q. S. Quek. 2021. Adaptive Anomaly Detection for Internet of Things in Hierarchical Edge Computing: A Contextual-Bandit Approach. *ACM Transactions on Internet of Things* 3, 1, Article 4 (oct 2021), 23 pages. https://doi.org/10.1145/3480172
[89] Long D Nguyen, Dongyun Lin, Zhiping Lin, and Jiuwen Cao. 2018. Deep CNNs for microscopic image classi□cation by exploiting transfer learning and feature concatenation. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (Firenze FI, Italy). IEEE, New York, NY, USA, 1–5.
[90] Zhiyou Ouyang, Xiaokui Sun, Jingang Chen, Dong Yue, and Tengfei Zhang. 2018. Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things. *IEEE Access* 6 (2018), 9623–9631.
[91] Simone Panicucci, Nikolaos Nikolakis, Tania Cerquitelli, Francesco Ventura, Stefano Proto, Enrico Macii, Sotiris Makris, David Bowden, Paul Becker, Niamh O'Mahony, et al. 2020. A cloud-to-edge approach to support predictive analytics in robotics industry. *Electronics* 9, 3 (2020), 492.
[92] Donghyun Park, Seulgi Kim, Yelin An, and Jae-Yoon Jung. 2018. LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks. *Sensors* 18, 7 (2018), 2110.
[93] Koeppe Patrick. 2020. *HUBER+SUHNER : gradually resumes production after cyberattack | MarketScreener*. Surperformance SAS. Retrieved 30 may, 2022 from https://www.marketscreener.com/quote/stock/HUBER-SUHNER-AG-278523/news/HUBER-SUHNER-nbsp-gradually-resumes-production-after-cyberattack-32074407/
[94] D Pavithra and Ranjith Balakrishnan. 2015. IoT based monitoring and control system for home automation. In *2015 global conference on communication technologies (GCCT)*. IEEE, 169–173.
[95] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, and Félix J García Clemente. 2020. Madics: A methodology for anomaly detection in industrial control systems. *Symmetry* 12, 10 (2020), 1583.

[96] SR Prathibha, Anupama Hongal, and MP Jyothi. 2017. IoT based monitoring system in smart agriculture. In *2017 international conference on recent advances in electronics and communication technology (ICRAECT)*. IEEE, 81–84.
[97] Associated Press. 2021. *Hacker tries to poison water supply in Florida city*. Telegraph Media Group. Retrieved May 3, 2021 from https://www.telegraph.co.uk/news/2021/02/09/hacker-tries-poison-water-supply-orida-city/
[98] Australian Associated Press. 2019. *Systems shut down in Victorian hospitals after suspected cyber attack*. Guardian Media Group. Retrieved May 30, 2022 from http://www.theguardian.com/australia-news/2019/oct/01/systems-shut-down-in-victorian-hospitals-after-suspected-cyber-attack
[99] Mohammad Riazi, Osmar Zaiane, Tomoharu Takeuchi, Anthony Maltais, Johannes Günther, and Micheal Lipsett. 2019. Detecting the onset of machine failure using anomaly detection methods. In *International Conference on Big Data Analytics and Knowledge Discovery* (Linz, Austria). Springer, Cham, Switzerland, 3–12.
[100] Mauro Ribeiro, Katarina Grolinger, and Miriam A.M. Capretz. 2015. MLaaS: Machine Learning as a Service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (Miami, Florida, USA). IEEE, New York, NY, USA, 896–902.
[101] Haakon Ringberg, Matthew Roughan, and Jennifer Rexford. 2008. The need for simulation in evaluating anomaly detectors. *ACM SIGCOMM Computer Communication Review* 38, 1 (2008), 55–59.
[102] Alina Roitberg, Nikhil Somani, Alexander Perzylo, Markus Rickert, and Alois Knoll. 2015. Multimodal human activity recognition for industrial manufacturing processes in robotic workcells. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, New York, NY, USA, 259–266.
[103] Beth Romanik. 2013. *Prison Computer 'Glitch' Blamed for Opening Cell Doors in Maximum-Security Wing*. Techwell Insights. Retrieved February 28, 2021 from https://www.techwell.com/techwell-insights/2013/08/computer-glitch-blamed-opening-prison-cell-doors
[104] Ellen Rushe and Brian Mac Namee. 2019. Anomaly detection in raw audio using deep autoregressive networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Brighton, UK, 3597–3601.
[105] Ali M Sadeghioon, Nicole Metje, David Chapman, and Carl Anthony. 2018. Water pipeline failure detection using distributed relative pressure and temperature measurements and anomaly detection algorithms. *Urban Water Journal* 15, 4 (2018), 287–295.
[106] Anam Sajid, Haider Abbas, and Kashif Saleem. 2016. Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges. *IEEE Access* 4 (2016), 1375–1384.
[107] Yasushi Sakurai, Yasuko Matsubara, and Christos Faloutsos. 2015. Mining and forecasting of big time-series data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, New York, NY, USA, 919–922.
[108] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2017. Recent advances in recurrent neural networks. arXiv:1801.01078 [cs.NE]
[109] Raed Abdel Sater and A. Ben Hamza. 2021. A Federated Learning Approach to Anomaly Detection in Smart Buildings. *ACM Transactions on Internet of Things* 2, 4, Article 28 (aug 2021), 23 pages. https://doi.org/10.1145/3467981
[110] Debarshi Sen, Amirali Aghazadeh, Ali Mousavi, Satish Nagarajaiah, Richard Baraniuk, and Anand Dabak. 2019. Data-driven semi-supervised and supervised learning algorithms for health monitoring of pipes. *Mechanical Systems and Signal Processing* 131 (2019), 524–537.
[111] Gauri Shah and Aashis Tiwari. 2018. Anomaly detection in iiot: A case study using machine learning. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. ACM, Goa, India, 295–300.
[112] Matti Siekkinen, Markus Hiienkari, Jukka K Nurminen, and Johanna Nieminen. 2012. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4. In *2012 IEEE wireless communications and networking conference workshops (WCNCW)*. IEEE, Paris, France, 232–237.
[113] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815 [cs.AI]
[114] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. 2018. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Transactions on Industrial Informatics* 14, 11 (2018), 4724–4734. https://doi.org/10.1109/TII.2018.2852491
[115] Daniel Sonntag, Sonja Zillner, Patrick van der Smagt, and András Lörincz. 2017. Overview of the CPS for smart factories project: Deep learning, knowledge acquisition, anomaly detection and intelligent user interfaces. In *Industrial internet of things*. Springer, Cham, Switzerland, 487–504.
[116] Thomas Stibor, Jonathan Timmis, and Claudia Eckert. 2005. A comparative study of real-valued negative selection to statistical anomaly detection techniques. In *International Conference on Arti cial Immune Systems* (Cham, Switzerland). Springer, Berlin, Germany, 262–275.

[117] Ljiljana Stojanovic, Marko Dinic, Nenad Stojanovic, and Aleksandar Stojadinovic. 2016. Big-data-driven anomaly detection in industry (4.0): An approach and a case study. In *2016 IEEE international conference on big data (big data)*. IEEE, Washington, DC, USA, 1647–1652.

[118] AbdulhahI2h9(Subasi,hI2h9(DaliahI2h9(Hon,)-25ammas,hI2h9(RahafhI2h9(DhI2h9(AIgduIdi,hI2h9(RagdudhI2h9(Aon,)-2Makawi,h