CASPER: CONTEXT-AWARE IOT ANOMALY DETECTION SYSTEM FOR INDUSTRIAL ROBOTIC ARMS

Hakan Kayan
School of Computer Science and Informatics
Cardiff University, UK
kayanh@cardiff.ac.uk

Ryan Heartfield
Exalens
London, UK
ryan.heartfield@exalens.com

Omer Rana
School of Computer Science and Informatics
Cardiff University, UK
RanaOF@cardiff.ac.uk

Pete Burnap
School of Computer Science and Informatics
Cardiff University, UK
BurnapP@cardiff.ac.uk

Charith Perera
School of Computer Science and Informatics
Cardiff University, UK
pererac@cardiff.ac.uk

December 29, 2023

ABSTRACT

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 benefits 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 network-based anomaly detection mechanisms insufficient 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 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 significance of our work lies in its contribution to developing a comprehensive solution for ICPS security, which goes beyond conventional network-based methods.

***K*eywords** neural networks, anomaly detection, industrial robotic arms, cyber-physical systems, ubiquitous computing

1 Introduction

Industrial cyber-physical systems (ICPS) Colombo et al. [2017], which is the backbone of Industry 4.0 Lasi et al. [2014], 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" Kusiak [2018] 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 Murphy [2022] 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 Tidy [2021], Press [2021] as ICPS supervise CIs. The majority of attack detection solutions rely on intrusion detection systems (IDS) Liao et al. [2013] which only perform network traffic analysis (NTA). As industrial systems have different security requirements, the characteristics of industrial IDS differ from their peers Hu et al. [2018]. 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 Urbina et al. [2016] 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 Chandola et al. [2009] 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) Mirkovic and Reiher [2004] 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 Miller and Valasek [2014] 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 Isermann [1997] 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 Isermann [2005], and signal-based Gao et al. [2015]. Model-based approaches attempt to generate an 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 Stibor et al. [2005], and data-driven Stojanovic et al. [2016]. The two significant 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 Tan et al. [2013]. The Stuxnet malware Langner [2011] attack on Iran's nuclear centrifuges is a real-world example of such an integrity attack, where attackers modified the gas centrifuge parameters. To address these drawbacks, data-driven approaches Narayanan and Bobba [2018a], Park et al. [2018] 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 Chandola et al. [2009]: supervised, semi-supervised, and unsupervised. The supervised models use labeled data for training, while the unsupervised models either do not require any training data Liu et al. [2008] or use non-labeled data for training Kravchik and Shabtai [2018]. Semi-supervised models combine these two.

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

Neural networks Gurney [2018] 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 Larochelle et al. [2009]: shallow neural networks (SNN), and deep neural networks (DNN). Bianchini and Scarselli [2014] propose a detailed comparison regarding the complexity of these two neural network types. The flexibility and scalability of neural networks make them desirable for industrial applications. In recent years, academia presented many DNN-based research papers Malhotra et al. [2015], Inoue et al. [2017], Kravchik and Shabtai [2018], Goh et al. [2017], which offer promising results, within the context of detecting physical anomalies in ICPS.

Computing infrastructures can be grouped into three based on computing location Yousefpour et al. [2019]: edge, fog, and central/cloud. In short, we define "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) Cloud [2011], Google Cloud Bisong [2019], and Microsoft Azure Microsoft [2022] are attractive options as they offer machine learning as a service (MLaaS) Ribeiro et al. [2015]. 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) Sisinni et al. [2018] is utilized. We consider IIoT as one of the requirements for advanced/smart manufacturing. While the initial IIoT solutions Wang et al. [2016], Lade et al. [2017] focus on increasing production efficiency, the use of IIoT to detect anomalies Ouyang et al. [2018], Shah and Tiwari [2018] 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 occurring in an industrial robotic arm. We utilize statistical and ML-based methods, including a neural network model employing 1D convolutional neural networks (1D-CNN) layers. Recognizing that 1D-CNNs have been applied in various domains, their use in IoT for anomaly detection based on IMU data is not extensively documented. Our study seeks to explore this and contribute to its literature. To the best of our knowledge, we are first to propose a context-aware anomaly detection system (CASPER) that detects movement-based anomalies by applying the 1D-CNN model on raw IMU data gathered from an industrial robotic arm. This data are gathered via an edge development board while anomalies are generated via the modification of arm's joint velocity. The gathered IMU data are not subject to the network vulnerabilities. This approach addresses the concern that built-in data being susceptible to spoofing. Our choice of 1D-CNN is driven by its computational advantages, suitable for the constraints of IoT environments, and while our research does not focus on identifying a superior detection method, we explore the capabilities of 1D-CNN within this specific context. Specifically, where 1D-CNN is capable of delivering comparable detection fidelity and performance to that of more sophisticated state of the art machine approaches, whilst in combination offering superior detection speed (low-latency inference) which is key for efficient response and recovery. CASPER also ensures the integrity of data generated via a cyber-physical edge resource, as data is 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 offering 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 files in total: (I) A file that consists of accelerometer, gyroscope, and magnetometer data of an arm that accomplishes a repetitive task, (II) two files (one per industrial arm) that consist of built-in arm parameters such as joint current, and velocity values, (III) one pcap file which contains all the network traffic 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, demonstrating how the proposed model performs when the data are correlated.

[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	Healt Facilities	Australia	Healthcare	Cyber	●	○	Health operations were delayed.
2020	HUBER+SUHNER	Switzerland	Manufacturing	Cyber	●	◐	All network was shut down.
2021	Colonial Pipeline	USA	Utility	Cyber-Physical	●	●	Pipeline was shut down.
2021	Water Plant	USA	Utility	Cyber-Physical	●	●	Water is poisoned.
2021	Caffitaly	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 affected, ◐ : The domain is indirectly affected, ○ : The domain is not affected.

2 Background

Our work focuses on the application of cyber-physical anomaly detection systems to robotic arms within manufacturing environments, where cyberattacks could cause significant disruptions. In this section, we detail a selection of real-world cyber incidents, emphasizing the physical impact they had on industrial systems as explored in our prior research Kayan et al. [2022]. These incidents are chosen for their clear demonstration of how cyber threats can translate into tangible consequences in a manufacturing setting. Inspired by these examples, our experimental design involves altering the joint velocity of a robotic arm thus simulating the disruptive effects of a cyber-physical attack. This decision allows us to create test scenarios that are not only representative of real-world attacks but also applicable to the physical domains our anomaly detection system aims to safeguard.

In 2013, the maximum-security prison Turner Guilford Knight Correctional Center in Florida, USA had been subjected to two cyber incidents in one month Romanik [2013]. The prison control system was recently upgraded for a cost of $1.4 by a firm 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 first event was done to test the response of the guards, and the second was carried out for a more specific 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 Press [2021]. 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 staff (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 Tidy [2021]. 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.

In March 2021, Canadian IoT as a service provider Sierra Wireless was subjected to a ransomware attack Bleeping-Computer [2021]. The IT systems of the company were locked down. The company announced that there was no damage done to any production units and the confidential customer data was not affected 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 December 14, 2020, HUBER+SUHNER, a fiber optic cable manufacturing company located in Switzerland, was subjected to a cyberattack Patrick [2020]. 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 confidential data was stolen.

In February 2021, the Italian coffee capsule/machine manufacturer Caffitaly System was subjected to a cyberattack Comunicaffè [2021]. 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 Wire [2021]. 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 Press [2019]. 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 Langner [2011] and Florida water poisoning Press [2021] 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 Fujimaki et al. [2005], Tsang and Kwong [2005], Chandola et al. [2009], Kayan et al. [2022]. 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 Bayram et al. [2021], Duman et al. [2019], IMU Narayanan and Bobba [2018b], joint current Panicucci et al. [2020], Chen et al. [2020], electromagnetic side-channel signal Khan et al. [2019], tension Riazi et al. [2019], vibration Park et al. [2018], or visual Yetis and Karakose [2018] data. In addition to these, we can utilize temperature data Tanuska et al. [2021] 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 Ghazal et al. [2020], 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 Mathur and Tippenhauer [2016] 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 Wu et al. [2017], Kravchik and Shabtai [2018], Perales Gómez et al. [2020] 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 Park et al. [2018], Karim et al. [2019]. This proposes no issue if the detection is done offline (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 Manimurugan [2021]. However, the occurrence of delay causes researchers to

pursue alternative approaches Sater and Hamza [2021], Ngo et al. [2021]. This delay can also be eliminated by applying anomaly detection on edge devices. The available methods are pretty limited but expanding David et al. [2020] thanks to the rapid development of ubiquitous technologies. IoT devices are also used for real-time monitoring Pavithra and Balakrishnan [2015], Prathibha et al. [2017] 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 Münzner et al. [2017], Roitberg et al. [2015]. For example, the Apple Watch Apple [2022] 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) Guo et al. [2016], Nguyen et al. [2018], Zhang et al. [2019] or utilized individually (ensemble classifiers) Wei et al. [2017], Subasi et al. [2018], Ani et al. [2017], Haladjian et al. [2020]. Traditional machine learning (ML) methods use a single modality for each stage of the ML application Rushe and Mac Namee [2019]. Multimodal fusion approaches employ all modalities at each stage Bernal et al. [2017], Debie et al. [2019]. Cross-modality learning approaches Hong et al. [2020], Zhang et al. [2021] utilize all modalities during feature learning while training and testing are performed with the same single modality, which differs from shared representation learning Yi et al. [2015], Mehrkanoon [2019], where different modalities are used for testing and training.

3.4 Sensor Data Analysis with ML-based Approaches

Data-driven ML methods are grouped into three Géron [2019] 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) Bedri et al. [2017]. 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 Sen et al. [2019] 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 Kayan et al. [2021] 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 Silver et al. [2017]. The learning might be online or offline. *Online* algorithms learn on the fly while *batch/offline* learning makes use of pre-gathered data to train the model. Adaptive ML models Moin et al. [2021] require online learning algorithms due to novel streaming data. Offline learning is more common in classification tasks such as natural language processing (NLP) Lopez and Kalita [2017] where the capacity of the model depends on the size/content of the training data. ML models can also be classified 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 classification Ciregan et al. [2012] while model-based methods are seen in predictive analytics/forecasting Sakurai et al. [2015].

CNNs offer 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 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 Salehinejad et al. [2017]. 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) Li et al. [2021], 1D-CNNs are useful for time series data as such data are expected to have strong temporal correlations LeCun et al. [1995]. 1D-CNNs are less computationally intensive and require significantly fewer operations, rendering them highly effective for real-time sensing applications. The survey of Kiranyaz et al. [2021] suggests that 1D-CNNs can perform motor fault detection much faster than other neural network-based approaches. Additionally, Shahid et al. [2022] demonstrate that 1D-CNNs achieve performance comparable to 2D-CNNs in classifying crank angle degree signals for engine fault detection. In predicting the remaining useful life of turbofan engines, Athanasakis et al. [2022] shows that 1D-CNNs can equal the performance of other models while having smaller sizes and lower inference latency. Freire et al. [2022] further provides a detailed computational complexity analysis, highlighting that 1D-CNNs scale more efficiently than their counterparts. Their efficiency extends to the point where they can be

Table 2: 1D-CNN Efficiency Across Use Cases

Reference	Use Case	Results
Athanasakis et al. [2022]	Remaining Useful Life Prediction of Turbofan Engines	1D-CNN achieves an optimal balance of efficiency compared to LSTM, XGBoost, and Random Forest.
Freire et al. [2022]	Digital Signal Processing	LSTM layers have the highest complexity among Dense and 1D-CNN layers.
Kiranyaz et al. [2021]	Motor Fault Detection	1D-CNN can provide detection up to 45 times faster than other neural network-based algorithms.
Shahid et al. [2022]	Motor Fault Detection	1D-CNNs demonstrate performance nearly similar to 2D-CNNs but with less processing required.

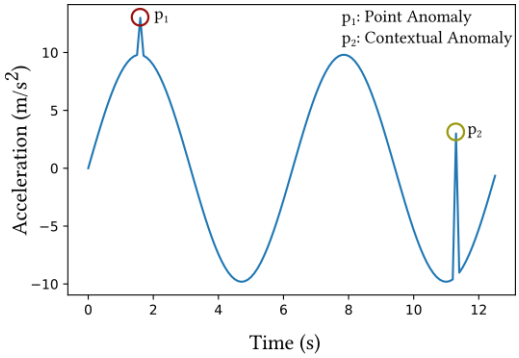


Figure 1: Demonstrates an example acceleration data of one industrial robotic arm joint. While the point anomaly p_1 does not appear across the data (or appears very less in numbers), the contextual p_2 does. While p_1 can be detected via simple thresholding, more sophisticated methods are required to detect p_2. Collective anomaly is the event where point/contextual anomaly occurs simultaneously across all joints.

implemented on ultra-low-power devices (<1 mW) [3], an aspect we plan to explore in future work. Table 2 summarizes these studies. The various recent applications of 1D-CNNs include ball bearing fault detection Ince et al. [2016], water treatment system anomaly detection Kravchik and Shabtai [2018], HAR Cho and Yoon [2018], seizure detection Jana et al. [2020], and music genre classification Allamy and Koerich [2021].

4 Anomalies

In the field of data science, anomalies 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 define a behavior. This section provides an overview of different types of anomalies, decision-making methods, and techniques for generating anomalous data.

4.1 Anomaly Types

Anomalies are classified into three categories Chandola et al. [2009]: (I) point anomalies, (II) contextual anomalies, and (III) collective anomalies. *Point anomalies* differ from the rest of the data. Being the most common ones, if the anomaly type is not mentioned, it usually refers to point anomalies Lu et al. [2017], Sadeghioon et al. [2018], Yan and Yu [2019]. *Contextual anomalies* are harder to detect as such detection requires context (e.g., time, location) analysis where defining one might be challenging. The application that generates time series data tends to contain contextual anomalies where the context is the time Carmona et al. [2021], Liu et al. [2017]. *Collective anomalies* is a group of data that differs 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 identified via network anomaly detection methods Ahmed and Mahmood [2014, 2015]. Figure 1 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 defined 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-defining methods such as SVMs Narayanan and Bobba [2018a] tend to utilize binary definitions, decision tree-based approaches such as Isolation forest Liu et al. [2008] utilizes anomaly scores. On the other hand, regression

[3] https://github.com/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/micro_mutable_op_resolver.h

Table 3: Anomaly Creation Methods

Reference	Testbed	Attack	Anomaly Creation Method
Narayanan and Bobba [2018a]	Industrial Robotic Arm	-	Set industrial arm to follow a different trajectory.
Chen et al. [2020]	Industrial Robotic Arm	-	Manually injecting faults.
Khan et al. [2019]	Robotic Arm Syringe Pump	✓	Implementing control-flow hijack and firmware modification attacks.
Riazi et al. [2019]	Belt-driven Robotic Arm	-	Loosening and tightening the belt.
Park et al. [2018]	Robot Manipulator	-	Adjusting the amount of air injected into vacuum ejector.
Angle et al. [2019]	High Voltage Motor Development Kit	-	Modifying the firmware to allow to damage the kit.
Vuong et al. [2014]	Robotic Vehicle	✓	Conducting DoS attack.
Wu et al. [2019]	3D Printer	-	Injecting faulty files to 3D printer to print a damaged product.
Gao et al. [2018]	3D Printer	-	Modifying the firmware to change printer features such as printing velocity.
Li et al. [2019c]	Rotor Kit	-	Adding weights to a mass load.
Bezemskij et al. [2016]	Robotic Vehicle	✓	Conducting replay attack, creating rogue node, manipulating compass, and breaking wheel.
Sonntag et al. [2017]	Industrial Robotic Arm	-	Hitting to an industrial arm.
Sisinni et al. [2018]	Robotic Vehicle	✓	Conducting DoS, command injection, and malware attack.
CASPER	Industrial Robotic Arm	-	Manually manipulating the joint velocity of the arm.

methods (e.g., gradient boosting, logistic regression) estimate a value. Then statistical methods are applied to the residuals which are the absolute difference between the predicted and actual values.

4.3 The Use Case Scenario

While the use of public datasets Mathur and Tippenhauer [2016], Li et al. [2019b,a], Deng and Hooi [2021], Goh et al. [2017] enables benchmarking similar works, having no control over anomaly creation beclouds the recreation of desired challenging scenarios. This also applies to simulation-only studies Filonov et al. [2016], Ringberg et al. [2008]. 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. Literature review reveals a preference for non-destructive methods in generating anomalies within cyber-physical systems, especially when dealing with high-value assets like industrial robotic arms Sonntag et al. [2017], Chen et al. [2020], Narayanan and Bobba [2018b]. Direct physical attacks tend to be reserved for lower-cost equipment to avoid the high costs and risk of irreparable damage to more expensive machinery Khan et al. [2019], Vuong et al. [2014], Bezemskij et al. [2016], Sisinni et al. [2018]. This study follows recognized methods. Due to the high precision required by industrial robotic arms, small alterations in velocity or trajectory can lead to significant operational disruptions. While past research Narayanan and Bobba [2018b] has examined trajectory-based anomalies, our investigation concentrates on velocity adjustments introducing anomalies within operational limits. Table 3 demonstrates the anomaly creation processes of related work.

In this work, we implement a scenario inspired by the Florida water poisoning incident Press [2021], where an adversary gains control of an industrial system. The attack unfolds in two main stages: (I) Initially, the adversary sends a phishing email to the enterprise network and gains initial access by acquiring the necessary credentials. Then, the adversary bypasses the firewall and begins spoofing the joint velocity data, thereby disrupting the manufacturing process. Due to the joint velocity data being spoofed, the network-based intrusion detection system fails to recognize this event, as the data appears normal. This also holds true for built-in Human-Machine Interfaces (HMIs), as the staff monitoring the data would perceive everything as functioning normally. There are only two methods to detect such an event when the integrity of the network data is compromised: either the onsite staff notices the unexpected changes, or a third-party edge anomaly detection mechanism that supervises the affected industrial robotic arm, independent of the network, can be employed as proposed in this work. Figure 2 illustrates an example attack scenario.

5 CASPER - System Overview

The CASPER consists of edge, fog, and central components that offer 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-efficient devices. The IoT environments are dynamic, heterogeneous, and resource-constrained. Thus, we need the aforementioned characteristics to have a sustainable model. (II) They should 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 offering

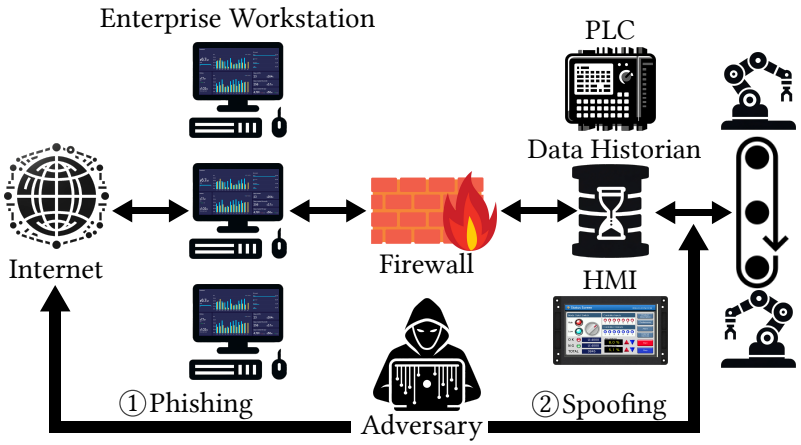


Figure 2: Example attack scenario implemented in this work.

Table 4: 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 flash	256 KB SRAM, 1MB flash	64 KB SRAM, 512 KB flash
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 5: 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

accessible deployment. We compare three edge development boards based on the aforementioned requirements: (I) Arduino Nano 33 BLE Sense Team [2021b], (II) Adafruit Feather nRF52840 Sense Adafruit [2021], (III) Nicla Sense ME Team [2021a][4]. Table 4 compares tech specifications 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 define an orientation.
- The edge development board must provide BLE Siekkinen et al. [2012] connectivity. We observed in our previous work Kayan et al. [2021] that BLE offers low power usage and flexibility 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 Ergen [2004] 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 Sajid et al. [2016]. In this case, the fog device is also expected to have enough capacity to perform preconfigured tasks

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

(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 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 Babu et al. [2019], Gonzalez-Huitron et al. [2021] offer promising benchmarking results Luo et al. [2018]. RPi4 runs on DietPi OS Knight [2021], that minimizes resource usage when running Node-RED and Grafana. A more cost-efficient 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 efficiency due to undesired events such as concept drift. However, in this work, we do offline learning as our primary target is to investigate the efficiency of 1D-CNN for anomaly detection while offering real-time 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 5 demonstrates the key specifications 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 different edge development boards in terms of the generated IMU data and introduce the CASPER dataset. We assess the effectiveness 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 6 and Table 7 present the testbed components while explaining their key features and tasks. Figure 3 visualizes each component, demonstrates how each component communicates, defines 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.
- 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.

Table 6: 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 7: Software Components

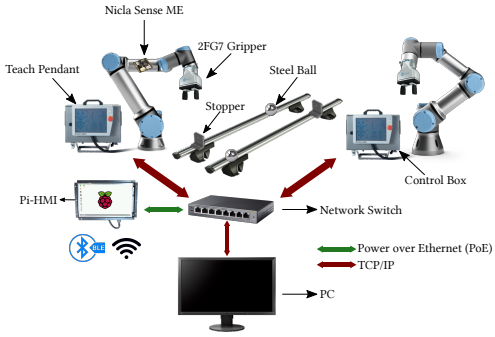
Software Name	Purpose	Version
Grafana	Provides interactive visualization of IMU data.	9.0.9
InfluxDB	Stores the IMU data.	1.8
DietPi OS	Manages Pi-HMI. Power efficient OS for Pi.	8.0
Ubuntu	Manages the central PC.	20.04
Python	Enables programming of the simulation.	3.8
Universal Robot Scripts (urp)	Communicate 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 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.

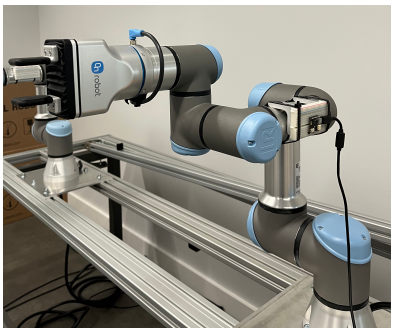
6.2 Sensor Fusion & Edge Development Board Comparison

Micro-electro-mechanical systems (MEMS) sensors that generate IMU[5] data are: (I) accelerometer and (II) gyroscope, and (III) magnetometer. The accelerometer measures the linear acceleration which defines the velocity change in units of either gravitational force (g) or meters per second squared (ms^{-2}). The gyroscope measures the angular velocity which defines the rotational change in motion in units of degrees per second (dps). The magnetometer measures local magnetic field 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 define an accurate 3D orientation Islam et al. [2017]. 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 suffer from gimbal lock which causes the loss of one degree of freedom. Thus, quaternion representations are preferred. Mahony Mahony et al. [2008] and Madgwick Madgwick et al. [2011] are two popular AHRS filters that define orientation via quaternions. Madgwick filter generates less root mean squared error (RMSE) while being computationally expensive in a negligible matter Ludwig et al. [2018] in Adafruit and Arduino boards where we utilize open-source libraries[6][7]. We use proprietary libraries[8] developed by Bosch for the Nicla Sense ME where quaternions are generated via the Mahony algorithm. We compare the quality of the IMU data

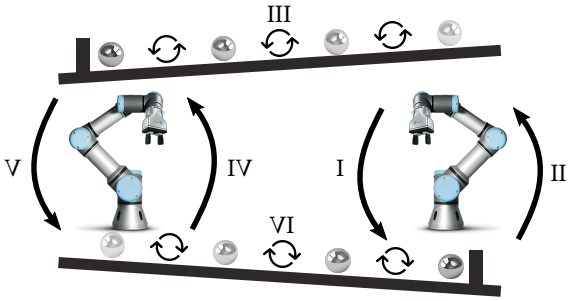
[5] Sometimes IMU is defined as magnetic and inertial measurement unit (MIMU) due to the presence of magnetometer.
[6] github.com/adafruit/Adafruit_AHRS
[7] github.com/arduino-libraries/Arduino_LSM9DS1
[8] github.com/arduino/nicla-sense-me-fw



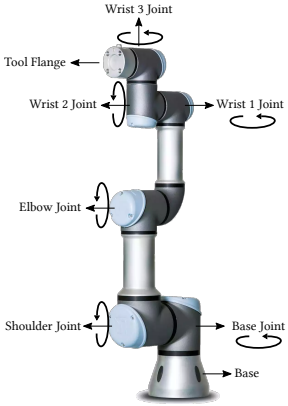
(a) Demonstrates the testbed. We transfer edge data to Raspberry Pi via BLE. The local PC controls two industrial robotic arms over TCP/IP.



(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 section 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 attach 2FG7 parallel gripper.

Figure 3: Testbed and use case scenario.

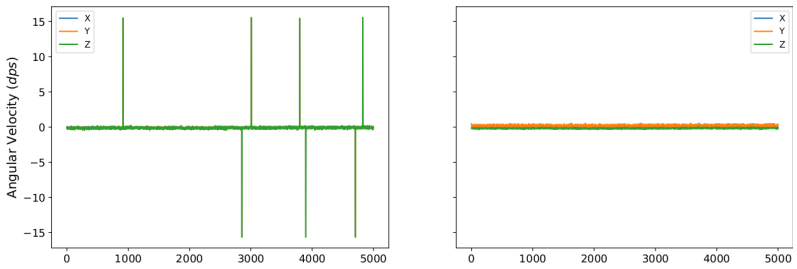
Table 8: Edge Development Board Testing

Edge Development Board	Arduino Nano 33 BLE Sense	Adafruit Feather nRF52840 Sense	Nicla Sense ME
Charge 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. ([$\pm$1300 (x, y), $\pm$2500(z)] μT & 0.02 μT)
Cost	35.10 £	31.92 £	59.82 £

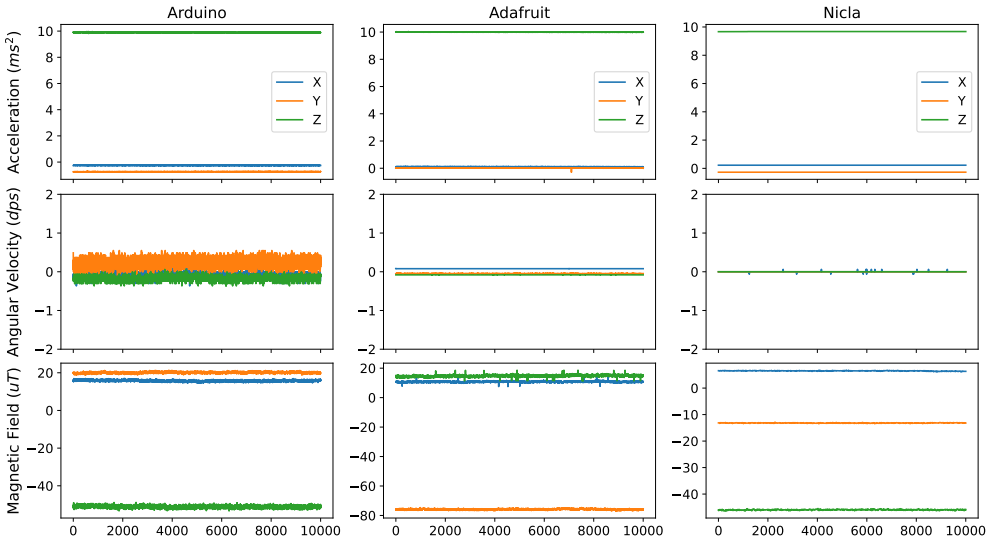
* By "Raw", we mean accelerometer, gyroscope, and magnetometer data. T: Tesla, dps: degrees per second, g: G-force. Acc: Accelerometer, Gyro: Gyroscope, Mag: Magnetometer. Ranges are the default ones.

while also observing the quaternion generation to visually observe the stability of sensors (see Figure 4). We observe the following:

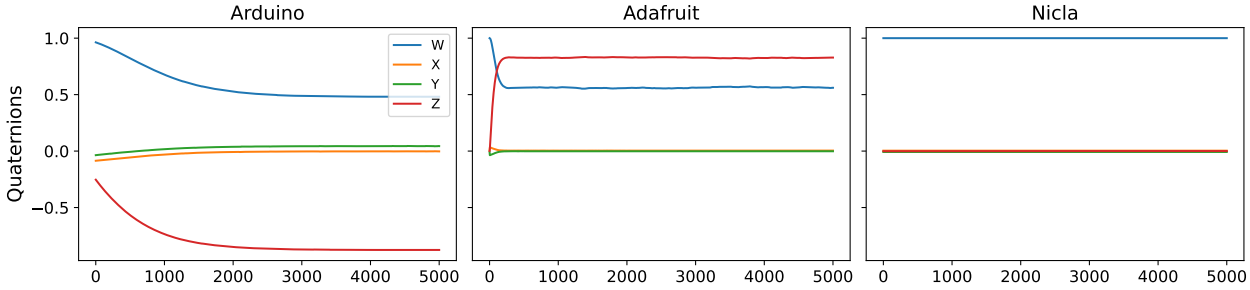
- *Adafruit consumes less power overall.* Out of three edge development boards, the power consumption of Adafruit is significantly lower than Arduino while being closer to Nicla. If we supply these boards with 9 Volts 250 mAh battery, we would expect the Adafruit to run around 20 hours, Nicla to run 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 reflects the actual change. However, analysis of gyroscope data revealed the existence of random spikes, which may introduce potential outliers to the data.



(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 left 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 drift.

Figure 4: Edge data generation comparison.

Table 10: 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.

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 different 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 first phase where the joint velocities are higher than the default, and the second phase where the opposite applies. We explore a range of velocities, from a 100% increase, which is the maximum permitted due to safety constraints, to a 5% decrease which represents the smallest change that consistently results in observable data alterations. These variations are pre-defined and timed hence allowing us to accurately label the dataset with the exact timestamps when the arm's movements transition from normal to anomalous behavior. The Table 9 demonstrates the anomalies with respect to time.

Table 9: 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 files generated from a pick-and-place operation lasting around 24 hours: The first Comma Separated Values (CSV) file 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 files (one file per arm) contain built-in arm parameters (e.g., joint positions, velocities, and currents). We gather both data at 20Hz which corresponds 50 ms difference between two consecutive data points. The final file is a PCAP containing the network traffic between the local controller PC and the arms. Table 10 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 effectiveness of an air-gapped IoT anomaly detection system. We share the built-in and network data for researchers who are working in related fields.

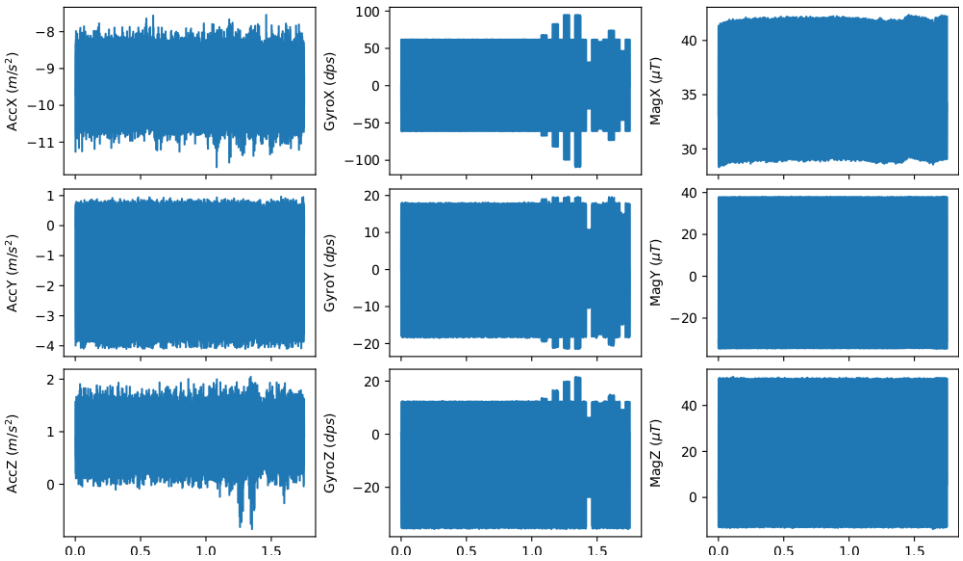


Figure 5: Demonstrates IMU data generated via an edge development board attached to an industrial robotic arm. We can easily see that the anomalies reflect on the X-axis of gyroscope data.

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 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.

6.4.2 Correlation Analysis

We apply autocorrelation to find the highest time-dependent Pearson correlation coefficient (r) denoted as ρ where E is the expected value, μ is the mean and σ is the standard deviation (see Equation 1) to find the periodicity. Our autocorrelation analysis reveals the period with the highest Pearson correlation coefficient which guides us to set a 755-point window size for the 1D-CNN and other detection methods enhancing anomaly sensitivity. Non-anomalous runs show a different periodicity that becomes evident when comparing with the anomalous ones as seen in Table 11. 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 12): (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.

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

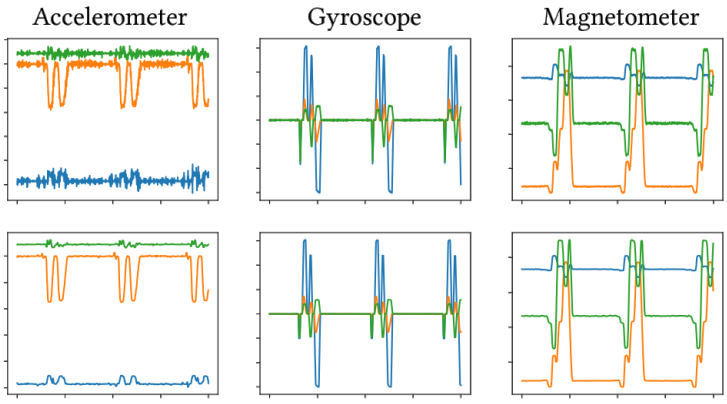


Figure 6: Demonstrates the effect of noise removal on all features. The bottom three figures are the noise-removed data.

Table 11: Autocorrelation Analysis for Non-Anomalous and Anomalous Runs

Run Type	AccX	AccY	AccZ	GyroX	GyroY	GyroZ	MagX	MagY	MagZ
Non-Anomalous (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
Anomalous (r, w)	0.994, 757	0.998, 799	0.971, 769	0.997, 770	0.996, 791	0.993, 775	0.997, 759	0.999, 799	0.998, 775

Note: r represents the Pearson correlation coefficient and w denotes the window length.

Table 12: 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]

6.4.3 Baseline

We employ a statistical baseline as a benchmark to validate the effectiveness of data-driven approaches. This baseline is crafted by segmenting the data into input windows derived exclusively from non-anomalous segments. The length of these windows corresponds to the period identified through our correlation analysis, which reflects the strong periodicity due to the robotic arm's movement patterns. We focus on the temporal correlations by adjusting the window sizes via reducing the lag observed between the input windows. This lag, initially varying from -3 to 3 data points, tends to increase over time, potentially leading to a quarter-window delay. To address this and strengthen our baseline, we select the initial window, comprising 755 data points, as our reference. Both mean and median windows are then computed from this reference. Subsequently, we assess the baseline performance by calculating the overall Root Mean Square Error (RMSE), as detailed in Equation 2:

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

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. This optimized approach ensures that the baseline is not only simple but

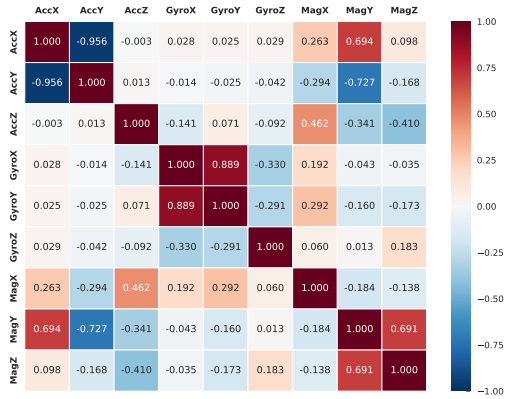
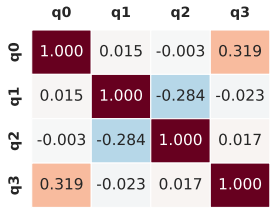
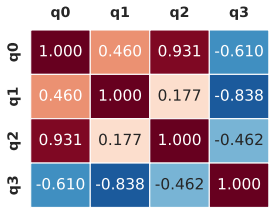


Figure 7: The correlation of input features. We see that several features are highly correlated (e.g., X and Y-axes of accelerometer. This is expected due to the nature of IMU data.



(a) Correlation heatmap of Madgwick Quaternions.



(b) Correlation heatmap of Mahony Quaternions.

Figure 8: A comparison of correlation heatmaps of two common quaternion algorithms.

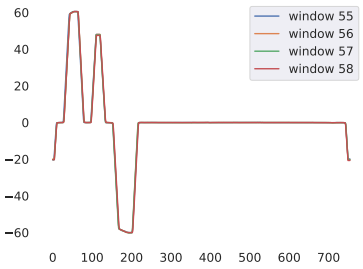
also robust capturing the periodic nature of our dataset. We measure the performance of anomaly detection methods via a confusion matrix consisting of four main parameters: (I) True positives (TP)-when an anomaly is detected as an anomaly, false positives (FP)-when normal is detected as an anomaly, true 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 difference, and confusion matrix of baseline.

6.4.4 Partial Least Squares regression

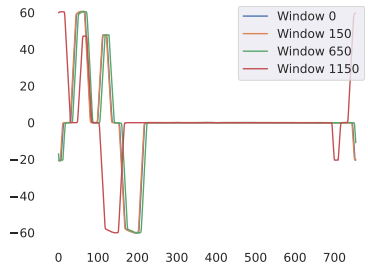
Due to the high correlation of input features, we investigate the feasibility of using Partial Least Squares regression as an anomaly detection method. PLS reduces the number of predictors to 7 capturing around 99% of the variation of the data where the correlations between the predictors are near-zero. 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

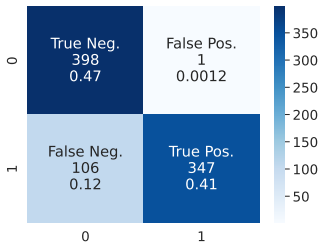
We design a 1D-CNN-based ML algorithm to detect anomalies. We expand the receptive field by stacking two 1d-CNN layers to extract deeper local/temporal features. These layers are followed by a max pooling layer that makes the model more robust to overfitting. Finally, we output our features via the fully 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 because it is well-known for its computational efficiency and its ability to introduce non-linearity, which is essential for capturing the complex patterns in the IMU data without overfitting Szandała [2021]. We employ grid search for hyperparameter tuning, optimizing loss on non-anomalous data to ensure our model generalizes. Hyperparameter limits are set to prevent overfitting, halting adjustments when they compromise model performance or loss metrics. 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.



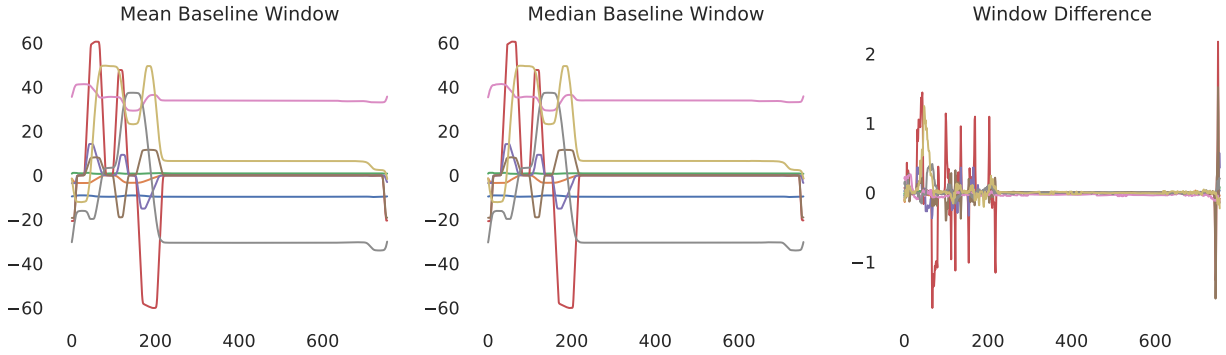
(a) Consecutive windows.



(b) Random windows.



(c) Confusion matrix.



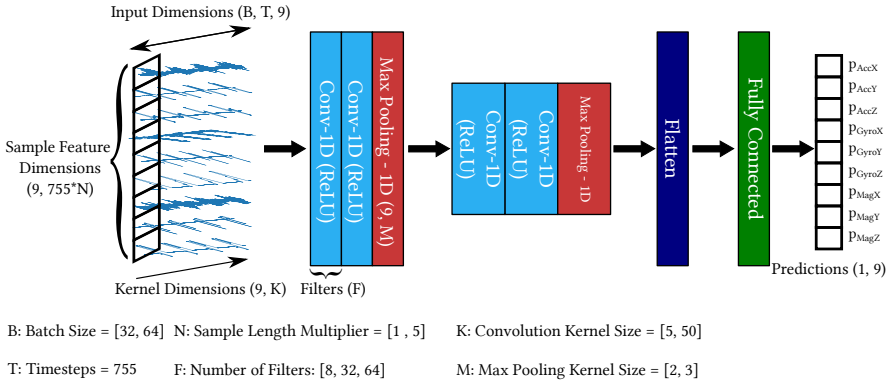
(d) Comparison of baseline input windows.

Figure 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 better than the medium baseline with metrics of 84.6% accuracy and 83.4% F1 score.

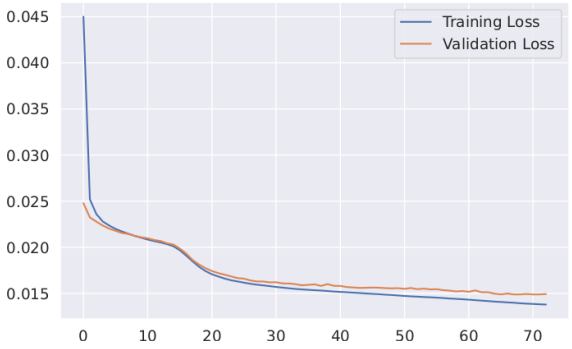
Algorithm 1 Sliding Window-based Anomaly Detection Algorithm

Require: Test data $X \in \mathbb{R}^{n\times 9}$, $\mu_{training} \in \mathbb{R}^{1\times 9}$, $\sigma_{training} \in \mathbb{R}^{1\times 9}$, threshold list $T \in \mathbb{R}^k$
Ensure: List $P \in 0, 1^l$, where $l = m - 755 + 1$, where $m = n - 755$, where $n = |X|$,

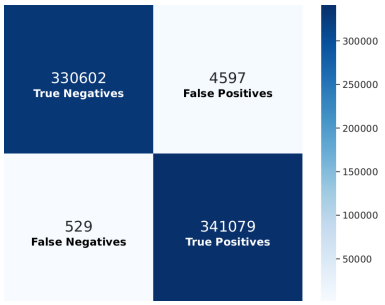
$\hat{X} = \frac{X - \mu_{training}}{\sigma_{training}}$ ▷ Normalize test data via training parameters
$W \in \mathbb{R}^{755\times 9}$ ▷ Initialize a sliding window with size 755
$R = [],\ S = [],\ P = []$ ▷ Initialize empty lists for RMSE values, RMSE rolling sums and final labels
for $i = 1$ to $n - 755$ **do**
    $W = \hat{X} i : i + 754, :$ ▷ Select the i^{th} window of test data
    $\hat{y} = f_{\text{1D-CNN}}(W) \in \mathbb{R}^{1\times 9}$ ▷ 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
    $r_i = \sqrt{\frac{1}{9}\sum_{j=1}^{9}(y_{i,j_{\text{target}}} - y_{i,j})^2}$ ▷ Calculate RMSE per time step
    $R \leftarrow [r_i]$ ▷ Append to RMSE list
end for
$S_i = \sum_{j=i-W+1}^{i} R_j \quad$ 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 $S_i > T_j$ **then**
            $P \leftarrow P + [1]$
        else
            $P \leftarrow P + [0]$
        end if
    end for
end for



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



(b) The loss graph of the model.



(c) The confusion matrix for 1D-CNN.

Figure 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.

6.4.6 Long Short-Term Memory Recurrent Neural Network

For time series data, Long Short-Term Memory (LSTM) networks are often the go-to choice due to their ability to effectively 'remember' past inputs over extended time intervals. In our approach, we utilize an LSTM model specifically tailored to our dataset's characteristics. The model consists of two LSTM layers, each followed by batch normalization to improve training stability. The first LSTM layer returns sequences to ensure continuity of state across the time steps, while the second LSTM layer does not, serving as a form of feature extraction. To mitigate overfitting, a dropout layer is included after the first batch normalization. This model also includes additional dense layers to further process the learned features. The final dense layer reshapes the output to match the number of features in our dataset, ensuring that the model's output is appropriately structured. Detailed insights into the performance of the model, loss graph, and specific hyperparameters are 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 our model. 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 One-Class SVM

The One-Class SVM is employed for its unique method of defining the normal operational state without requiring labeled anomaly data. This feature proves beneficial in situations where anomalies are not frequent (e.g., industrial cyber-physical systems). The One-Class SVM creates a boundary that seeks to contain all these data points by constructing a model based on the "non-anomalous" operational data. Anomalies are then identified as data points that fall outside this decision boundaries. In our work, the One-Class SVM fails to perform optimally. It struggles with contextual anomalies, which are anomalies defined by their occurrence within specific contexts in a temporal sequence. These anomalies require an analysis of temporal relationships between data points to be accurately identified, a capability the One-Class SVM does not have.

6.4.9 Autoencoders

Autoencoders are designed to compress data into a reduced dimensionality and subsequently reconstruct it back to its original form. In anomaly detection applications, the reconstruction error is used to determine whether an input is anomalous. Their versatility comes from the types of layers used, such as LSTM, 1D-CNN, 2D-CNN, or dense layers, each offering different characteristics. Autoencoders are computationally more expensive than previously mentioned neural network regression methods due to their dual components and the necessity to reconstruct the entire input. In this work, we implement three types of autoencoders: Dense-AE, 1D-CNN-AE, and LSTM-AE. The Dense-AE effectively detects anomalies when there is an increased joint velocity but struggles with decreased velocity runs, as the reconstructed samples mimic anomalous behavior, resulting in low RMSEs. On the other hand, both 1D-CNN-AE and LSTM-AE perform well for both types of anomalous runs. The best performing network architectures and hyperparameters are identified via grid search.

6.4.10 Comparison of anomaly detection methods.

Table 13 showcases the performance of various anomaly detection systems implemented on IMU data. To provide a foundational benchmark, we include a null model that consistently predicts the majority class in the dataset (for example, "All Anomaly"), alongside the statistical baseline. This approach validates the efficacy of more complex, data-driven methods. The statistical baseline, adjusted manually to eliminate lags, demonstrates strong performance, achieving approximately 96% accuracy. PLS faces challenges in anomaly detection, struggling to differentiate between normal and anomalous data, which results in small losses and renders thresholding methods ineffective. Similarly, the One-Class SVM performs poorly, primarily due to its inability to account for the temporal nature of the data and lack of contextual understanding. In contrast, 1D-CNN-AE and 1D-CNN show robust performance with low inference latency. 1D-CNN-AE achieves the highest F1 score, followed by the 1D-CNN, XGBoost, and the statistical baseline. Despite having the lowest inference latency, XGBoost maintains high accuracy and precision, making it ideal for time-sensitive applications. However, it generates a higher amount of false positives than 1D-CNN, which affects its desirability in real-world settings compared to the savings in detection latency for the time that would be taken analyzing false alarms.

Overall, the experimental results support our choice of selecting 1D-CNN as a viable low-latency and accurate model architecture for cyber-physical anomaly detection. Our testing shows that it delivers superior or comparable detection performance to more complex algorithms and model architectures, with superior detection speed. This makes it better suited to scenarios requiring accelerated response and recovery. All models utilized in this study are available in our GitHub repository for further exploration and use.

6.5 IoT Supervision System

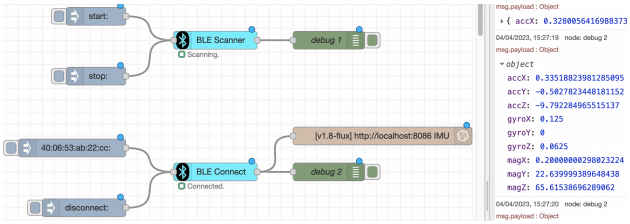
In this work, we present a method for real-time monitoring in an industrial environment where industrial robotic arms present. Our system employs an IoT device to collect IMU data from the arm. This data are then transmitted to a local fog device (PiHMI) for instant monitoring. Data transfer from the edge is conducted over BLE, with the Nicla Sense ME leveraging an nRF52832 microcontroller for BLE 4.2 connectivity. This ensures encrypted data transmission. We use Node-RED, an open-source flow-based programming tool, to build our real-time monitoring system. We utilize a Node-RED package[9] which we developed to enable receiving data from the edge device at the fog layer. This setup enables continuous IMU data monitoring, vital for safety and efficiency in industrial processes as evidenced by past incidents (refer to Section 2). InfluxDB is used as a data historian akin to those in industrial settings. Grafana retrieves IMU data from InfluxDB and displays it in real-time on the screen of the PiHMI. Fog device runs on DietPi OS which is an lightweight operation system. Figure 11a demonstrates the Node-RED setup, Figure 11b displays the Grafana dashboard, and Figure 11c presents the utilized hardware and software tools.

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

Table 13: Comparison of Anomaly Detection Approaches

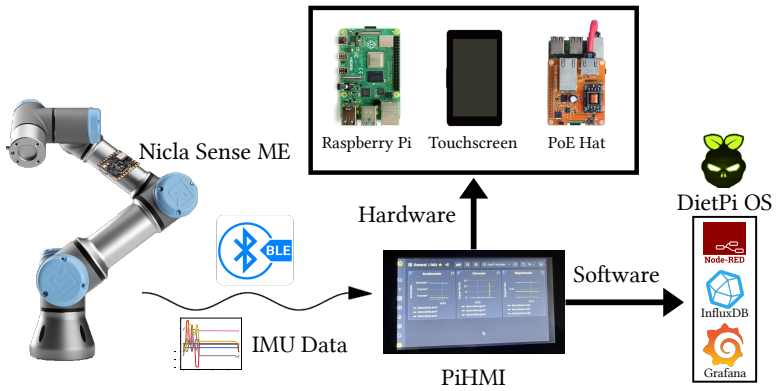
Approach	Accuracy	Recall (TPR*)	FPR*	Precision	F1 Score	Inference Latency (μs)
Null Model	0.505	1.0	1.0	0.505	0.671	NA*
Statistical Baseline	0.9576	0.9339	0.0180	0.9814	0.9571	124.18
One-Class SVM	0.502	0.506	0.6352	0.637	0.564	896.07
PLS	0.5047	1.0	1.0	0.5047	0.6708	41.73
1D-CNN	0.9924	0.9984	0.0137	0.9867	0.9925	36.97
XGBoost	0.9920	0.9995	0.0154	0.9850	0.9922	5.27
LSTM	0.9226	0.8922	0.0463	0.9514	0.9209	51.80
Dense-AE	0.7464	0.5783	0.0818	0.8782	0.6974	103.96
1D-CNN-AE	0.9954	0.9982	0.0073	0.9928	0.9955	214.66
LSTM-AE	0.9118	0.8957	0.0717	0.9272	0.9112	1031.3

NA*: Not applicable.
FPR*: False Positive Rate.
TPR*: True Positive Rate



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

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



(c) The overview of the real-time monitoring system.

Figure 11: The open-source real-time IoT-based monitoring system.

We prefer open-source and lightweight tools that offers high degree of customization and system longevity. The edge device runs a *cpp* file, and the fog device is configured using Node-RED and Grafana interfaces while the database is set up using the command line interface (CLI) of InfluxDB. Table 14 reveals the RAM usage and power consumption of the Nicla Sense ME. Generating IMU data accounts for 44.3% of RAM usage, while the effect of use of BLE on RAM usage is negligible. A moderate increase in resource consumption, especially in BLE and data visualization phases, indicates the system can function effectively without straining the hardware. With a 9-volt 500mAh battery, Nicla operates for 32.5 hours generating IMU data, 48.5 hours when idle, and 28.4 hours transmitting data over BLE. Table 15 displays PiHMI's resource usage. The minimal active CPU and RAM usage of PiHMI highlight the lightweight nature of the employed tools. These insights confirm the system's ability to meet real-time data processing and visualization demands, a critical component for immediate monitoring and decision-making in industrial settings.

Table 14: Nicla Resource Usage

	RAM Usage (bytes)	Charge Consumption (Ah)
Idle	7720 (12%)	0.0103
IMU*	36224 (56.3%)	0.0154
BLE**	36360 (56.6%)	0.0176

IMU*: *imu.cpp* file only generates IMU data. *ble.cpp* generates IMU data and sends over BLE to PiHMI.

Table 15: PiHMI Resource Usage

	RAM Usage (Megabytes)	CPU Usage
Idle	179 (5%)	0.8%
BLE*	235 (6.2%)	5%

BLE*: When PiHMI is actively displaying IMU data on 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[10], 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 detection implemented via forecasting. Therefore, it is crucial to explore novel methodologies that enable more efficient 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

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

accuracy requirements of industrial applications, any unexpected physical deviation would likely have been detected by the relevant staff, leading to a halt in training/production.

Continuous anomalous runs longer than the input window. The proposed baseline approach relies on a sample window generated through averaging non-anomalous windows. A stronger baseline approach that accounts for these correlations would involve averaging averaging the root-mean-square errors (RMSEs) of consecutive windows. However, while this method can effectively detect the beginning of an anomalous run, it 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, which learn non-anomalous feature representations of sequence data, are more robust to anomalous inputs. These models may struggle to accurately predict anomalous data, since it deviates from the learned pattern during training, leading to higher RMSE, which enables the detection of anomalous windows via thresholding. The proposed 1D-CNN model, representing a data-driven approach, shows promising results in anomaly detection, particularly for industrial cases where high accuracy is crucial.

7 Conclusions and Future Work

IT and OT convergence continues to accelerate the development of smart manufacturing systems, where ubiquitous network connectivity and automation optimize production process quality, output speed/volume, and reduce maintenance downtime. However, this greater connectivity and automation inversely lead to an expanded attack surface, exposing cyber-physical systems to attacks and exploitation. Now, more than ever, these can lead to cascading impacts and safety incidents across industrial operations. Today, while network security monitoring is heavily relied upon to detect threats across OT systems, network-based intrusion detection systems alone are not sufficient. Modern attackers targeting industrial domains often evade network monitoring tools by "living off the land" and using insecure-by-design industrial applications and devices for lateral system movement and attack execution. As the primary motivations for attacks against cyber-physical systems are sabotage or denial of service, where attackers aim to manipulate physical sensing or actuation, building resilience in detecting and responding to such incidents is key. Cyber-physical monitoring mechanisms that can learn and report abnormal physical and process behavior are crucial. Moreover, these mechanisms require a higher order of data integrity for analysis, which necessitates: i) segregated analysis mediums and data sources resistant to tampering, ii) low-resource edge computing systems practical for deployment, and iii) low-latency inference for rapid anomaly detection and response.

Toward addressing these challenges, we proposed CASPER, an out-of-band IoT anomaly detection system for cyber-physical systems that utilizes physical machine analytics to detect movement-based anomalies in an industrial robotic arm process. Our experimental results showed that a 1D-CNN-based model is capable of accurately detecting anomalies in the robotic system with comparable performance and lower detection latency than state-of-the-art machine learning and deep learning methods. Furthermore, our feature-design and model architecture enable the system to learn the behavior of time series (sequential) data, even when input features are highly correlated. For instance, the proposed model can detect a 5% decrease in joint velocities, the minimal applied deviation for the system. We also proposed and demonstrated the deployment of the anomaly detection system on an open-source IoT monitoring platform using BLE to transmit edge data via Node-RED. This exemplifies the feasibility of our approach as a practical retrofitted edge-computing platform for a realistic autonomous industrial endpoint system. Future research and development are expected to follow two paths: 1) The continued development of edge-based cyber-physical anomaly detection systems for industrial OT and IoT endpoints, using our architecture as a reference template, and 2) the exploration of 1D-CNNs as an effective machine learning architecture and model for resource-efficient and accurate AI-driven anomaly detection in resource-constrained edge security systems. In future work, we plan to expand the range of cyber-physical anomaly use cases (e.g., adding additional weight, touching the arm, shaking the testbed) to show the system's efficacy across various cyber-physical threats, implement online anomaly detection learning via cloud/fog system architecture, use quaternions as an anomaly detection feature to increase model accuracy and resource efficiency, and enable near real-time edge-based anomaly detection to minimize detection latency for rapid incident response and system recovery.

References

Adafruit. Adafruit feather nrf52840 sense, sep 2021. URL `https://learn.adafruit.com/adafruit-feather-sense`.

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

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

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

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

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

Apple. Track your sleep with apple watch, 2022. URL `https://support.apple.com/en-gb/guide/watch/apd830528336/watchos`.

Georgios Athanasakis, Gabriel Filios, Ioannis Katsidimas, Sotiris Nikoletseas, and Stefanos H Panagiotou. Tinyml-based approach for remaining useful life prediction of turbofan engines. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2022.

R Ganesh Babu, P Karthika, and V Aravinda Rajan. Secure iot systems using raspberry pi machine learning artificial intelligence. In *International conference on computer networks and inventive communication technologies*, pages 797–805, Cham, Switzerland, 2019. Springer.

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

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

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

Anatolij Bezemskij, George Loukas, Richard J Anthony, and Diane Gan. 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)*, pages 61–68, Granada, Spain, 2016. IEEE.

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

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

BleepingComputer. Sierra Wireless resumes production after ransomware attack, apr 2021. URL `https://www.bleepingcomputer.com/news/security/sierra-wireless-resumes-production-after-ransomware-attack/`.

Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series, 2021.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL `https://doi.org/10.1145/1541880.1541882`.

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

Heeryon Cho and Sang Min Yoon. Divide and conquer-based 1d cnn human activity recognition using test data sharpening. *Sensors*, 18(4):1055, 2018.

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

Amazon Elastic Compute Cloud. Amazon web services. *Retrieved November*, 9(2011):2011, 2011.

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

Comunicaffè. Caffitaly, gli hacker all'assalto delle capsule di Gaggio, February 2021. URL `https://www.comunicaffe.it/caffitaly-gli-haker-allassalto-delle-caspule-di-gaggio-montano/`.

Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, et al. Tensorflow lite micro: Embedded machine learning on tinyml systems. *arXiv preprint arXiv:2010.08678*, 2020.

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

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4027–4035, Virtual, 2021. AAAI Press.

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

Sinem Coleri Ergen. Zigbee/ieee 802.15. 4 summary. *UC Berkeley, September*, 10(17):11, 2004.

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

Pedro J Freire, Sasipim Srivallapanondh, Antonio Napoli, Jaroslaw E Prilepsky, and Sergei K Turitsyn. Computational complexity evaluation of neural network applications in signal processing. *arXiv preprint arXiv:2206.12191*, 2022.

Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. 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*, KDD '05, page 401–410, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 159593135X. doi: 10.1145/1081870.1081917. URL `https://doi.org/10.1145/1081870.1081917`.

Yang Gao, Borui Li, Wei Wang, Wenyao Xu, Chi Zhou, and Zhanpeng Jin. 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):1–27, 2018.

Zhiwei Gao, Carlo Cecati, and Steven X Ding. 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): 3757–3767, 2015.

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

Mohammed Ghazal, Tasnim Basmaji, Maha Yaghi, Mohammad Alkhedher, Mohamed Mahmoud, and Ayman S El-Baz. Cloud-based monitoring of thermal anomalies in industrial environments using ai and the internet of robotic things. *Sensors*, 20(21):6348, 2020.

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

Victor Gonzalez-Huitron, José A León-Borges, AE Rodriguez-Mata, Leonel Ernesto Amabilis-Sosa, Blenda Ramírez-Pereda, and Hector Rodriguez. Disease detection in tomato leaves via cnn with lightweight architectures implemented in raspberry pi 4. *Computers and Electronics in Agriculture*, 181:105951, 2021.

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

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

Juan Haladjian, Daniel Schlabbers, Sajjad Taheri, Max Tharr, and Bernd Bruegge. Sensor-based detection and classification of soccer goalkeeper training exercises. *ACM Transactions on Internet of Things*, 1(2), apr 2020. ISSN 2691-1914. doi: 10.1145/3372342. URL `https://doi.org/10.1145/3372342`.

Danfeng Hong, Naoto Yokoya, Gui-Song Xia, Jocelyn Chanussot, and Xiao Xiang Zhu. X-modalnet: A semi-supervised deep cross-modal network for classification of remote sensing data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167:12–23, 2020.

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

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

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

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

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

Tariqul Islam, Md Saiful Islam, Md Shajid-Ul-Mahmud, and Md Hossam-E-Haider. Comparison of complementary and kalman filter based data fusion for attitude heading reference system. In *AIP Conference Proceedings*, volume 1919, page 020002, New York, NY, USA, 2017. AIP Publishing LLC.

Gopal Chandra Jana, Ratna Sharma, and Anupam Agrawal. A 1d-cnn-spectrogram based approach for seizure detection from eeg signal. *Procedia Computer Science*, 167:403–412, 2020.

Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245, 2019.

Hakan Kayan, Yasar Majib, Wael Alsafery, Mahmoud Barhamgi, and Charith Perera. Anoml-iot: An end to end re-configurable multi-protocol anomaly detection pipeline for internet of things. *Internet of Things*, 16:100437, 2021.

Hakan Kayan, Matthew Nunes, Omer Rana, Pete Burnap, and Charith Perera. Cybersecurity of industrial cyber-physical systems: A review. *ACM Comput. Surv.*, 54(11s), sep 2022. ISSN 0360-0300.

Haider Adnan Khan, Nader Sehatbakhsh, Luong N Nguyen, Robert L Callan, Arie Yeredor, Milos Prvulovic, and Alenka Zajić. Idea: Intrusion detection through electromagnetic-signal analysis for critical embedded and cyber-physical systems. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1150–1163, 2019.

Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398, 2021.

Daniel Knight. Dietpi os, dec 2021. URL `https://dietpi.com/`.

Moshe Kravchik and Asaf Shabtai. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 72–83, New York, NY, USA, 2018. ACM.

Andrew Kusiak. Smart manufacturing. *International Journal of Production Research*, 56(1-2):508–517, 2018.

Prasanth Lade, Rumi Ghosh, and Soundar Srinivasan. Manufacturing analytics and industrial internet of things. *IEEE Intelligent Systems*, 32(3):74–79, 2017.

Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. *Journal of machine learning research*, 10(1):1–40, 2009.

Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 703–716, Cham, Switzerland, 2019a. Springer.

Guangxia Li, Yulong Shen, Peilin Zhao, Xiao Lu, Jia Liu, Yangyang Liu, and Steven CH Hoi. Detecting cyberattacks in industrial control systems using online learning algorithms. *Neurocomputing*, 364:338–348, 2019b.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021.

Zhe Li, Jingyue Li, Yi Wang, and Kesheng Wang. A deep learning approach for anomaly detection based on sae and lstm in mechanical equipment. *The International Journal of Advanced Manufacturing Technology*, 103(1):499–510, 2019c.

Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422, Pisa, Italy, 2008. IEEE.

Qi Liu, Rudy Klucik, Chao Chen, Glenn Grant, David Gallaher, Qin Lv, and Li Shang. Unsupervised detection of contextual anomaly in remotely sensed data. *Remote Sensing of Environment*, 202:75–87, 2017.

Marc Moreno Lopez and Jugal Kalita. Deep learning applied to nlp, 2017.

Huimin Lu, Yujie Li, Shenglin Mu, Dong Wang, Hyoungseop Kim, and Seiichi Serikawa. Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE internet of things journal*, 5(4):2315–2322, 2017.

Simone A Ludwig, Kaleb D Burnham, Antonio R Jiménez, and Pierre A Touma. 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*, volume 10598, pages 644–650, Washington, DC, USA, 2018. SPIE.

Chunjie Luo, Fan Zhang, Cheng Huang, Xingwang Xiong, Jianan Chen, Lei Wang, Wanling Gao, Hainan Ye, Tong Wu, Runsong Zhou, et al. Aiot bench: towards comprehensive benchmarking mobile and embedded device intelligence. In *International Symposium on Benchmarking, Measuring and Optimization*, pages 31–35, Cham, Switzerland, 2018. Springer.

Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE international conference on rehabilitation robotics*, pages 1–7, Zurich, Switzerland, 2011. IEEE.

Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control*, 53(5):1203–1218, 2008.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, volume 89, pages 89–94, Bruges, Belgium, 2015. IEEE.

S Manimurugan. Iot-fog-cloud model for anomaly detection using improved naïve bayes and principal component analysis. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–10, 2021.

Aditya P Mathur and Nils Ole Tippenhauer. 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)*, pages 31–36, Porto, Portugal, 2016. IEEE.

Siamak Mehrkanoon. Deep shared representation learning for weather elements forecasting. *Knowledge-Based Systems*, 179:120–128, 2019.

Microsoft. Azure, mar 2022. URL `https://azure.microsoft.com/en-gb/`.

Charlie Miller and Chris Valasek. A survey of remote automotive attack surfaces. *black hat USA*, 2014:94, 2014.

Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

Ali Moin, Andy Zhou, Abbas Rahimi, Alisha Menon, Simone Benatti, George Alexandrov, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1):54–63, 2021.

Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. Cnn-based sensor fusion techniques for multimodal human activity recognition. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 158–165, New York, NY, USA, 2017. ACM.

Andrew Murphy. Industrial: Robotics outlook 2025, feb 2022. URL `https://loupfunds.com/industrial-robotics-outlook-2025/`.

Vedanth Narayanan and Rakesh B. Bobba. Learning based anomaly detection for industrial arm applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, CPS-SPC '18, page 13–23, New York, NY, USA, 2018a. Association for Computing Machinery. ISBN 9781450359924. doi: 10.1145/3264888.3264894. URL `https://doi.org/10.1145/3264888.3264894`.

Vedanth Narayanan and Rakesh B. Bobba. Learning Based Anomaly Detection for Industrial Arm Applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 13–23, Toronto Canada, January 2018b. ACM. ISBN 978-1-4503-5992-4. doi: 10.1145/3264888.3264894. URL `https://dl.acm.org/doi/10.1145/3264888.3264894`.

Mao V. Ngo, Tie Luo, and Tony Q. S. Quek. Adaptive anomaly detection for internet of things in hierarchical edge computing: A contextual-bandit approach. *ACM Transactions on Internet of Things*, 3(1), oct 2021. ISSN 2691-1914. doi: 10.1145/3480172. URL `https://doi.org/10.1145/3480172`.

Long D Nguyen, Dongyun Lin, Zhiping Lin, and Jiuwen Cao. Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, New York, NY, USA, 2018. IEEE.

Zhiyou Ouyang, Xiaokui Sun, Jingang Chen, Dong Yue, and Tengfei Zhang. Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things. *IEEE Access*, 6:9623–9631, 2018.

Simone Panicucci, Nikolaos Nikolakis, Tania Cerquitelli, Francesco Ventura, Stefano Proto, Enrico Macii, Sotiris Makris, David Bowden, Paul Becker, Niamh O'Mahony, et al. A cloud-to-edge approach to support predictive analytics in robotics industry. *Electronics*, 9(3):492, 2020.

Donghyun Park, Seulgi Kim, Yelin An, and Jae-Yoon Jung. Lired: A light-weight real-time fault detection system for edge computing using lstm recurrent neural networks. *Sensors*, 18(7):2110, 2018.

Koeppe Patrick. HUBER+SUHNER : gradually resumes production after cyberattack | MarketScreener, dec 2020. URL `https://www.marketscreener.com/quote/stock/HUBER-SUHNER-AG-278523/news/HUBER-SUHNER-nbsp-gradually-resumes-production-after-cyberattack-32074407/`.

D Pavithra and Ranjith Balakrishnan. Iot based monitoring and control system for home automation. In *2015 global conference on communication technologies (GCCT)*, pages 169–173. IEEE, 2015.

Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, and Félix J García Clemente. Madics: A methodology for anomaly detection in industrial control systems. *Symmetry*, 12(10):1583, 2020.

SR Prathibha, Anupama Hongal, and MP Jyothi. Iot based monitoring system in smart agriculture. In *2017 international conference on recent advances in electronics and communication technology (ICRAECT)*, pages 81–84. IEEE, 2017.

Associated Press. Hacker tries to poison water supply in Florida city, feb 2021. ISSN 0307-1235. URL `https://www.telegraph.co.uk/news/2021/02/09/hacker-tries-poison-water-supply-florida-city/`.

Australian Associated Press. Systems shut down in Victorian hospitals after suspected cyber attack, oct 2019. URL `http://www.theguardian.com/australia-news/2019/oct/01/systems-shut-down-in-victorian-hospitals-after-suspected-cyber-attack`.

Mohammad Riazi, Osmar Zaiane, Tomoharu Takeuchi, Anthony Maltais, Johannes Günther, and Micheal Lipsett. Detecting the onset of machine failure using anomaly detection methods. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 3–12, Cham, Switzerland, 2019. Springer.

Mauro Ribeiro, Katarina Grolinger, and Miriam A.M. Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902, New York, NY, USA, 2015. IEEE.

Haakon Ringberg, Matthew Roughan, and Jennifer Rexford. The need for simulation in evaluating anomaly detectors. *ACM SIGCOMM Computer Communication Review*, 38(1):55–59, 2008.

Alina Roitberg, Nikhil Somani, Alexander Perzylo, Markus Rickert, and Alois Knoll. Multimodal human activity recognition for industrial manufacturing processes in robotic workcells. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 259–266, New York, NY, USA, 2015. ACM.

Beth Romanik. Prison computer 'glitch' blamed for opening cell doors in maximum-security wing, aug 2013. URL `https://www.techwell.com/techwell-insights/2013/08/computer-glitch-blamed-opening-prison-cell-doors`.

Ellen Rushe and Brian Mac Namee. Anomaly detection in raw audio using deep autoregressive networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3597–3601, Brighton, UK, 2019. IEEE.

Ali M Sadeghioon, Nicole Metje, David Chapman, and Carl Anthony. Water pipeline failure detection using distributed relative pressure and temperature measurements and anomaly detection algorithms. *Urban Water Journal*, 15(4): 287–295, 2018.

Anam Sajid, Haider Abbas, and Kashif Saleem. Cloud-assisted iot-based scada systems security: A review of the state of the art and future challenges. *IEEE Access*, 4:1375–1384, 2016.

Yasushi Sakurai, Yasuko Matsubara, and Christos Faloutsos. Mining and forecasting of big time-series data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 919–922, New York, NY, USA, 2015. ACM.

Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks, 2017.

Raed Abdel Sater and A. Ben Hamza. A federated learning approach to anomaly detection in smart buildings. *ACM Transactions on Internet of Things*, 2(4), aug 2021. ISSN 2691-1914. doi: 10.1145/3467981. URL https://doi.org/10.1145/3467981.

Debarshi Sen, Amirali Aghazadeh, Ali Mousavi, Satish Nagarajaiah, Richard Baraniuk, and Anand Dabak. Data-driven semi-supervised and supervised learning algorithms for health monitoring of pipes. *Mechanical Systems and Signal Processing*, 131:524–537, 2019.

Gauri Shah and Aashis Tiwari. 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*, pages 295–300, Goa, India, 2018. ACM.

Syed Maaz Shahid, Sunghoon Ko, and Sungoh Kwon. Performance comparison of 1d and 2d convolutional neural networks for real-time classification of time series sensor data. In *2022 International Conference on Information Networking (ICOIN)*, pages 507–511, 2022. doi: 10.1109/ICOIN53446.2022.9687284.

Matti Siekkinen, Markus Hiienkari, Jukka K Nurminen, and Johanna Nieminen. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4. In *2012 IEEE wireless communications and networking conference workshops (WCNCW)*, pages 232–237, Paris, France, 2012. IEEE.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11):4724–4734, 2018. doi: 10.1109/TII.2018.2852491.

Daniel Sonntag, Sonja Zillner, Patrick van der Smagt, and András Lörincz. Overview of the cps for smart factories project: Deep learning, knowledge acquisition, anomaly detection and intelligent user interfaces. In *Industrial internet of things*, pages 487–504. Springer, Cham, Switzerland, 2017.

Thomas Stibor, Jonathan Timmis, and Claudia Eckert. A comparative study of real-valued negative selection to statistical anomaly detection techniques. In *International Conference on Artificial Immune Systems*, pages 262–275, Berlin, Germany, 2005. Springer.

Ljiljana Stojanovic, Marko Dinic, Nenad Stojanovic, and Aleksandar Stojadinovic. 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)*, pages 1647–1652, Washington, DC, USA, 2016. IEEE.

Abdulhamit Subasi, Dalia H Dammas, Rahaf D Alghamdi, Raghad A Makawi, Eman A Albiety, Tayeb Brahimi, and Akila Sarirete. Sensor based human activity recognition using adaboost ensemble classifier. *procedia computer science*, 140:104–111, 2018.

Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. *Bio-inspired neurocomputing*, pages 203–224, 2021.

Rui Tan, Varun Badrinath Krishna, David KY Yau, and Zbigniew Kalbarczyk. Impact of integrity attacks on real-time pricing in smart grids. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 439–450, New York, NY, USA, 2013. ACM.

Pavol Tanuska, Lukas Spendla, Michal Kebisek, Rastislav Duris, and Maximilian Stremy. Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0. *Sensors*, 21(7):2376, 2021.

The Arduino Team. Nicla sense me, dec 2021a. URL http://store.arduino.cc/collections/sensors-environment/products/nicla-sense-me.

The Arduino Team. Nano 33 ble sense: Arduino documentation, dec 2021b. URL https://docs.arduino.cc/hardware/nano-33-ble-sense.

Joe Tidy. Colonial hack: How did cyber-attackers shut off pipeline?, may 2021. URL https://www.bbc.com/news/technology-57063636.

Chi-Ho Tsang and Sam Kwong. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. In *2005 IEEE international conference on industrial technology*, pages 51–56, Hong Kong, China, 2005. IEEE.

David I Urbina, David I Urbina, Jairo Giraldo, Alvaro A Cardenas, Junia Valente, Mustafa Faisal, Nils Ole Tippenhauer, Justin Ruths, Richard Candell, and Henrik Sandberg. *Survey and new directions for physics-based attack detection in control systems*. US Department of Commerce, NIST, Maryland, USA, 2016.

Tuan Vuong, Avgoustinos Filippoupolitis, George Loukas, and Diane Gan. Physical indicators of cyber attacks against a rescue robot. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 338–343, New York, NY, USA, 2014. IEEE.

Kun Wang, Yihui Wang, Yanfei Sun, Song Guo, and Jinsong Wu. Green industrial internet of things architecture: An energy-efficient perspective. *IEEE Communications Magazine*, 54(12):48–54, 2016. doi: 10.1109/MCOM.2016.1600399CM.

Leyi Wei, Shixiang Wan, Jiasheng Guo, and Kelvin KL Wong. A novel hierarchical selective ensemble classifier with bioinformatics application. *Artificial intelligence in medicine*, 83:82–90, 2017.

Actusnews Wire. Mnd, mar 2021. URL `https://www.actusnews.com/en/mnd/pr/2021/03/24/mnd-statement-on-cyber-attack`.

Dazhong Wu, Shaopeng Liu, Li Zhang, Janis Terpenny, Robert X Gao, Thomas Kurfess, and Judith A Guzzo. A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. *Journal of Manufacturing Systems*, 43:25–34, 2017.

Mingtao Wu, Zhengyi Song, and Young B Moon. Detecting cyber-physical attacks in cybermanufacturing systems with machine learning methods. *Journal of intelligent manufacturing*, 30(3):1111–1123, 2019.

Weizhong Yan and Lijie Yu. Neural contextual anomaly detection for time series, 2019.

Hasan Yetis and Mehmet Karakose. Image processing based anomaly detection approach for synchronous movements in cyber-physical systems. In *2018 23rd International Scientific-Professional Conference on Information Technology (IT)*, pages 1–4, Zabljak, Montenegro, 2018. IEEE.

Dong Yi, Zhen Lei, and Stan Z Li. Shared representation learning for heterogenous face recognition. In *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, volume 1, pages 1–7, Ljubljana, Slovenia, 2015. IEEE.

Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 98:289–330, 2019.

Dingwen Zhang, Guohai Huang, Qiang Zhang, Jungong Han, Junwei Han, and Yizhou Yu. Cross-modality deep feature learning for brain tumor segmentation. *Pattern Recognition*, 110:107562, 2021.

Fukai Zhang, Ce Li, and Feng Yang. Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation. *Sensors*, 19(3):594, 2019.