Journal of Network and Computer Applications 226 (2024) 103888

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Research paper

Digital twin-driven secured edge-private cloud Industrial Internet of Things (IIoT) framework

Muna Al-Hawawreh [a], M. Shamim Hossain [b,*]

[a] School of Information Technology, Deakin University, Geelong, 3216, Victoria, Australia
[b] Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, 12372, Saudi Arabia

ARTICLE INFO

Keywords:
Attack detection
Online machine learning
Ensemble
IIoT
Digital twin
Edge cloud

ABSTRACT

With the growing popularity of Industrial Internet of Things (IIoT) technologies and the recent development of edge private cloud systems to fulfill the demands of industrial environments for high data rates, low latency, and powerful computing and storage resources, the security of these systems has become an increasingly important concern. Many existing machine learning-based attack detection models for the IIoT face challenges in the early identification of attacks. This is due to network traffic and physical process data's heterogeneous, high-dimensional, and unbalanced nature. Moreover, these models are typically trained offline and deployed in the cloud or embedded in devices, leading to resource strain and delayed attack detection. Thus, this paper proposes a real-time security framework for detecting attacks and mitigating their impact using Digital Twin and online ensemble machine learning. We explore various ensemble techniques and algorithms and evaluate their performance using gas pipeline and X-IIoTID datasets. The experimental results illustrate the effective performance of the proposed framework for detecting attacks, showcasing a comparable efficiency to offline ensemble techniques.

1. Introduction

The rapid expansion of the Industrial Internet of Things (IIoT), edge and cloud computing, and fifth-generation (5G) communication with the digitization of physical and industrial processes have enabled the real-time collection and analysis of data, which can be used to improve the efficiency and productivity of the deployed systems (Ferrag et al., 2023). Given the critical and time-sensitive nature of the physical and industrial processes of these IIoT systems, many organizations and industries deploy edge computing capabilities—that bring computation and data storage closer to the sources of data—on-premises, forming what is referred to as an edge private cloud system (Yang et al., 2023; Kar et al., 2023; Cao et al., 2021). The edge-private cloud system is a small-sized cloud data center that deploys the capabilities of edge and cloud computing with private 5G in the on-premises or the local physical area of the organization or industry, incorporating a variety of stationary and mobile devices such as mobiles, sensors, actuators, Remote Terminal Unit (RTU) and Programmable Logic Controller (PLC) (Cao et al., 2021; Sonkoly et al., 2020). These systems provide an ultra-low latency edge platform that improves bandwidth and utilization of high-end devices and enhances the industrial and physical processes' performance and efficiency (Kar et al., 2023). Although deploying these systems strengthens security and privacy by collecting and analyzing critical real-time data within the local data center and minimizing external communication and data transfer, they are not immune to cyberattacks.

The security of edge private cloud systems is usually handled by the organization or industry (not the cloud provider), which makes these systems, similar to other systems of any industry, vulnerable to many cyberattacks, including denial of service, ransomware, data exfiltration, and insider malicious, making their security, in particular in remote locations (Cao et al., 2021). Efficient security solutions are required to protect the devices and networks of these systems. However, the design of security solutions for these systems is more challenging due to (1) the time-sensitivity of deployed IIoT devices, (2) the presence of heterogeneous devices, and (3) a multi-type of communications (He et al., 2024b).

Existing traditional security solutions, proposed in studies such as Yang et al. (2023), Alghamdi and Bellaiche (2021), de Araujo-Filho et al. (2020) presented attack detection and prevention using machine learning (ML) for mobile, fog, or edge cloud but were introduced within or close to systems and devices after they deployed and operated in the environment, instead of during the initial design process. These solutions also have reactive capabilities and highly depend on offline

* Corresponding author.
E-mail addresses: muna.alhawawreh@deakin.edu.au (M. Al-Hawawreh), mshossain@ksu.edu.sa (M. Shamim Hossain).

https://doi.org/10.1016/j.jnca.2024.103888
Received 9 March 2024; Received in revised form 25 March 2024; Accepted 4 April 2024
Available online 24 April 2024

attack analysis and detection. Existing security solutions and attack detection models have a vulnerability—concept drift—that renders them swiftly outdated and ineffective. Concept drift can occur for various reasons. For instance, attackers might introduce new malicious functionalities, alter their applications to elude detection or devise novel malware tactics and techniques that have not been encountered before. Additionally, legitimate data with newly deployed IIoT technologies continuously changes (Chen et al., 2023). The state-of-the-art solutions depend on offline, fixed, and active learning, where new test samples are selected to evaluate the models, and these samples are then added to the training data to retrain the ML model/classifier. In addition, these ML algorithms achieve high performance during training. But when we deploy these solutions in the real world, their performance drops, providing inconsistent and less robust security solutions. Hence, such solutions are time-consuming processes that may lead to the late identification of attacks, while the goal is to detect the attack early and efficiently use the system resources and the security analysts' time.

Therefore, we propose a real-time security framework for protecting edge private cloud systems that combines continuous learning for ML with real-time monitoring and analysis of collected data automation and Digital Twin (DT) technology. Continuous learning based on ensemble ML will handle concept drift with new attack tactics, techniques, and dynamic changes in legitimate data and traffic. This unique combination enables efficient and robust security that was previously technically or operationally infeasible without these technologies, particularly the digital twin. A DT is a powerful new technology that can help organizations understand their systems and data and boost their efficiency (Xu et al., 2023). It is a virtual representation of the physical systems and devices, such as controllers, robots, sensors, or edge gateways, continuously updated with data about the operations of these systems and devices. In cybersecurity, researchers have identified the possibility of using DT to proactively detect cyberattacks and prevent their malicious impact on critical processes (Salim et al., 2022). DT can help create and design real-time detection and mitigation modules by identifying any behavior deviation between the real physical system and the virtual one (a replica of the real physical system) and mitigating the impact of attacks using virtual resources. Such a design can preserve the computing resources of the real system and avoid any impact on its operations and efficiency. This is particularly important for IIoT systems and industrial environments due to the real-time constraints.

The main contribution of this paper is described as follows:

1. We propose a digital twin-enabled real-time security framework for securing the edge private cloud systems in industrial environments (e.g., gas pipeline systems).
2. We present a real-time attack detection model using online ensemble machine learning, testing various ensemble techniques.
3. We provide a comprehensive analysis of the performance of the online and offline ensemble algorithms using two datasets (X-IIoTID network and gas pipeline).

The remainder of this paper is organized as follows. In Section 2, we present the background and related work, after which, in Section 3, we introduce our proposed security framework and discuss its various components. In Section 4, we introduce the experimental evaluation of the proposed framework. We then conclude the paper with our significant findings in Section 5.

2. Background and related works

2.1. Background

2.1.1. Digital twins and cybersecurity

Digital Twin (DT) is defined as a software or computer-based model that simulates, emulates or twins the life of a physical machine or system. The main aim of implementing DT is to describe the physical process by using specification-based techniques, mathematical models, and application programming interfaces (APIs) that run on servers and/or virtualized resources, such as virtual machines (VMs) and containers (Alcaraz and Lopez, 2022; Li et al., 2022). This replica of a physical process or system is beneficial for predicting potential errors, variations, and significant deviations that could alter the natural behavior of a system.

A DT distinguishes itself from other mirroring systems like conventional simulators, digital shadows, or digital models. DT establishes a bidirectional and automatic connection between the physical and digital realms, ensuring granular and precise representations of the system and its parameters (Sai et al., 2023). In contrast, conventional simulators lack this level of accuracy, while digital models operate in isolation without automatic ties to the real world, and digital shadows involve automated one-way communication between the physical and virtual spaces. DT is currently used in many applications to create accurate virtual representations of assets and simulations of physical processes. Examples of these applications include designing new products and process optimization in manufacturing, farm management and resource optimization in agriculture, drug development, advanced diagnosis, and preventive treatment in healthcare (Attaran and Celik, 2023; Sai et al., 2024).

Implementing DT in the IIoT environment can significantly improve the security of IIoT devices and systems due to the provided digital replicas in the real-world environment (Xu et al., 2023). Thus, DT provides a better understanding of how a specific system or device works and connects with other devices and the Internet, enabling the identification of any potential security risks before they cause any harmful actions or damage to the system. After identifying security risks, digital twins offer a valuable tool for formulating effective mitigation strategies. These proactive measures may encompass the implementation of security controls, firmware updates, and vigilant monitoring for any signs of suspicious activity. Utilizing DT also enables the thorough testing of security controls before deploying them in a real-world environment. This testing phase is crucial for pinpointing any potential vulnerabilities in the security framework and verifying the overall effectiveness of the implemented controls.

A recent study by NIST and the University of Michigan (Balta et al., 2023) demonstrated the great benefit of using DT to enhance the cybersecurity of 3D printers at the factory. Their study used DT to replicate the 3D printing process, equipping it with real data from the lab's printer. Throughout the printing process, computer programs monitored and analyzed continuous data streams, including the actual temperature of the physical print and the temperature calculated by the digital twin. They also used Machine Learning-based programs with pattern recognition capabilities trained on standard operating data to differentiate between malicious attacks and routine anomalies. Then, researchers tried to perform a cyber attack to perturbate the printer. Consequently, the DT could identify normal conditions for the printer and detect any deviations indicating a potential threat. When the model detects abnormal behavior, it transmits the information to other models that compare the unusual signals against a library of known issues. Subsequently, the system classifies the abnormal behavior as an anticipated anomaly or a potential cyber threat. Ultimately, a human expert is responsible for analyzing the data, deciding, and taking appropriate action.

Alongside this study, we also present a digital twin as a foundation for the security framework in the IIoT environment by designing and building a digital twin-enabled attack detection and mitigation modules to protect the physical edge IIoT gateway devices and the virtual replica of advanced cyber attacks and mitigate their impact on the system operations and functions in edge private cloud systems.

2.2. Related work

This subsection examines the latest research studies in edge and private cloud security, including digital twin security, focusing on attack detection.

2.2.1. Related work to edge and private cloud

Gumaste et al. (2020) presented a DDoS attack detection for the private cloud. The proposed model depends on collecting traffic coming to the VM server from other VMs and external networks every two seconds. The preprocessing of this traffic and the feature extraction have been done using Spark, a distributed platform for faster data preprocessing, while different machine learning was used for attack detection. The designed model achieves a good performance. In the study of Arzani et al. (2020), the authors proposed a framework for trusted and secured detection of compromised VMs in the private cloud. The authors used a MinHash method and developed a similarity classifier to distinguish between clean and infected volatile memory dumps. The similarity classifier achieved the best performance detecting malware compared with random forest and decision tree.

Singh et al. (2022) proposed a hybrid edge attack detection model. Their proposed model uses different machine learning at multiple levels, including decision tree, Naive base, and AdaBoost. Al-Hawawreh and Hossain (2023) presented a new privacy-preserving intrusion detection framework for identifying cyber attacks against edge medical devices. The authors used a contractive autoencoder with differential privacy for fusing various data sources and then the quantum deep neural network to distinguish between normal and attack. Their proposed model achieved good performance in detecting attacks while protecting data privacy.

Tian et al. (2019) proposed a distributed attack detection for detecting web attacks at edge devices. Their proposed model was built using a deep learning algorithm, i.e., M-ResNet, where different models will be trained using Web URLs separately and deployed on different servers. The appropriate action will be taken based on the results from concurrent models. Although the experimental results showed the proposed detection model's efficiency, its capabilities are limited to specific attacks and cannot be used to detect them. In the study of Al-Hawawreh et al. (2021a), the authors proposed a ransomware detection model for protecting edge devices in IIoT systems. Their proposed model used asynchronous peer-to-peer federated deep learning, combining data purifying and diagnostic and decision modules. Contractive Denoising Auto-Encoder powered the purifying module to filter the input data and protect against adversarial evasion attacks. A deep neural network powered the detection module to identify any malicious behavior related to ransomware attacks. While all these models achieved a good performance, they were designed offline and not deployed as real-time security measures in real-world applications.

2.2.2. Related work to digital twin

In the study of Tärneberg et al. (2021), the authors proposed an intrusion detection embedded with the industrial controller cloud-based digital twin to protect the connected industrial control systems against cyber attacks. The proposed system depends on a rule-based that uses the physical properties of the industrial process in question. The model's evaluation was based on a proof-of-concept consisting of a cyber–physical system (CPS) (Hossain, 2017), a custom off-the-shelf (COTS)-based cloud-native digital twin, and a Kubernetes cluster. Although the proposed model proved its efficiency in detecting and mitigating attacks, it inherits the limitations of the traditional rule-based intrusion detection systems. Akbarian et al. (2020) depended on comparing the predicted output signal based on the Kalman Filter with the output signal from the digital twin to detect attacks. The authors also used a multiclass support vector machine to classify the detected cyberattacks. Similar to the previous study, the intrusion detection system is included within a digital twin. He et al. (2024a) proposed The authors provide an innovative approach to intrusion detection that are 6G enabled. This approach uses digital twins and federated meta-learning in imbalanced data settings. The authors built a meta-sampler that discovers the sampling method using FL's meta-learning capabilities. During the meta-training phase, the meta-sampler is fine-tuned using reinforcement learning to continually sample the client-side balanced dataset. This method greatly reduces the effect of the training data class imbalance problems.

In the study of Balta et al. (2023), the authors presented a digital twin framework for detecting cyber attacks against manufacturing systems and differentiating between cyberattacks and faults in the closed control loop system. The main contribution of this study was the cybersecurity digital twin, which consists of anomaly detection DT based on one class support vector machine and the consistency DT, which performs further analysis for any data to determine if the abnormal data is an attack or fault. Their proposed framework was tested using off-the-shelf 3D printers, demonstrating its effectiveness in detecting cyberattacks.

Salim et al. (2022) proposed a new blockchain-enabled DT framework for securing IoT devices against botnets. Each DT is synchronized with a group of IIoT devices and collects and inspects packet headers using deep learning to detect suspicious network communication with external IP addresses. Any DT is compromised by attackers as part of the botnet network and is isolated using certificate revocation to prevent spreading. The experimental results show that the designed blockchain-DT framework could detect botnet attacks and maintain the data integrity for training the botnet detection model. In related work, Feng et al. (2023) presented anomaly detection and trust management framework using blockchain for protecting IIoT devices at the edge layer. In this framework, each DT is connoted with a group of IoT devices and deployed in the edge server, creating a self-healing architecture to improve anomaly detection-based deep learning. The simulation results proved the security and efficiency of the proposed framework.

Alongside these studies, we present a new framework for security using DT to protect edge private cloud for smart gas pipeline systems in the industrial environment. The DT is used to deploy the real-time detection module and mitigate the impact of any detected malicious activity.

3. Proposed digital twin-enabled security framework for edge private cloud systems in IIoT environment

3.1. Overview

The proposed digital twin-enabled security framework aims to identify the malicious behavior within the system and mitigate its impact. In practice, it predicts the network behavior that is likely to be malicious and then mitigates the effects of the attack by reducing the traffic and data selected from this device and replacing it with legitimate data from the digital twin to ensure continuous system operations and physical processes. Fig. 1 illustrates the proposed security framework for edge private cloud systems and the attacker activities against these connected devices and systems. The edge private cloud system, which consists of a set of sensors (e.g., pressure transducer), actuators (e.g., pressure switch), and controller (e.g., Remote terminal unit (RTU)), and smartphone, connects with edge gateways at conventional edge computing layer. These edge gateways communicate with others and connect these devices with platform and enterprise layers over the Internet, where the data of these devices are stored and analyzed for further insights and decisions about the physical process of gas pipeline systems. All edge gateways at the edge layer are replicated using digital twins embedded with a real-time detection model. The digital twins are connected with the zones of these devices and the upper layers synchronously to predict any malicious activities against the edge of private cloud devices and physical processes that could come from the cyber or physical world. In the attack scenario, malicious attackers can compromise the operator's mobile, sensor, RTU, or any connected IoT devices in these lower layers. The attacker could then perform various malicious activities, such as modifying the collected data from the sensor, poisoning the transmitted data to the edge server, or sending vast amounts of packets to the controller to shut it down.

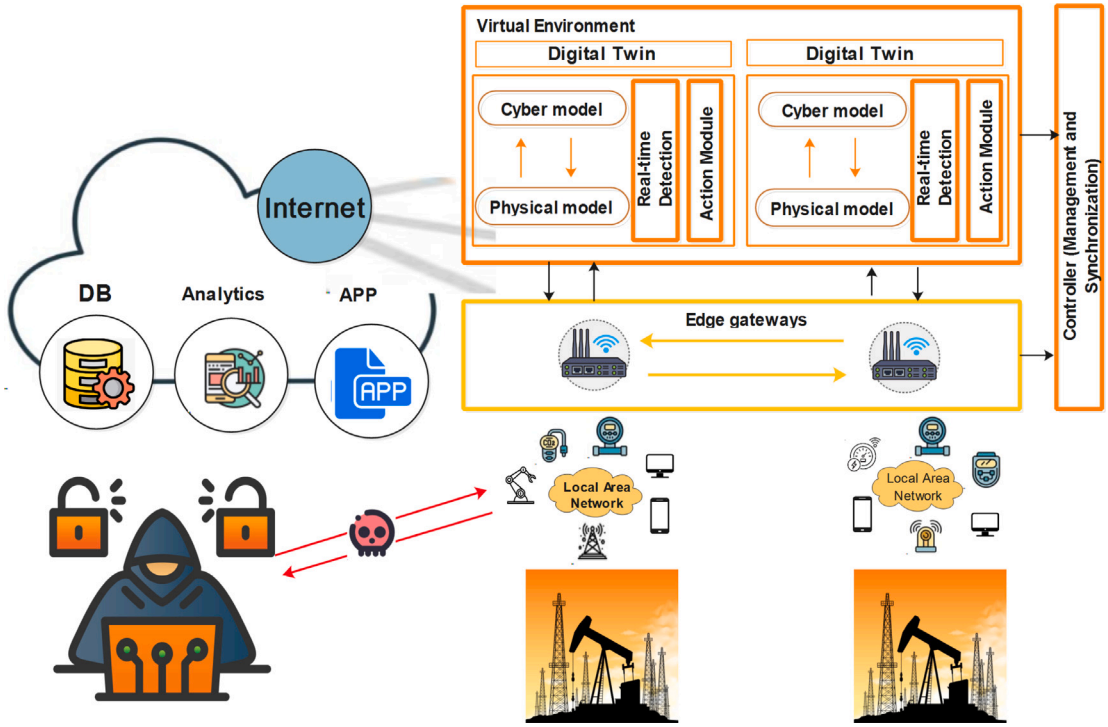


Fig. 1. The architecture of digital twin-enabled security framework for edge private cloud systems in IIoT environment.

With a real-time ensemble detection model within the digital twin, the attack will be detected early, and the cyber controller will take the appropriate action or response without any impact on the operational physical devices.

The dynamic nature of our proposed DT-based security framework facilitates real-time attack detection by continuously monitoring the edge gateways and connected devices, updating the DT with real-time data from data sources, analyzing the behavior of connected devices and systems, and utilizing online ensemble machine learning. The details of the components of the proposed framework are described in the following subsections.

3.1.1. Proposed digital twin-based solution

In our proposed solution, we assume the DT replicates the edge gateway, which connects the edge private cloud systems with the enterprise layer over the Internet. The edge gateway connects the cyber world with the physical world. Thus, it receives data on physical and industrial processes and new IT technologies. Therefore, we create a networking digital twin, which includes a physical model and a network (cyber) model (as shown in Fig. 1). The former model focuses on analyzing data related to the physical processes and industrial devices such as sensors, actuators, and controllers. It uses real-time data to enable learning and reasoning the full closed-control loop system and its communications. The latter model uses real-time data to understand and learn the cyber world and IT communications. Both models communicate and maintain the current status of physical processes and networks through and continuously update the models from multiple sources throughout their lifespan. These models within DT provide a comprehensive overview of the edge gateway and connected devices' state, facilitating real-time attack identification.

The DT synchronizes its models for its physical and cyber spaces/sides. This involves aligning scenarios by matching contextual parameters such as sensor readings and actuator status for the physical side and communications with the enterprise level on the cyber side. Each DT (i.e., simulator) has bidirectional interfaces and operates by receiving information and then juxtaposing it with its internally processed information. If any attack is detected, the DT will identify and react. For example, if the attack is detected in the physical industrial processes, the DT will compare the received data from the physical processes with the expected ones. Then, the DT will react to mitigate the impact of this attack through the action module. The controller will handle the synchronization processes between the DT and the physical world and will be responsible for performing the attack mitigation actions. The details of designed and deployed attack identification and action modules are explained in the following subsections.

3.1.2. Real-time detection module

Our proposed real-time detection framework is composed of data preparation, data identification (as shown in Fig. 2).

Data Preparation Module: The data collected from connected physical devices through the DT in real-time situations—due to the synchronization approach between DT and physical devices—is sent to the preparation module at DTs. In this module, one part focuses on the network data, which is collected based on the traffic flow, including source IP, destination IP, source port, and destination Port and the second part focuses more on the physical industrial process parameters for connected devices at edge private cloud systems. Different network traffic characteristics of features can be created based on the data flows. For example, a sample of network traffic features (X_n) includes Protocol, service, duration, originated bytes, response bytes, received packets, total number of bytes, byte rate, average packet size. Also, many features can be automatically created based on the network flow using a snippet of code and mathematical equations, which helps identify the normal and malicious activities in the next module. Examples of these features include the total number of bytes exchanged between source and destination, the total number of packets exchanged between source and destination, the percentage of receiving bytes to the total bytes, and the percentage of sending packets to the total packets.

The data preparation for the industrial process differs from the network traffic as it does not include routing and flows, and only the parameters of the process and the industrial protocols from each device are collected. However, the data is organized to represent the closed control loop process, where for each sensor measurement, the status of the actuator, controller and even the industrial protocol will

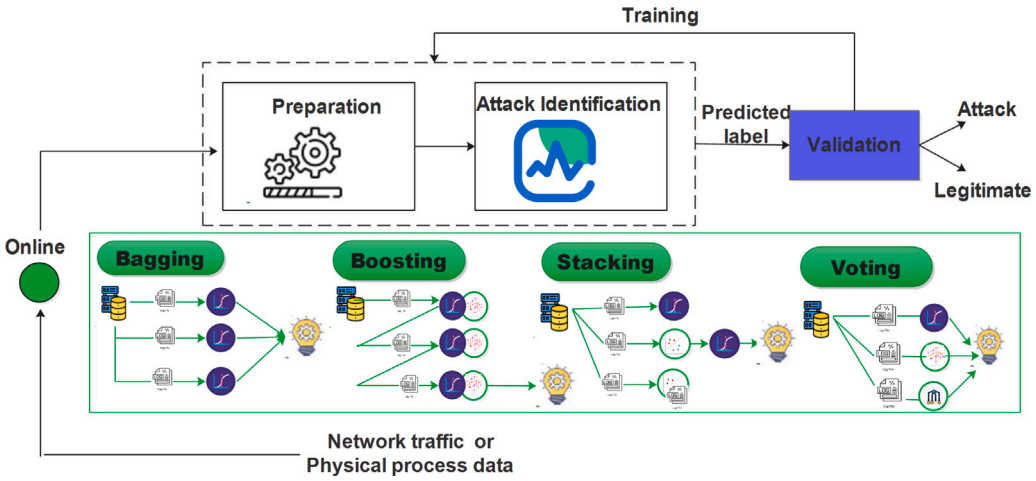


Fig. 2. The components of real-time detection module.

be considered. For example, a sample of features of industrial process (X_p) includes command address, sensor measurement, response write function code, response write, function code. Two DT models do The data preparation for the edge gateway's cyber and physical sides. This is important to facilitate effective and speedy attack detection on each side and take the appropriate response action.

Real time Identification Module:

In the identification module, we present a real-time ensemble-based detection model, shown in Fig. 2, to identify any malicious activity in the network. Multiple ensemble models, including bagging, boosting, stacking and voting, have been included in the framework and tested (see Algorithm 1. The main goal is to investigate the effectiveness of different approaches in differentiating legitimate from attack data collected from different sources. However, we could adopt the best performance ensemble model as the primary model to power our real-time identification module. The details of how each real-time or online ensemble approach works are described as follows:

Algorithm 1 Real time attack detection module using online ensemble machine learning algorithms

Data: Initial model parameters $(e.g., Features(f1, f2, f3, f_n), D(X, Y), X = [f1, f2, f3...f_n])$
Result: Output (Attack or Normal)
$Model = (bagging, boosting, stacking, voting)$
for *(i,* $(X, Y) \in D$*)* **do**
  Pred= Model.$predict\_one(X)$ **if** $(Pred \approx 0)$ **then**
    $Output \leftarrow Normal$
  else
    if $(Pred \approx 1)$ **then**
      $Output \leftarrow Attack$
    end
  end
  Model = Model.$learn\_one(X, Y)$
  Model= Model.update()
end

1. **Bagging approach:** In this approach, we consider a simple but effective classifier for binary data, i.e. logistic regression, as the primary classifier. The online logistic regression predicts the label of incoming observation (X) using sign function $(Y = (sin(W.X))$ which considers confidence in the choice of label and its interpreted as probability assigned to the labels, and then utilizes the logistic loss function (as described in Equation. (1)) to evaluate this probability and define if the classifier is correctly classified the incoming observation or not. Based on these results, the logistic regression updates its parameters (θ). In the bagging techniques, each classifier (h) is used to predict the label of the coming observation, and then the results are aggregated to produce the final results based on unweighted voting. Specifically, each based model's training set consists of (k) copies of each original training observation. For each received data (X, Y), we select (K) examples from a Poisson distribution with a mean of 1 for each base model. The base model is then updated using the online base model learning algorithm (similar to the offline (batch) learning) (Oza and Russell, 2001). The final prediction *(HBagging)*is the class that appears most frequently (i.e., *(Mode)* among all the base models $(h_1(X), h_2(X), \ldots h_m(X))$ (see Equation. (2)).

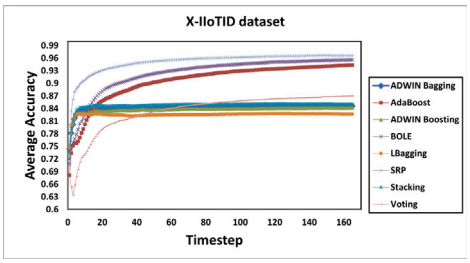
$$L\theta = \log(1 + exp{-}Y(w.X)) \tag{1}$$

$$H\,Bagging = Mode(h_1(X)h_2(X), \ldots, h_m(X)) \tag{2}$$

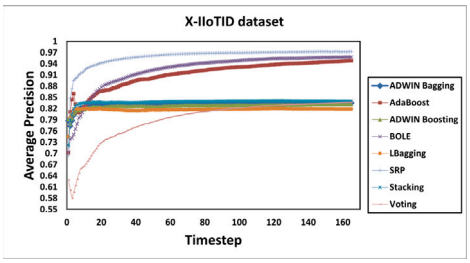
2. **Boosting approach:** This approach is similar to the bagging one. Still, the logistic regression or Hoeffding Tree classifier models (depending on the algorithm used) will learn in an adaptive way where the base model depends on the previous ones. The Hoeffding Tree classifier is an online decision tree learner which comes with a guarantee that the learned tree is asymptotically identical to that of the offline learner using large enough training observations. Hoeffding has the same model of learning where the first observation is used to create the root test node. The succeeding ones are passed down to the corresponding leaves, where each node is expanded if there is sufficient statistical evidence that optimal features exist. This decision is based on distribution-independent Hoeffding Bound (i.e., HBound) as defined in Equation. (3). Suppose that (X) is an independent observation of a random variable (r) with range(R), where (r) is a feature selection on a measure. If we calculate the mean of (r), Hoeffding Bound (i.e., HBound) states that the true mean of (r) is at least $(1 - \delta)$ where (δ) represents the degree of confidence in the expanding and splitting node.

$$H\,Bound = \sqrt{\frac{(R^2 \ln(1/\delta))}{(2X)}} \tag{3}$$

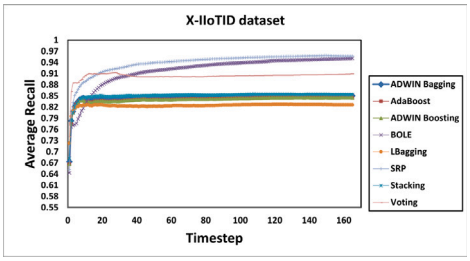
In classical boosting approach (as explained in Equation. (4)), a series of base models $(h_1, h_2, h_3, ..h_m$ by utilizing weighted training sets $(D_1, D_2, D_3, ..D_m)$, D=$(X,Y))$. In this process, the training observation misclassified by the previous model (h_{m-1}) are assigned half of the total weight when generating the subsequent model (h_m). In contrast, the correctly classified observations receive the remaining half of the weight. In real-time or online boosting, sampling is simulated using replacement through



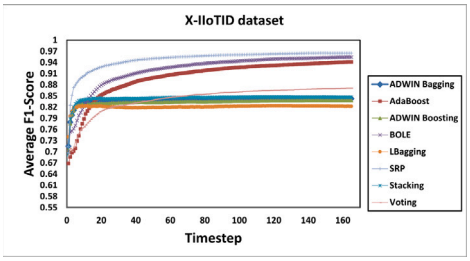
(a) Average accuracy over 160 steps where each is 1000 observations



(b) Average precision over 160 steps where each is 1000 observations



(c) Average precision over 160 steps where each is 1000 observations



(d) Average precision over 160 steps where each is 1000 observations

Fig. 3. The performance metrics for different real-time ensemble algorithms using X-IIoTID dataset.

the Poisson distribution. The parameters of Poisson distribution associated with training observations are adjusted based on the base model's classification. Specifically, if a base model misclassifies a training observation, the associated parameter is increased before presenting it to the next base model; otherwise, it is decreased (Oza and Russell, 2001).

$$Bosoting\_Model = HB(X) = h\_1(X) + \sigma h_2(X) + \sigma h_3(X) + \sigma h_m(X) \quad (4)$$

3. **Stacking approach:** In this approach (as described in Equation. (6), a set of classifiers $(h_1, h_2, h_3, ..h_m)$ are trained in parallel and then combined by training a meta-model to predict the label of training observation *(X)* based on the different model's predictions. In our approach, we use logistic regression and Passive-aggressive learning algorithms (Crammer et al., 2006). Passive-aggressive learning predicts the label of incoming observation and keeps learning if this observation is drawn from the same distribution, likely without a significant modification on their weights. However, if the incoming observation comes from a different distribution, the weights will slowly forget the previous ones and start to learn the new distribution. For simplicity, given a received observation *(X)* with label *(Y)* and the current weight vector *W*, the algorithm predicts the label using the sign function $(\sin(w, x))$, and the Hinge loss function (like a support vector machine) as defined in Equation. (5). If it predicts the observation label correctly, the value of loss function $(L(\theta)$ will be zero, and it will keep passively learning. At the same time, if it mispredicts the label, the value of $(L(\theta)$ will be one, and it will aggressively update its rules and weights, and this new update classifier should stay as close as to the previous one.

$$L(\theta) = Max(0, 1 - Y.f(X, \theta)) \quad (5)$$

$$HS(X) = Meta\_Model(h_1(X), h_2(X), h_m(X)) \quad (6)$$

4. **Voting approach:** In this approach, multiple models (classifiers) are trained independently, and their predictions are combined through a voting mechanism. The final prediction is based on the majority vote.

3.1.3. Action module:

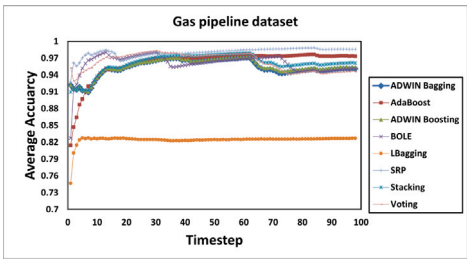
Based on the output of the identification module, the action module will take action to mitigate the attack. For example, if the attack is detected in the physical industrial processes, the DT will compare the received data from the physical processes with the expected ones. Then, the DT will react to mitigate the impact of this attack through this action module. The controller will handle the synchronization processes between the DT and the physical world and will be responsible for performing the attack mitigation actions. For instance, the controller will reduce the traffic and data selected from this infected device and replace it with legitimate data from the digital twin to ensure continuous system operations and physical processes.

4. Experimental evaluation

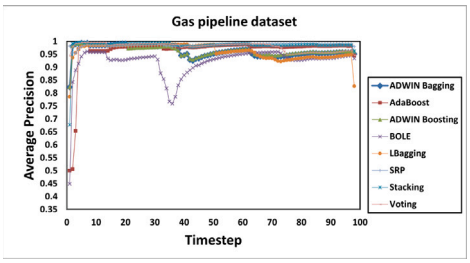
4.1. Real-time ensemble model evaluation

The Python library River was used to build online or real-time ensemble machine-learning models. Different ensemble models, including bagging, booting, stacking, and voting, test their capabilities in powering intrusion detection in the DT. These models consist of ADWIN Bagging, ADWIN Boosting, AdaBoost, Boosting Online Learning Ensemble (BOLE), Leveraging Bagging Ensemble (LBagging), Streaming Random Patches (SRP), Stacking, and Voting. The ADWIN Bagging and ADWIN Boosting include logistic regression with the ADWIN algorithm for dealing with any concept drift in the data (Gonçalves et al., 2014). The AdaBoost is built based on the Hoeffding Tree classifier, and the BOLE includes the Hoeffding Tree with a Drift Detection Method (DMM) to handle any concept drift in the receiving observations. LBagging is an improved version of bagging (increasing the re-sampling) using logistic regression with ADWIN to handle the concept drift. The SRP includes Hoeffding Tree in a bagging approach with random subspaces. The stacking consists of logistic regression and passive-aggressive learning algorithms, whereas the logistic regression acts as a meta-model. The voting uses logistic regression, Hoeffding Tree, and Gaussian Naive Bayes. We have used the default parameters of these algorithms using the River library because they are selected based on many experiments and are the most adequate for these ensemble machine learning models.

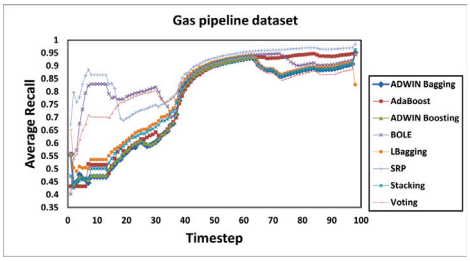
The datasets used in this experiment include X-IIoTID (Al-Hawawreh et al., 2021b) and gas pipeline dataset (Morris et al., 2015) as they represent the network traffic of edge cloud systems and the physical



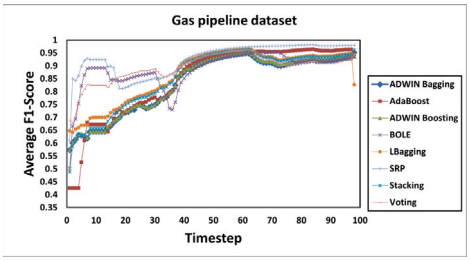
(a) Average precision over 100 steps where each is 1000 observations



(b) Average precision over 100 steps where each is 1000 observations



(c) Average recall over 100 steps where each is 1000 observations



(d) Average precision over 100 steps where each is 1000 observations

Fig. 4. The performance metrics for different real-time ensemble algorithms using gas pipeline dataset.

Table 1
The performance of online/real-time ensemble algorithms for the entire dataset.

Algorithm	X-IIoTID dataset				Gas pipeline dataset			
	Acc (%)	P (%)	R (%)	F1 (%)	Acc (%)	P (%)	R (%)	F1 (%)
ADWIN Bagging	84.62	83.51	85.08	84.29	95.12	95.89	90.70	93.22
AdaBoost	94.37	94.81	93.51	94.15	97.34	98.24	94.51	96.34
ADWIN Boosting	84.18	83.10	84.56	83.83	95.42	96.12	91.29	93.65
BOLE	95.62	95.62	95.09	95.46	95.22	94.68	92.24	93.45
LBagging	82.71	81.88	82.61	82.25	96.25	98.53	79.15	80.62
SRP	**96.61**	**97.25**	**95.72**	**96.48**	**98.55**	**99.01**	**97.04**	**98.02**
Stacking	85.05	84.03	85.39	84.70	96.17	98.66	90.86	94.6
Voting	87.01	83.68	90.94	87.16	94.62	96.49	88.67	92.42

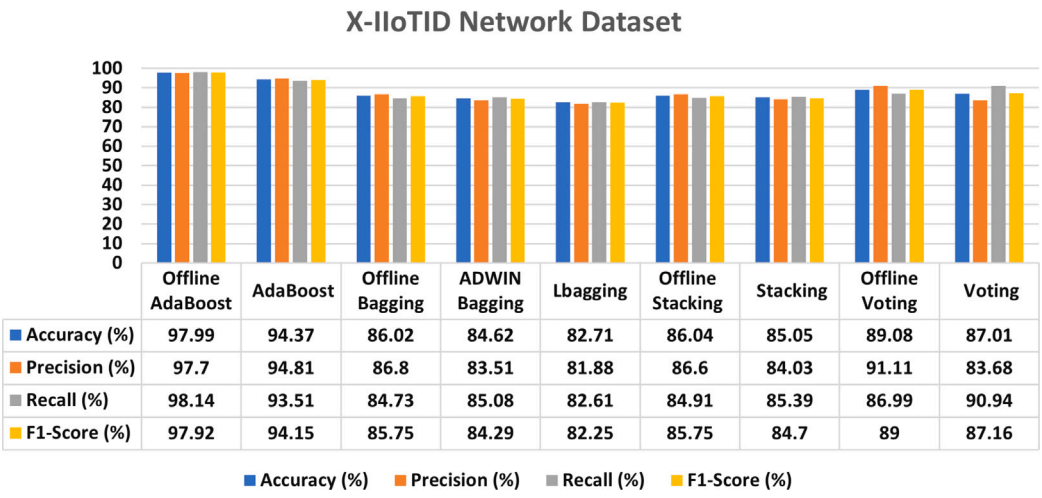
processes in the industrial environment. We have selected only a sample of data from X-IIoTID, which includes 27 features related to the network traffic only, including 84,582 observations for normal and 79585 observations for attacks. The gas pipeline dataset consists of 61156 observations for normal and 35863 observations for attacks, with 25 features for each. The experiments tested many performance metrics, including loss, accuracy (Acc), precision (P), recall (R), and F1-Score (F1).

We start the experiments by evaluating the effectiveness of selected online or real-time ensemble models using the X-IIoTID dataset, as shown in Fig. 4. Each model predicts the label of each receiving observation, which is then used to train the model. The average accuracy of each algorithm increases approximately with the increasing number of training observations (Fig. 4. a). It is evident that, in the initial timestep, all algorithms predicted labels for the first 1000 observations with an average accuracy exceeding 63%. However, the SRP and BOLE achieved the best average accuracy, reaching more than 96%. As shown in Figs. 3. b, Fig. 3. c, and Fig. 3.d, the SRP and BOLE achieved the best precision, recall, and F1 scores. Their performance continuously increases over timesteps to reach more than 96%. While these algorithms use different ensemble approaches (boosting for BOLE and bagging for SRP), they are built using the Hoeffding Tree, which is the reason for their high performance.

Fig. 4 shows the results using a gas pipeline dataset where all data observations are ordered based on time. Similar to the previous scenario, each algorithm first predicts the label of receiving observation, and then it is used in training. The proposed framework performs well in terms of all performance metrics. Fig. 4.a illustrates the average accuracy over 100 steps (where each step represents 1000 observations). All the used algorithms perform well, including the LBagging, which is less accurate than other algorithms. The LBagging has an unstable performance where its precision, recall, and F1-score (see Fig. 4.b, Fig. 4.c) fluctuate and drop at step 100. Still, it has a good performance metric range between 80% and 95%. Importantly, the SRP also achieved the best performance metrics using the gas pipeline dataset, similar to the X-IIoTID dataset. While the BOLE has high performance, it is unstable compared with the case of the X-IIoTID dataset. The AdaBoost algorithm performed better than BOLE using the gas pipeline dataset.

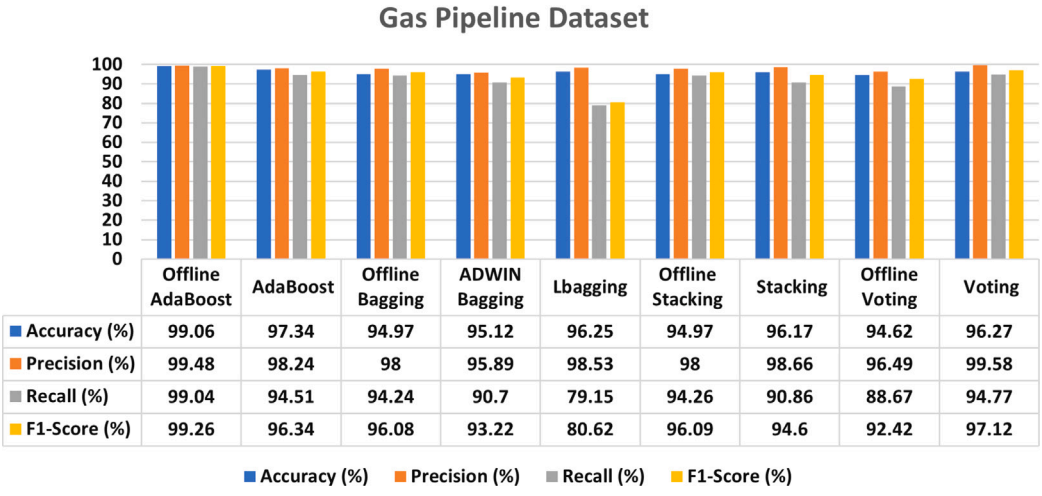
We calculated the total average accuracy, precision, recall, and F1-score for the fully used X-IIoTID and gas pipeline datasets. As described in Table 1, for X-IIoTID, the SRP achieved 96.61%, 97.25%, 95.72%, and 96.78% for accuracy, precision, precision, recall, and F1-score, respectively. Using the gas pipeline dataset, SRP also achieved 98.55%, 99.01%, 97.04%, and 98.02% for accuracy, precision, precision, recall, and F1-score. The BOLE comes second-highest using the X-IIoTID dataset, while the AdaBoost is second-highest regarding the gas pipeline dataset performance metrics. Stacking, voting, ADWIN bagging, and ADWIN Boosting perform better while using the gas pipeline dataset. Overall, these three algorithms, SRP, BOLE, and AdaBoost, perform best in identifying attacks. However, to power our identification model, we can use SRP as it achieves the best performance.

Upon detection of an attack, such as the lateral movement of a ransomware attack from an edge gateway to a physical device, the real-time detection module triggers an alarm to the controller. Subsequently, the controller mitigates the traffic originating from the edge gateway and substitutes it with traffic from the physical and cyber DT. Similarly, in the event of a false data injection attack detected in one of the sensors, the controller replaces the data received from this sensor with data from the physical DT.



	Offline AdaBoost	AdaBoost	Offline Bagging	ADWIN Bagging	Lbagging	Offline Stacking	Stacking	Offline Voting	Voting
Accuracy (%)	97.99	94.37	86.02	84.62	82.71	86.04	85.05	89.08	87.01
Precision (%)	97.7	94.81	86.8	83.51	81.88	86.6	84.03	91.11	83.68
Recall (%)	98.14	93.51	84.73	85.08	82.61	84.91	85.39	86.99	90.94
F1-Score (%)	97.92	94.15	85.75	84.29	82.25	85.75	84.7	89	87.16

Fig. 5. Comparison of real-time/online ensemble machine learning algorithms with offline ensemble algorithms using X-IIoTID network dataset.



	Offline AdaBoost	AdaBoost	Offline Bagging	ADWIN Bagging	Lbagging	Offline Stacking	Stacking	Offline Voting	Voting
Accuracy (%)	99.06	97.34	94.97	95.12	96.25	94.97	96.17	94.62	96.27
Precision (%)	99.48	98.24	98	95.89	98.53	98	98.66	96.49	99.58
Recall (%)	99.04	94.51	94.24	90.7	79.15	94.26	90.86	88.67	94.77
F1-Score (%)	99.26	96.34	96.08	93.22	80.62	96.09	94.6	92.42	97.12

Fig. 6. Comparison of real-time/online ensemble machine learning algorithms with offline ensemble algorithms using gas pipeline dataset.

4.2. Comparison with offline ensemble algorithms

We also investigated the results of real-time or online ensemble algorithms with offline ensemble algorithms (batch learning). For offline ensemble algorithms, we used 10 cross-validations to evaluate their performance in terms of all selected metrics. Cross-validation was employed to ensure that all observations in the dataset participated in both training and testing phases, mirroring the online or real-time scenario where each observation serves for both testing and training. Fig. 5 shows the results of the X-IIoTID dataset. The results show differences between outcomes obtained through offline algorithms and those from online/real-time algorithms. For instance, employing the entire dataset and employing 10 cross-validation with offline AdaBoost resulted in approximately a 2% increase in accuracy and precision, along with 4% and 3% improvements in recall and F1-score, respectively. However, offline bagging and stacking outperform their online/real-time counterparts in identifying attacks (recall) and F1-score. Online voting performs better in terms of all metrics than offline voting.

Fig. 6 illustrates the results of the gas pipeline dataset. The results of offline ensemble algorithms are better than the online/real-time algorithms. For example, the offline AdaBoost resulted in approximately a 3% increase in accuracy, precision and F1-score and approximately a 5% increase in recall value. Offline voting resulted in roughly a 7% rise in the value of precision. The online voting, stacking, LBagging, and ADWIN Bagging performed better in attack identification (i.e., recall) than offline ones for this dataset.

4.3. Pros and cons of our proposed framework

The proposed digital twin-driven secured edge-private cloud IIoT framework has several strengths. Firstly, it can identify the attack in its early stage because it is built using online machine learning, reducing the dwell time of threats and minimizing their impact on the system operations. Secondly, it provides automated mitigation and enhanced detection capabilities because of a digital twin. The digital twin represents a virtual replica of the physical and cyber sides of the edge gateway, which facilitates the detection of any abnormal behavior on any side and mitigation by replacing some data (if needed) from the virtual replica. Finally, our proposed framework demonstrated strong performance during evaluation using two heterogeneous datasets representing physical and cyber systems. Its versatility allows for application in various industrial systems, as specific system settings do not limit it.

Our proposed framework currently addresses only binary classes (normal and attack) and does not specify the type of attack. This limitation will be addressed by training the model using multiple classes. Additionally, the efficiency of digital twins in both cyber and physical aspects has not yet been evaluated. However, this can be resolved using simulated and real-time edge private cloud systems.

5. Conclusion

This paper proposes a real-time security framework for edge private cloud systems in an industrial environment that uses digital twin and

online ensemble machine learning algorithms to detect attacks and mitigate their effects. We explored various online ensemble machine-learning techniques and tested them using the X-IIoTID network and gas pipeline datasets as a specific use case. These online ensemble algorithms, such as SRP, BOLE, and AdaBoost, performed efficiently in identifying attacks compared with other techniques, such as stacking and voting. However, all the online ensemble algorithms demonstrated efficacy comparable to offline ensemble algorithms (which are extensively used in the literature). This demonstrates that offline learning does not reflect the real-time changes in data and does not handle the evolving attack behavior. At the same time, online models process one data point at a time and update the model over time, enhancing attack detection capabilities. It is evident that our proposed framework, using real-time detection and digital twin capabilities, is robust for securing edge private cloud systems in the IIoT environment. In future work, we will use explainability techniques to understand how these online and offline ensemble algorithms identify malicious behavior and test how well the proposed framework works in simulated and real-time edge private cloud systems that are part of gas pipeline systems.

CRediT authorship contribution statement

Muna Al-Hawawreh: Writing – original draft, Validation, Methodology, Formal analysis, Conceptualization. **M. Shamim Hossain:** Writing – review & editing, Supervision, Software, Resources, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Researchers Supporting Project number (RSP2024R32), King Saud University, Riyadh, Saudi Arabia.

References

Akbarian, F., Fitzgerald, E., Kihl, M., 2020. Intrusion detection in digital twins for industrial control systems. In: 2020 International Conference on Software, Telecommunications and Computer Networks. SoftCOM, IEEE, pp. 1–6.

Al-Hawawreh, M., Hossain, M.S., 2023. A privacy-aware framework for detecting cyber attacks on internet of medical things systems using data fusion and quantum deep learning. Inf. Fusion 101889.

Al-Hawawreh, M., Sitnikova, E., Aboutorab, N., 2021a. Asynchronous peer-to-peer federated capability-based targeted ransomware detection model for industrial iot. IEEE Access 9, 148738–148755.

Al-Hawawreh, M., Sitnikova, E., Aboutorab, N., 2021b. X-IIoTid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. IEEE Internet Things J. 9 (5), 3962–3977.

Alcaraz, C., Lopez, J., 2022. Digital twin: A comprehensive survey of security threats. IEEE Commun. Surv. Tutor. 24 (3), 1475–1503.

Alghamdi, R., Bellaiche, M., 2021. A deep intrusion detection system in lambda architecture based on edge cloud computing for IoT. In: 2021 4th International Conference on Artificial Intelligence and Big Data. ICAIBD, IEEE, pp. 561–566.

Arzani, B., Ciraci, S., Saroiu, S., Wolman, A., Stokes, J., Outhred, G., Diwu, L., 2020. {PrivateEye}: Scalable and {privacy-preserving} compromise detection in the cloud. In: 17th USENIX Symposium on Networked Systems Design and Implementation. NSDI 20, pp. 797–815.

Attaran, M., Celik, B.G., 2023. Digital twin: Benefits, use cases, challenges, and opportunities. Decis. Anal. J. 6, 100165.

Balta, E.C., Pease, M., Moyne, J., Barton, K., Tilbury, D.M., 2023. Digital twin-based cyber-attack detection framework for cyber-physical manufacturing systems. IEEE Trans. Autom. Sci. Eng..

Cao, K., Hu, S., Shi, Y., Colombo, A.W., Karnouskos, S., Li, X., 2021. A survey on edge and edge-cloud computing assisted cyber-physical systems. IEEE Trans. Ind. Inform. 17 (11), 7806–7819.

Chen, Y., Ding, Z., Wagner, D., 2023. Continuous learning for android malware detection. arXiv preprint arXiv:2302.04332.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive aggressive algorithms.

de Araujo-Filho, P.F., Kaddoum, G., Campelo, D.R., Santos, A.G., Macêdo, D., Zanchettin, C., 2020. Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment. IEEE Internet Things J. 8 (8), 6247–6256.

Feng, X., Wu, J., Wu, Y., Li, J., Yang, W., 2023. Blockchain and digital twin empowered trustworthy self-healing for edge-AI enabled industrial internet of things. Inform. Sci. 642, 119169.

Ferrag, M.A., Debbah, M., Al-Hawawreh, M., 2023. Generative ai for cyber threat-hunting in 6g-enabled iot networks. In: 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops. CCGridW, IEEE, pp. 16–25.

Gonçalves, Jr., P.M., de Carvalho Santos, S.G., Barros, R.S., Vieira, D.C., 2014. A comparative study on concept drift detectors. Expert Syst. Appl. 41 (18), 8144–8156.

Gumaste, S., Narayan, D., Shinde, S., Amit, K., 2020. Detection of ddos attacks in openstack-based private cloud using apache spark. J. Telecommun. Inf. Technol. (4), 62–71.

He, S., Du, C., Hossain, M.S., 2024a. 6G-enabled consumer electronics device intrusion detection with federated meta-learning and digital twins in a meta-verse environment. IEEE Trans. Consum. Electron. 70 (1), 3111–3119.

He, G., Li, C., Shu, Y., Luo, Y., 2024b. Fine-grained access control policy in blockchain-enabled edge computing. J. Netw. Comput. Appl. 221, 103706.

Hossain, M.S., 2017. Cloud-supported cyber–physical localization framework for patients monitoring. IEEE Syst. J. 11 (1), 118–127.

Kar, B., Lin, Y.-D., Lai, Y.-C., 2023. Cost optimization of omnidirectional offloading in two-tier cloud–edge federated systems. J. Netw. Comput. Appl. 215, 103630.

Li, X., Liu, H., Wang, W., Zheng, Y., Lv, H., Lv, Z., 2022. Big data analysis of the internet of things in the digital twins of smart city based on deep learning. Future Gener. Comput. Syst. 128, 167–177.

Morris, T.H., Thornton, Z., Turnipseed, I., 2015. Industrial control system simulation and data logging for intrusion detection system research. In: 7th annual southeastern cyber security summit. pp. 3–4.

Oza, N.C., Russell, S.J., 2001. Online bagging and boosting. In: International Workshop on Artificial Intelligence and Statistics. PMLR, pp. 229–236.

Sai, S., Prasad, M., Upadhyay, A., Chamola, V., Herencsar, N., 2024. Confluence of digital twins and metaverse for consumer electronics: Real world case studies. IEEE Trans. Consum. Electron..

Sai, S., Rastogi, A., Chamola, V., 2023. Digital twins for consumer electronics. IEEE Consum. Electron. Mag..

Salim, M.M., Comivi, A.K., Nurbek, T., Park, H., Park, J.H., 2022. A blockchain-enabled secure digital twin framework for early botnet detection in IIoT environment. Sensors 22 (16), 6133.

Singh, A., Chatterjee, K., Satapathy, S.C., 2022. An edge based hybrid intrusion detection framework for mobile edge computing. Complex Intell. Syst. 8 (5), 3719–3746.

Sonkoly, B., Haja, D., Németh, B., Szalay, M., Czentye, J., Szabó, R., Ullah, R., Kim, B.-S., Toka, L., 2020. Scalable edge cloud platforms for IoT services. J. Netw. Comput. Appl. 170, 102785.

Tärneberg, W., Skarin, P., Gehrmann, C., Kihl, M., 2021. Prototyping intrusion detection in an industrial cloud-native digital twin. In: 2021 22nd IEEE International Conference on Industrial Technology, Vol. 1. ICIT, IEEE, pp. 749–755.

Tian, Z., Luo, C., Qiu, J., Du, X., Guizani, M., 2019. A distributed deep learning system for web attack detection on edge devices. IEEE Trans. Ind. Inform. 16 (3), 1963–1971.

Xu, H., Wu, J., Pan, Q., Guan, X., Guizani, M., 2023. A survey on digital twin for industrial internet of things: Applications, technologies and tools. IEEE Commun. Surv. Tutor..

Yang, R., He, H., Xu, Y., Xin, B., Wang, Y., Qu, Y., Zhang, W., 2023. Efficient intrusion detection toward IoT networks using cloud–edge collaboration. Comput. Netw. 228, 109724.

Muna Al-Hawawreh is an Assistant Professor (Lecturer) at the School of Information Technology at Deakin University, Australia. She received her Ph.D. degree in Computer Science from UNSW Canberra in 2022. She also received bachelor's and master's degrees (First Class Honors) in computer science from Mutah University, Jordan. She is a passionate cyber security researcher with academic and industrial experience. Dr. Al-Hawawreh's research is multidisciplinary and focuses on cyber security and privacy-preserving in cyber environments like the industrial Internet of things, industrial control systems, cloud computing, cyber–physical systems, and smart satellites, with a focus on investigating, analyzing, and detecting current and future cyber-attacks (offensive and defensive research). She is also looking into using artificial intelligence applications for cybersecurity automation. She has a strong publication record and has published many peer-reviewed research papers in top-tier journals. Her contribution is recognized both

nationally and internationally through achieving various awards such as the first prize for high impact publication, UNSW Canberra, "Dr. KW Kong" best paper award from the publications of 2018–2020, ARC PGC Student Award—in recognition efforts to continue researching and contribute to the UNSW higher degree research community during the COVID-19, the Ria De Groot Prize for the best performance by female postgraduate student and Stephen Fester Prize for the most outstanding thesis on an information technology related topic by masters by research or Ph.D.

M. Shamim Hossain is currently a Professor at the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an Adjunct Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, ON, Canada. He received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, ON, Canada in 2009. His research interests include cloud networking, smart environment (smart city, smart health), AI, deep learning, edge computing, Internet of Things (IoT), multimedia for health care, and multimedia big data. He has authored and co-authored more than 355 publications including refereed journals, conference papers, books, and book chapters. Recently, he co-edited a book on "*Connected Health in Smart Cities*", published by Springer. He has served as the cochair, general chair, workshop chair, publication chair, and TPC in several IEEE and ACM conferences. He is the **chair of *IEEE Special Interest Group on Artificial Intelligence (AI) for Health*** with the IEEE ComSoc eHealth Technical Committee. He is currently the Organizing Co-Chair of the Special Sessions with IEEE I2MTC 2022. He serves as the Co-Chair of the 2ndIEEE GLOBECOM 2022 Workshop on Edge-AI and IoT for Connected Health. He is the Symposium Chair of Selected Areas in Communications (E-Health) with IEEE GLOBECOM 2024. He is theTrack Chair of the IEEE International Conference on Consumer Electronics (ICCE 2024). He is the Technical Program Co-Chair of ACM Multimedia 2023Currently, he is the Chair of the Saudi Arabia Section of the Instrumentation and Measurement Society Chapter. He was the recipient of several awards, including the Best Conference Paper Award, the **2016 *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM) Nicolas D. Georganas Best Paper Award,*** *the* ***2019 King Saud University Scientific Excellence Award (Research Quality****), and the Research in Excellence Award from the College of Computer and Information Sciences (CCIS), King Saud University* (**3 times in a row**). He is on the editorial board of *theIEEE Transactions on Instrumentation and Measurement (TIM), IEEE Transactions on Multimedia (TMM), ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *IEEE Multimedia, IEEE Network, IEEE WirelessCommunications, Journal*of Network and Computer Applications (Elsevier), International Journal of Multimedia Tools and Applications (Springer), and Games for Health Journal.He has served as a Lead Guest Editor for more than two dozen of Special Issues (SIs) including *ACMTransactions on Multimedia Computing, Communications, and Applications (TOMM)*, *ACM Transactions on Internet Technology*,IEEE Transactions on Consumer Electronics,*IEEE Communications Magazine, IEEE Network, IEEE Transactions on Information Technology in Biomedicine* (currently *JBHI*), *IEEE Transactions on Cloud Computing*, International Journal of Multimedia Tools and Applications (*Springer*), Cluster Computing (*Springer*), Future Generation Computer Systems (*Elsevier*), Sensors (*MDPI*), and International Journal of Distributed Sensor Networks. He is a senior member of the IEEE and a Distinguished Member of the ACM. He is an IEEE Distinguished Lecturer (DL). He is the Highly Cited Researcher in the field of Computer Science—2022 and 2023 (Web of Science™).