International Journal of Research and Applied Technology

5(1)(2024) 34-64

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Deep Learning Security Schemes in IIoT: A Review

Shavan Askar[1*], Diana Hussein[1], Media Ibrahim[1], Marwan Aziz Mohammed [2]

[1]Information System Engineering Department, Erbil Technical Engineering College, Erbil Polytechnic University, Erbil, Iraq

[2]Department of Computer science, college of engineering, knowledge university, Erbil, Iraq

Email: *shavan.askar@epu.edu.iq

Abstract. The Industrial Internet of Things (IIoT) is a fast-growing technology that might digitize and connect numerous industries for substantial economic prospects and global GDP growth. By the fourth industrial revolution, Industrial Internet of Things (IIoT) platforms create massive, dynamic, and inharmonious data from interconnected devices and sensors. Security and data analysis are complicated by such large diverse data. As IIoT increases, cyberattacks become more diversified and complicated, making anomaly detection algorithms less successful. IIoT is utilized in manufacturing, logistics, transportation, oil and gas, mining, metallurgy, energy utilities, and aviation. IIoT offers significant potential for industrial application development, however cyberattacks and higher security requirements are possible. The enormous volume of data produced by IoT devices demands advanced data analysis and processing technologies like deep learning. Smart assembly, smart manufacturing, efficient networking, and accident detection and prevention are possible with DL algorithms in the Industrial Internet of Things (IIoT). These many applications inspired this article on DL's IIoT potential.

Keywords: Industrial Internet of Things (IIoT); cybersecurity; intrusion detection system and deep learning (DL).

1. Introduction

The Industrial Internet of Things (IIoT) is a vast network of smart devices that benefit cognitive computing in infrastructures and businesses, from manufacturing to services. Modern technology automates manufacturing and industrial processes in Industry 4.0 [1]. Integration of the Internet of Things (IoT) and large-scale machine-to-machine connections improves automation, communications, self-monitoring, and intelligent machines that can identify errors without human intervention. IIoT platforms require precise data collection, processing, and safe transmission due to vast sensors and devices working together. Despite the benefits and prospects of the IIoT revolution, hackers actively try to steal data or cause damage to IIoT and industrial devices [2]. More IIoT applications mean more security threats and cyberattacks. In the recent decade, various IIoT cyberattacks compromised software and hardware, such as pumps and sensors [3],[4]. Stuxnet [5] was one of the initial 2010 IIoT assaults on security. Ukraine's power grid outage changed the customary security system [6] or Colonial Pipeline attack in the USA in 2021 are other instances [7]. These instances show that infrastructure and IIoT cyberattacks are expanding, posing security dangers to systems. Edge devices acting as IIoT nodes may disrupt industrial output (e.g., transmitters and sensors [8]) run in anomalous programs. Internal anomalies like abnormal traffic or irregular frequencies produce some of these anomalous behaviors, while external anomalies like attackers' destructive conduct cause others [9], [10], [11], [12]. To monitor and safeguard IIoT structures from assaults and discover unusual data, reliable anomaly prediction and detection techniques must be used. As shown in Fig. 1, an IIoT platform has four layers: physical (sensors and instruments for sensing and collecting data), transmission (sending or receiving data), storage and processing, and application (using data to provide a service or production). Due to the huge volume of data sensing (gathering and/or receiving), transferring, processing (storage), and usage complexity in IIoT, measuring location, sampling frequency, and transmission method and rate differ from traditional platforms [13], [14], [15]. These changes and other IIoT characteristics affect the form and quality of raw data, which may be high-dimensional, large-scale, time-dependent, dynamic, or imbalanced [13] ,[16]. The IIoT platform's storage and processing layer senses, processes, and collects time-series data from distributed edge devices (nodes) to study their activity [17]. Therefore, time-dependency (time-series type) is a key property of IIoT data that aids data analysis and forecasting [18]. Data at a given time may be linked to a lengthy period of time or a previous point in time, depending on the short- and long-term. Such relationships require aberrant data. Thus, abnormal data at one time may be linked to past data. This dependence feature can better identify and anticipate IIoT dataset outliers.

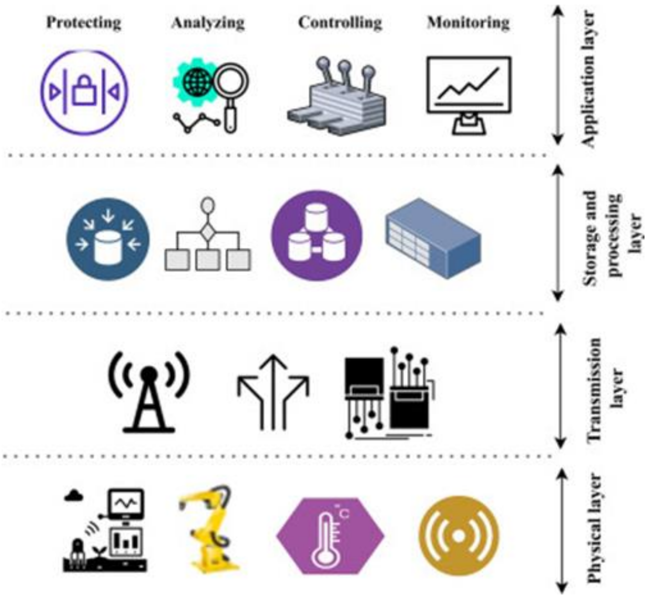

Figure 1. IIoT platform overview

Two main types of anomaly and attack detection methods are anomaly-based and signature-based [2],[19], [20], [21]. Signature-based approaches can detect previously identified abnormalities (or those induced by humans) but not new ones [2],[22]. Raising the volume of abnormal data and previously reported anomalies complicates anomaly detection and slows response time [23]. Furthermore, Refs introduced certain methods [24],[25] concentrating on log-based cyber threat hunting and federated threat hunting techniques [26]. These new solutions combat cyberattacks and security issues. IIoT systems have seen a significant surge in networked devices in the fourth industrial revolution. Intelligent big data processing is crucial to IIoT smart application performance. IIoT networks need intelligent information processing frameworks for massive data analysis. In this case, artificial intelligence (AI) and DL can help IIoT systems produce relevant outcomes from massive data [27],[28],[29]. DL approaches allow systems to learn from experience. Data properties and learning algorithm performance determine DL solution efficiency and efficacy [30]. The right DL algorithm for an application can be difficult to choose. Therefore, knowing how different DL algorithms operate and how they're used in practical applications like smart homes, smart cities, cybersecurity services, smart healthcare, smart transportation, sustainable agriculture, business enterprises, and others is crucial [31]. Analyzing developing and cutting-edge IoT/IIoT research can reveal the importance of DL. In Fig. 3, DL outperforms standard algorithms for big data sets.

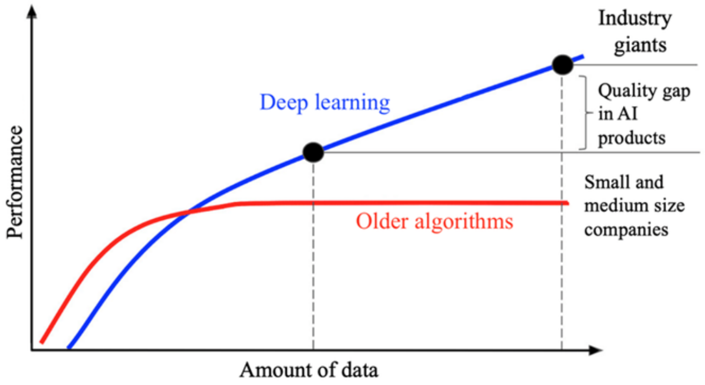

Figure 2. Comparing of IIoT DL and classical algorithms

1.1. Intrusion Detection System (IDS)

IDS monitors harmful attacks in interconnected networks or nodes. It defends the node or network from attacks [32]. An attack that damages sensor nodes is malevolent. IDS systems can be hardware or software. IDS can identify harmful network activities and known attacks from human behaviors. It analyzes node and network activity to detect intrusions and alerts users. It's an alert or network observer. System damage is prevented by alarm generating before illegal attacks. Internal and external attacks can be detected by the IDS. Malicious network nodes generate IAs. Third-party EAs are obtained elsewhere. IDS monitors network traffic to identify authorized and illegal users. IDS involves monitoring, investigation, and alert. The monitoring component tracks traffic, resources, and network patterns. Study is key to determining intrusions based on algorithm. When intruders are detected, alert module alarms. Three types of IDSs are listed below:

1. **Host-based IDS System (HIDS):** A single or multiple host systems' design, operating system, and application files are estimated by HIDS. This system collects internal data to the operating system computer, monitors user activity, and systems program execution. It had better recovery, detailed logging, fewer false positives, and unknown attack prediction.

Unintelligible data, total coverage, indirect data, outsiders, and host influence are drawbacks.

2. **Anomaly-based IDS System (AIDS):** This is event-based intrusion detection. Events are analyzed to identify harmful conduct. First, it details the attack's routine. An intrusion occurs when actions differ from typical.
3. **Network-based IDS System (NIDS):** This system monitors network traffic through routers, switches, and network interface devices to detect intrusions. Network streams like internet packets capture most data. All LAN assaults can be detected by NIDS, which host-based IDS cannot. Easy implementation, affordability, detection range, forensics integrity, and Every effort is NIDS advantages. Wire speed failure, direct attack susceptibility, indecipherable packets, and coverage problems are drawbacks.

1.2. Requirement for IDS in Internet of Things (IoT) Networks

IoT is an emerging technology that identifies physical objects that may exchange information. The items converse without human involvement. IoT is a smart network that exchanges data with recognized protocols over the internet. Thus, the user can access anything anytime, anyplace. It communicates and collaborates with objects to create new services and applications using unique addressing mechanisms. Smart cities, residences, environment, health monitoring, and water are its applications. IoT delivers many services for daily living based on its reliable and available actions, but it requires multi-class integrity, privacy, and verification solutions. The IoT network should be secure, and sensor data should be uploaded encrypted. In the IoT network, secure communication is crucial. Among the many IoT challenges, security is crucial since devices can be accessed from anywhere via unauthorized networks [33]. Without security analysis, important data can be attacked at any time. Thus, security vulnerabilities must be identified from these angles:

1. **Confidentiality:** The aggressor can easily intercept the transmission from origin to destinations to leak user-sensitive data and modify it. Thus, secure data transformation matters most.
2. **Integrity:** The receiving device should receive transmit data unchanged. Integrity ensures that unauthorised intruders did not modify transmitted data.
3. **Availability:** Resources should be available as needed. Intruders can overload resources to disable access. Malicious attacks can impair this accessibility.
4. **Authenticity:** It verifies identity. Users can identify others they communicate with. Validation is possible via verification. Thus, unauthorized attacks cannot interact.
5. **Non-repudiation:** Increases transmitter and recipient inability to reject information. It verifies data origin and integrity.

1.3. Requirement for IDS in Internet of Things (IoT) Networks

IIoT is an innovative approach to smart manufacturing eco-systems using IoT for management of industrial processes. IIoT quickly grows the following sectors and services: Healthcare systems use IoT devices to track, sense, and monitor machines, patients, and drugs. IoT devices are used in agriculture for farm security, plant irrigation, and product storage management [34]. The supply chains industries depend on transportation and logistics [35]. IoT devices track vehicle movements in this field by determining its location. It also determines product supply time. In the energy sector, IIoT manages grid supply, billing, and leakage. IoT devices manage warning systems, sense crisis signals, follow underground miners, and monitor shipments in the mining industry [36]. The automation industry's strength defines ICS, which includes SCADA networks and PLCs. Most cyberattacks target industrial automated systems as Stuxnet, German steel mill blast furnace, Shamoon, Mirai, etc. Many hacks target industrial companies worldwide. IoT devices have many weaknesses for cybercriminals to exploit industrial processes. Traditional networks have stable defenses. To defend industrial systems against intrusions, a strong intrusion detection technique is needed. Deep learning-based IIoT intrusion detection systems are described next.

2. Review of Literature

An Anomaly Detection System (ADS) packet capture and decoding engine monitors network traffic and detects unusual activities, improving security management [37]. Security monitoring and abnormal activity identification are its key tasks. Any deviation from these patterns can be detected by the ADS as potential intrusions, including known and unknown threats [38], by creating patterns from the regular data [38]. During semi-supervised feature selection, Coelho et al. [36] suggested measuring label and data cluster similarity using a homogeneity metric. They found that cluster information can help assess feature relevance and pick features when labelled data is scarce. Regarding the 42 elements that comprise the entire UNSWNB15 dataset, Primartha and Tama [39] investigated the effectiveness of detection systems for intrusions (IDSs). They assessed performance metrics' accuracy and false alarm rate using 10-fold cross-validation. This experiment used NSL-KDD, UNSW-NB15, and GPRS datasets. The study compared the suggested model to the multi-layered perception (MLP), Decision Tree, and NB-Tree classifiers. The results revealed that the cross-validation model and Random Forest classifier with particular parameter values worked. The [40] strategy selected informational components specific to each assault category rather than generic elements for all attacks. Testing with the CICIDS2017 dataset showed that the suggested strategy accurately detected intrusions. Dahiya et al. [41] They suggested using Apache Spark to develop an intrusion detection system. They reduced features using LDA and CCA. The Bayes naïve, REP Tree, Random Tree, Random Forest, Random Committee, and bagging methods are prominent categorization algorithms. A Convolutional Neural Network (CNN) was used in [42] to create a model for classifying malware. A dataset of 9,339 samples from 25 different malware groups were used in the investigation, which had a remarkable accuracy rate of 94.5%. Similar to this, in [30], a deep CNN that used color image visualization to detect online malware threats was constructed. They found that cybersecurity threat categorization had improved. Researchers in [43] proposed the Random Coefficient Selecting and Mean Modification Method (RCSMMA)-based approach. Our system handled modern cyberattacks successfully. In [44], the authors discussed the potential applications of smart cities for malware attacks and the privacy and security issues that occur when designing smart city apps. Using multivariate tuples, a reliable steering and monitoring system was demonstrated by [44] to mitigate global sensor network adversaries. This protocol protected the sensor network against malware threats to increase security and integrity.

2.1. The Role of DL in IIoT Security

Industrial and commercial businesses can easily obtain important data from sensors, equipment, devices, instruments, and control systems via the IIoT. These devices might be at industries or in remote regions (agricultural, mining, oil and gas). New IIoT entry points are added to achieve high connectivity on all levels of the pyramid. This highly networked design creates enormous attack surfaces and exposes IIoT systems to major cybersecurity concerns that must be considered. In fact, the sector has had several major cybersecurity issues. For example, the closure of a Saudi petrochemical factory was caused by TRISIS malware. Experts believe the attack was meant to murder workers by disabling the safety system and disrupting the petrochemical process. The attack killed no one, but the site lost millions due to the sudden shutdown [45]. Internet-connected webcams were used to cyberattack the Baku-Tbilisi-Ceyhan (BTC) gas pipeline, shut down alarms, and over pressurize it, causing a huge explosion [46], [47]. Internet-connected mobile cards with weak authentication allowed hackers to access a Bowman Avenue Dam human-machine interface [48].

These occurrences show that IIoT security and privacy are important and should be researched. Indeed, the industry has collaborated to identify IIoT system dangers and vulnerabilities and develop standards and best practices to mitigate them. The Industrial Internet Security Framework (IISF) [28], the security protocols described by the Agency for Cybersecurity of the European Union (ENISA) [49], and similar efforts by National Institute of Standards and Technology (NIST) [50] and IEEE [49] are some promising steps towards securing the IIoT. IIoT security is increasingly employing deep-learning technologies. Deep learning is typically used to detect intruders in IoT and IIoT infrastructures. The main methods are briefly reviewed below. IoT malicious traffic was detected using deep, fully-connected neural networks (DNN) [51]. The seminal work in [52] proposes using both CNNsss and LSTMs to learn both the low-level spatial features and the high-level temporal features of network traffic. Learned traits distinguish well from harmful network data. A combination of CNNs and RNNs is also proposed in [53] help safeguard systems from multiple threats via host-based intrusion detection. A vast body of literature suggests autoencoders are preferred for intrusion detection [54], [55], [56]. Deep-belief networks have also been used for intrusion detection [57]. Denial-of-service attacks and malware have also been solved by CNNs and deep reinforcement learning [58], [59],[60]. IIoT security challenges and solutions can be found in intriguing and extensive surveys in [61][62][63][64]. Over the last few years, deep-learning algorithms have gained popularity in IIoT cybersecurity research. The proposed approaches are computationally complex, which conflicts with the IIoT's constrained nature, where many devices may have limited battery capacity, memory, and processing power. The issue in adopting such methods is designing lightweight, maybe distributed or decentralized DL algorithms.

2.2. Industrial Internet of Things Reference Architecture (IIoT)

IIoT's seamless device connectivity has revolutionized modern technology during the past decade. Mobile apps, devices (smartwatches, tablets/iPads, laptops, etc.), and more send important data to IoT networks in this age of Internet access. These IoT/IIoT systems have unique architectures with layers and components that serve specific purposes [65],[66]. Fig. 3 shows a detailed IIoT 7-layer design.

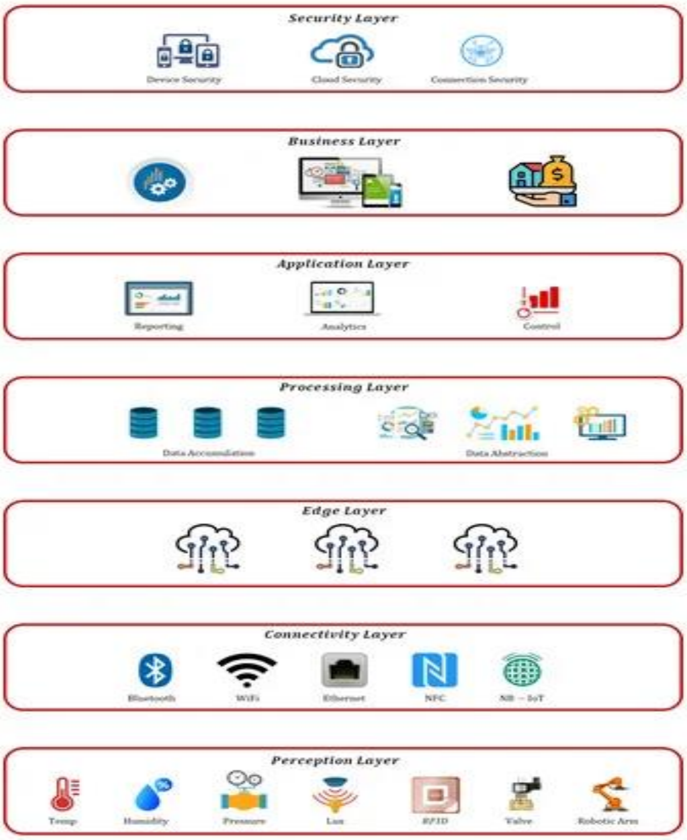

Figure 3. Industrial Internet of Things reference architecture

2.2.1 Perception Layer

Environmental data is collected and preprocessed by this "physical" layer. It digitizes analog data to make it compatible with other system layers[66]. Actuators and sensors make up this layer's major constituents.

1. **Sensors:** These little devices can detect environmental changes and gather useful data. Sensors typically have limited memory and processing power. Modern sensors can acquire environmental cues more accurately. The most widely used sensors in numerous industries measure temperature, humidity, air pressure, weight, acceleration, position, and others.
2. **Actuators:** Electromechanical devices usually transform electrical signals into physical actions. In industry, linear and rotary actuators are most common. Linear actuators convert electrical signals into linear motions for position adjustment. Meanwhile, rotary actuators convert electricity into rotation. These are used to control conveyor belt positions.

2.2.2 Connectivity Layer

This layer connects perception and edge layers using modern communication methods [67]. This layer allows two communication methods. TCP or UDP/IP stacks are used for direct communication in the first manner. Smart gateways connect LANs and WANs in another way. This layer uses several complex communication technologies and protocols.

1. **WIFI:** It's the most adaptable and widely utilized communication system. WiFi modems are ideal for personal and professional use, enabling LAN-WAN connections.
2. **Ethernet:** This outdated technology allows LAN or WAN devices to interact via a specified protocol. Ethernet lets network wires like optical fiber to copper and inversely communicate.
3. **Bluetooth:** This wireless protocol is commonly used in personal area networks for short-distance information sharing.
4. **NFC:** NFC wirelessly connects smart devices for secure short-range communication. The typical NFC communication range is 10 cm.
5. **LPWAN:** For long-distance communication, a class of radio technologies known as Low-Power Wide-Area Networks (LPWAN) is employed. Top LPWAN technologies include Nwave, Sigfox, and LoRa. Smaller data packets are typically sent over longer distances with LPWANs as opposed to other wireless technologies like Bluetooth and Wi-Fi.
6. **ZigBee:** This device is specifically made for IEEE 802.15.4-compliant sensor networks by the Zigbee alliance. ISA-100.11. a and Wireless HART are the two most often utilized data transmission protocols for this communication standard. These protocols specify the physical layers and Media Access Control (MAC) needed to manage multiple devices at slow data speeds.
7. **LTE-M:** One of the top LPWA network technologies for Internet of Things applications is Long Term Evolution for Machine. Through radio modules, it is utilized to connect items like Internet of Things actuators, sensors, and other industrial equipment.
8. **NB-IoT:** This low-power wide-area (LPWA) technology is standards-based and supports a large range of smart devices and applications. The power consumption, spectrum efficiency, and system capacity of smart devices are all enhanced by NB-IoT.

2.2.3 Edge Layer

Latency often hinders IoT network expansion early on. Edge computing speeds IoT network growth, which may solve this problem. It helps the system process and analyze data near the source. Edge computing is already typical for 5G mobile networks, enabling increased system connectivity with less latency. All processes at the edge boost IoT network performance [68].

2.2.4 Processing Layer

This layer stores and analyzes edge layer data [69]. These operations are all performed by IoT systems and include two main stages. This layer stores and analyzes edge layer data [69]. These operations are all performed by IoT systems and include two main stages. Data Abstraction: Consumer apps can gain insights from data after collection and preparation. It involves integrating data from many sources, reconciling formats, and aggregating data in one place. These methods are used combined to improve smart device interoperability. Common processing layer protocols are listed below:

1. **Transmission Control Protocol (TCP):** It breaks up big data sets into packets and resends and reassembles them for host-to-host communication.
2. **User Datagram Protocol (UDP):** Process-to-process communication is facilitated by this IP-based protocol. UDP transfers data quicker than TCP, making it the preferred protocol for applications that are mission-critical.

3. **Internet Protocol (IP):** Many IoT protocols utilize IPv4, while newer ones use IPv6. The latest IP update directs Internet traffic and locates network devices

2.2.5 Processing Layer

Software evaluates data to offer promising answers to major business challenges in this tier. Versions of this layer show that thousands of IIoT apps differ in design complexity and functions. Each employs different OS and technologies. Software-controlled device monitoring, business intelligence (BI) services, AI-based analysis solutions, and mobile apps for simple interactions are popular uses. Recent approaches include building the application layer on top of IoT/IIoT frameworks that give software-based architectures with data visualization, mining, and analytics tools [70]. Some popular applications layer protocols are listed below.

1. **Advanced Message Queuing Protocol (AMQP):** It lets messaging middleware talk. It allows many systems and applications to communicate, leading in large-scale standardized communications.
2. **Constrained Application Protocol (CoAP):** Constrained-bandwidth network protocol for machine-to-machine communication between low-capacity devices. CoAP transfers documents via UDP.
3. **Data Distribution Service (DDS):** A versatile peer-to-peer protocol that can connect high-performance networks and tiny devices. DDS streamlines deployment, improves reliability, and reduces complexity.
4. **Message Queue Telemetry Transport (MQTT):** A low-bandwidth messaging protocol for machine-to-machine and distant communications. The publisher-subscriber MQTT protocol is ideal for small devices with constrained battery life and bandwidth.

2.2.6 Business Layer

If beneficial for business planning and strategy, IoT data are useful. Every organization needs to extract meaningful data for goal-oriented tasks. Businesses set future goals using previous and present data. Modern companies use clever data analysis to improve decision-making [71]. Industries have turned to software and business analytics to boost performance and profitability.

2.2.7 Security Layer

Due to rising problems, IIoT architecture must include security. Hacking, denial of service, malicious software, and data breaches are IIoT infrastructure's biggest issues [72]. Three primary tasks are carried out by this layer, which are as follows.

2.3 Security Issues

Many IoT security issues arise from limited storage, power, and processing capabilities. The effect of default user credentials allowing unauthorized access to IIoT devices remains unfixable [73]. Although manufacturers are aware of this defect, little is being done to address it, leaving users with a technological challenge. Ironically, IoT devices, as proven by Li et al. [74], About 48% of users are unaware that their gadgets could be exploited for cyberattacks, and 40% never update software. To reduce security threats, many believe manufacturers and software developers should upgrade. As indicated, IIoT architecture has seven layers, Layers have vulnerabilities. Due to their outdoor location, IIoT devices in the perception layer are subject to physical attacks, which aim to steal or tamper with sensor data. Man-in-the-middle and DoS attacks occur in middleware and networks. Privilege escalation and SQL injections are examples of traditional computer attacks on businesses and applications. Each layer can be attacked in Fig. 4. The lack of IoT communication standards is the key concern [75], Since there are no norms, security is difficult. Thus, general security solutions can be difficult to create. As determined by S. Garg et al. [76],Hacker's target software and web servers. In addition to connecting devices, servers store a lot of sensitive data. Hackers may benefit from inserting malicious code that infects connected devices. Hardware security features include hardware-supported software isolation and the hardware root of trust [77]. Using hardware to construct isolated units and securely store cryptographic keys is not new and is similar to traditional IT studies. Smart cities' IIoT systems' computational and energy constraints make hardware security difficult. Due to their inaccurate real-time clocks, some IIoT devices may make particular networks unfeasible. These variables can compromise higher-layer security [78].

Software presents different issues. Current OSes provide process isolation, so one process cannot interfere with another. Memory management unit (MMU) supports isolation. In IIoT, there is no centralized OS overseeing all activities, hence maintaining process isolation without an MMU is difficult [79]. Even though process isolation is well-known, IIoT devices with additional resources require novel ways to provide resource-constrained OS isolation. A common issue is access restriction. System resources are protected by OS access control from untrusted code. Two are typical concepts for access security. First, give part of the code an OS-only identity. Second, supply a token that only the process may use. Access control systems are difficult to build [73]. Access control, while still relevant to IoT platforms, presents new usability issues in system design. An fascinating task is to create an IoT access control system that uses our natural understanding about physical objects, while the majority of earlier systems use files, processes, and virtual objects [80].

IoT security also requires authentication. For IoT devices, services, and platforms, passwords are the most common authentication method. However, weak passwords have permitted enormous botnet DoS attacks, causing alarm [81],[82]. A comprehensive and well-deployed security method in multiple computing sectors detects rogue network devices. Tuning anomaly detectors to emit fewer false positives and negatives requires obtaining relevant information from them [83]. The heterogeneity of networked devices makes this challenge difficult. Since most of them perform diverse duties, network traces are complicated, making "bad" behaviors hard to identify. IIoT devices aim for simpler network dynamics, which makes behavior models easier to anticipate and reduces anomaly detector mistakes [84].

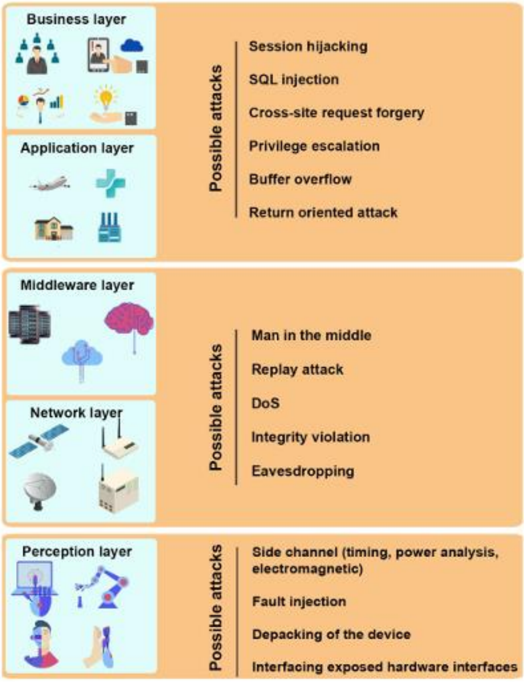

Figure 4. An illustration of potential assaults on each tier of an IIoT architecture.

3. Related Work

Many studies explore IIoT cyber threat hunting methodologies because to researchers' interest in its security [85],[86]. The ensemble method's anomaly detection performance is better than other machine learning approaches, and its use has increased in past studies. For instance, Hasan et al. [87] have evaluated machine learning algorithms for IIoT anomaly detection. They evaluated ML models using precision, accuracy, f1-score, and recall. For ANN, RF, and DT, the system achieved 99.4% accuracy. The RF model outperforms other methods in f1-score and recall. In Ref. [88], Jabbar and his colleague suggested an ensemble technique using the Average One-Dependence Estimator (AODE) and RF model to categorize network data as threat or normal. The proposed technique has an accuracy of 90.51% and a False Alarm Rate (FAR) of 0.14 using Kyoto data. Moreover, AL-Hawawreh et al. [89] have developed an ensemble deep learning model to detect IIoT traffic anomalies and malicious behavior. They employed TCP/IP data packets to train and validate a feed-forward and AE architecture. The model achieves 92.4% and 98.6% accuracy on the UNSW-NB15 and NSL-KDD datasets. Ref. shows an ensemble tree model. [90] to forecast IIoT turbofan engine maintenance. Using the C-MAPSS dataset, the Gradient Boosted Tree (GBT) technique achieves 93.91% accuracy, surpassing the Random Forest (RF) model at 91.78%. RF computes faster and performs better than GBT. In addition, Ref. introduces a multi-stage and edge device-based IIoT ensemble learning pruning pipeline. [91]. First, the pruned model is used to construct an ensemble model, then integer quantization is clustered, and lastly the best prototype cluster is chosen. To assess model efficacy, CIFAR-10 and CIFAR-100 datasets are used. The results show that the approach provides alternative prediction models. Gu et al. [92] IoT traffic data and key characteristics have been our attention. They offer a reinforcement attack detection learning model. In reinforcement learning, attack patterns are automatically learned and identified. Results demonstrate that dynamically adjusting the proposed model's feature thresholds can increase attack detection by 98.5%. Several ensemble model-based IDS studies exist. Kurniawan et al. [93] are developing Synthetic Minority Oversampling Technique. The experimental test uses the NSL-KDD dataset with accuracy as a performance factor. The technique has a 97% detection rate, 97.02% model accuracy, and 0.16% false alarm rate, according to the study. In Ref. [94] Abdel-Basset and colleagues have concentrated on federated threat-hunting in industrial cyber-physical systems and developed a model with 92.84% accuracy and 91.61% f1-score. Peng Gao et al. [26] proposed Threat Raptor, a computer system threat hunting system using OSCTI, and their report indicates sufficient accuracy and efficiency for real threat hunting. Liu et al. [95] present a deep IIoT model that detects sequential data anomalies. An FL framework is used to train the model on decentralized edge devices. Outliers are accurately detected using the Attention Mechanism-based Convolutional Neural Network- LSM model. Next, Top-k and compression-based algorithms improved communication efficiency. Implementing the proposed approach on four datasets showed that the framework can find abnormalities quickly and accurately. Communication overhead is reduced by 50% compared to non-gradient compression schemes. Table 2 summarizes recent relevant works.

Table 1. Recent IIoT cyberattack detection investigations.

Ref.	Year	ML model	Datasets	Target
[87]	2019	DT, RF, SVM, ANN	DS2OS	reach a high degree of precision
[88]	2019	AOED and RF	Kyoto data	Obtain good accuracy & decrease FAR
[89]	2018	Auto-encoder	NSL-KDD, KDD CUP 99	Using feed-forward and AE for high accuracy
[90]	2019	Gradient Boosted Trees	C-MAPSS	RF model uses GBT technique for increased accuracy and faster calculating time.
[91]	2020	Prune2Edge	CIAFR-10, CIFAR100	Propose new IIoT ensemble gaining knowledge of pruning pipelines using multi-phase and edge devices.
[92]	2020	Reinforcement learning	IoT attack	Increasing assault detection with reinforcement learning
[93]	2020	SMOTE	NSL-KDD	Promotes SMOTE for high gains
[95]	2020	AMCNN-LSTM	Power demand, Space shuttle, ECG, and Engine	Makes suggestions a deep model as AMCNN-LSTM, doing detect anomalies accurately and timely
[24]	2021	Federated Deep learning models	ToN IoT, LITNET-2020	Novel federated deep learning model for cyber threat hunting
[26]	2021	THREATRAPTOR	DARPA TC	Proposes THREATRAPTOR is a system that helps identify threats in computer systems.

4. DL Techniques

4.1. Deep Learning for the IIoT

Deep Machine Learning (DML), or just DL, is crucial to IIoT because of its versatility and broad implementation in practically all sectors. We will discuss some of the main themes here. DL works for IIoT security in Fig. 5.

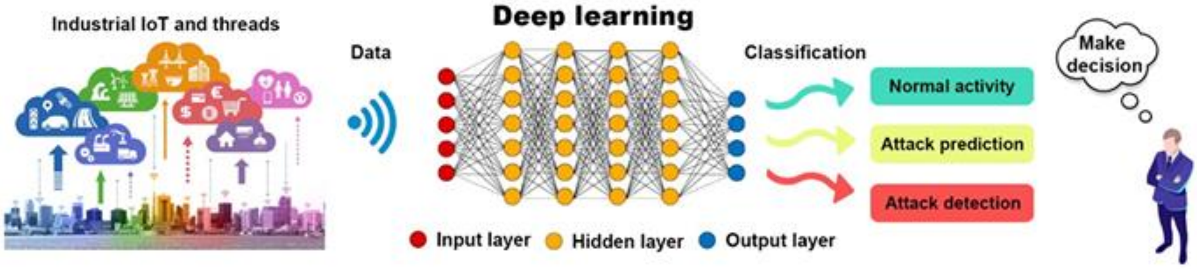

Figure 5. An illustration of DL working principle for IIoT Security

Intelligent manufacturing in the IIoT has many benefits. Creating intelligent manufacturing and production processes can be useful [96]. New industrialists use IIoT to boost productivity and profits. In IoT-enabled sectors, sensor and smart device data helps smarten production [96]. Therefore, modern firms must use intelligent data analysis methodologies. One of the most powerful AI algorithms is DL. Through multi-layer information processing, DL approaches in smart industries can optimize smart production. Due to its inherent learning, pattern recognition, and smart decision-making, DL techniques are helpful. DL's major advantage over ML is automatic feature learning. This option eliminates the requirement for a feature learning method. [96]. In smart industries, DL approaches can do the above analyses well. Next, we cover some popular IIoT DL-based methods.

4.1.1 Deep Feedforward Neural Networks

The most basic deep neural network (DNN) advances node connections. Fig. 6 shows the DFNN architecture. DNN's multiple hidden layers can model complicated nonlinear relations better than shallow networks. This design is popular in all technical domains due to its simplicity and robust training procedure [97]. For DFNN training, the most popular gradient descent algorithm is preferred. This approach initializes weights randomly and minimizes error via gradient descent. The entire learning process requires successive forward and backward propagation [98]. Forward propagation uses numerous hidden layers to process input to the output layer and compare the computed output to the desired output. For weight adjustment, the backward method calculates network parameter error rate of change. This will continue until the neural network produces the desired result.

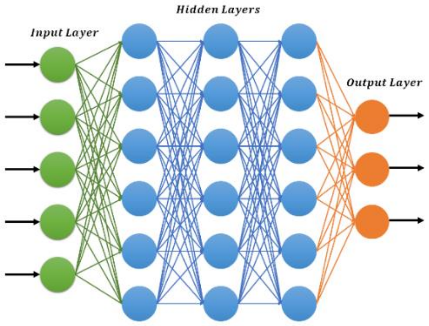

Figure 6. A deep feedforward neural network architecture in its common form.

Let xi be the neural network input and fi be layer i's activation function. For layer i, the output can be calculated as

$xi+1=fi(wixi+bi)\ xi+1=fiwixi+bi$ (1)

Here, xi+1 becomes the input for the next layer, wi and bi are the essential parameters that connect the layer i with prior layer. The settings are adjusted during the backward procedure, as illustrated below.

$wnew=w-\eta\partial E\partial Wwnew=w-\eta\partial E\partial W$ (2)

$bnew=b-\eta\partial E\partial bbnew=b-\eta\partial E\partial b$ (3)

Here, w new and b new are The updated w and b parameters. E represents the cost function, while η is the learning rate. The cost function of the DL model is determined by the desired job, such as classification or regression

4.1.2 Restricted Boltzmann Machines (RBM)

Also called stochastic neural networks, RBM. Its ability to learn the input probability distribution supervised and unsupervised makes this DL technique popular. Harmonium, invented by Paul Smolensky in 1986, was popularized by Hinton in 2002 with novel training methods. [99]. Since then, RBM has been used for Prediction, representation learning, and dimensionality reduction. RMB-based deep belief network training was popular. The Netflix dataset's performance makes RBMs popular for collaborative filtering [100]. RBM extends the Boltzmann Machine by restricting unit intra-layer connections. An undirected graphical model with visible and hidden layers forms a bipartite graph. Recent studies have introduced several RBMs with improved learning algorithms [101]. These varieties include conditional, temporal, convolutional, factored, and recurrent RBMs. Different nodes, such as Gaussian, Bernoulli, etc., can handle data properties. RBM nodes process information for stochastic decisions. General RBM architecture is shown in Fig. 7.

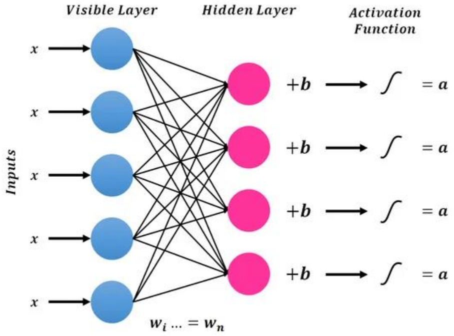

Figure 7. RBM architecture in general.

The Gibbs distribution can be used to characterize the joint probability distribution of a typical RMB $p(v,\ h)$=1ze−$E(v,h)p(v,h)$=1ze−$E(v,h)$
Here energy function $E\ (v,\ h)E(v,h)$ can be described as
$E(v,h)$=−∑i=1n∑j=1$mwijhjvi$−∑j=1$mbjvi$−∑i=1$ncihiE(v,h)$=−∑i=1n∑j=1$mwijhjvi$−∑j=1$mbjvi$ −∑i=1$ncihi$ (4)

Here m and n represent the number of visible and hidden units. $hjhj$ and $vjvj$ are the states of the hidden unit i and visible unit j respectively. $bjbj$ and $cjcj$ describes real-valued biases corresponding to the jth$j$th and ith$i$th units, respectively. $wijwij$ are the weights that connect visible units with hidden units. Z is the normalizing constant that makes sub-probability distributions 1. RBM training algorithm Contrastive Divergence was proposed by Hinton. This RMB training technique maximizes training sample probability. Training stabilizes the model by minimizing energy by updating its parameters, as shown in Equations (5)– (7).
Δwij=ϵ(⟨$vihj$⟩data−⟨$vihj$⟩model) Δwij=$\epsilon vihj$data−$vihj$model (5)
Δbi=ϵ(⟨vi⟩data−⟨vi⟩model) Δbi=ϵvidata−vimodel (6)
ΔCi=ϵ(⟨hj⟩data−⟨hj⟩model) ΔCi=ϵhjdata−hjmodel (7)

Here, $\epsilon\epsilon$ indicates learning rate, ⟨.⟩data⟨.⟩data, and ⟨.⟩model⟨.⟩model represent expected data and model values, respectively.

4.1.3 Deep Belief Networks (DBN)

DBNs have multiple latent variable layers. The hidden properties of input observations are indicated by these binary variables. The undirected link between the uppermost two layers makes DBN with one layer an RBM [102]. The remaining DBN connections have input-layer-directed graphs. A DBN model generates samples top-down [103]. Gibbs sampling is used to sample the top layer RMB. After then, visible units do top-down ancestral sampling. A general DBN model is shown in Fig. 8.

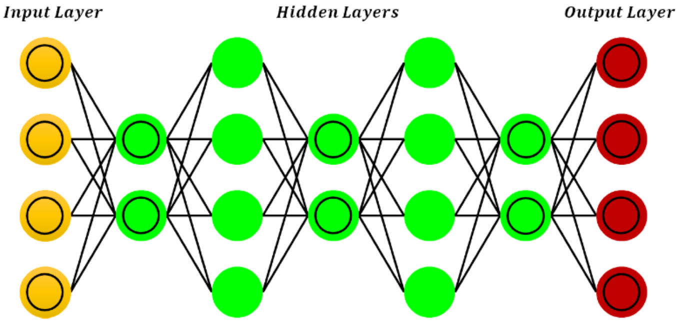

Figure 8. General Deep Belief Network architecture.

The latent variable model's explaining away effect makes DBN inference uncontrollable. Hin proposed a fast and efficient DBN training approach using RBM stacking: the lowest level of RBM learns the input data distribution at the start [104]. Next, RBM calculates the higher-order correlation between the preceding layer's concealed units. Each concealed layer undergoes the same process. Next, RBM calculates the higher-order correlation between the preceding layer's concealed units. Each concealed layer undergoes the same process. The visible joint distribution is modeled by a DBN with Z hidden layers. layer v and hidden layer $hzhz$, here z=1,2, 3,…,Zz=1,2,3,…,Z as described in the following.
$p(v,h1,\ldots,hz)$=$p(v|h1)\left(\prod z=1z-2p(hz|hz+1)\right) p(hz-1,hz)pv,h1,\ldots,hz=pv|h1\prod z=$ $1z-2phz|hz+1phz-1,hz$ (8)

As the first effectively trained deep architecture, Hinton's method launched contemporary DL. Logarithmic probability of training data can be considerably boosted by adding NN layers. DBN classifiers are used in computer and phone recognition. In speech recognition, DBN pre-trains DNN, deep convolutional neural network, and others.

4.1.4 Autoencoders (AE)

Unsupervised learning with autoencoders trains neural networks to ignore noise and represent input more efficiently. Fig. 9 shows AE, a 3-layer neural network. While The hidden layer typically has fewer neurons than the input and output layers, which have the same number of units. The buried layer compacts input data. RBM employs specific distributions, but AE uses deterministic distributions [105]. AE training mainly uses backpropagation. This training involves encoding and decoding. The model encodes input into hidden representations using weight metrics in the first stage. The model uses weight metrics to rebuild identical data from a model that is encoded during decoding. Encoding and decoding can be formally explained in Equations (9 and 10).

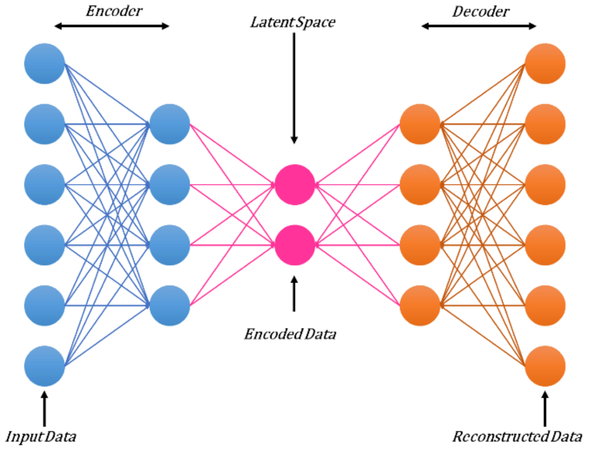

Figure 9. General Autoencoder architecture

During encoding.

$y'=f(wX+b)y'=f(wX+b)$ (9)

X is an input vector, f is an activation function, w and b are tuning parameters, and y is the hidden representation.

The decoding steps

$X'=f(w'y'+c)X'=fw'y'+c$ (10)

X represents the output layer's reconstructed input, w' represents the transpose of w, and c represents the output layer's bias value.

$wnew=w-\eta\partial E\partial Wwnew=w-\eta\partial E\partial W$ (11)

$bnew=b-\eta\partial E\partial bbnew=b-\eta\partial E\partial b$ (12)

The revised parameters after the current iteration are *wnewwnew* and *bnewbnew*. E is output layer reconstruction error. A deep autoencoder (DAE) has many hidden AE layers. Multiple layers make AE training tough. [106]. Each DAE layer can be trained as simple AE to overcome this issue. The DAE has been used in speech recognition, picture retrieval, document encoding for speedier retrieval, and more. Researchers were drawn to AEs' non-generative and non-probabilistic properties.

4.1.5 Convolutional Neural Network (CNN)

CNNs are visual-inspired neural networks. LeCun proposed CNNs in 1998 [107] became common in DL frameworks when Krizhevsky et al.[108]. ILSVRC-2012 winner with AlexNet architecture. This astonishing discovery began a new AI trend as data scientists saw CNN and its variants' strong classification skills. CNN algorithms excelled in human recognition systems in various applications. Fig. 10 shows CNN's basic architecture. It has numerous convolutional and pooling layers and a final fully linked layer. The convolutional layer extracts key characteristics from the input image using pixel spatial relationships. Pooling layers reduce feature map dimensionality while preserving details about features. Lastly, a completely linked layer links the neural network. Using the output layer to achieve the desired outcome. CNNs are great for latent spatial image descriptor extraction. Common picture qualities include color, contours, edge, textures, strokes, gradient, and orientation. CNNs separate images based on these attributes and represent them in layers. Computer vision applications like image identification, classification, segmentation, and super-resolution reconstruction favored CNNs. Multiple CNN frameworks have been presented to meet real-world application needs and high accuracy. Modern architectures like R-CNN and YOLO are popular. Naive CNNs use a massive amount region proposal to find an object in an image, making them computationally expensive. R-CNN-based selective search selects a region of interest (ROI) to address this issue. Redmon et al. [109] YOLO was first proposed in 2016. It's faster than R-CNN without sacrificing performance. One CNN glimpse at an object teaches it its generic image representation. However, this method has spatial restrictions when detecting tiny objects. Other CNN frameworks include AlexNet, LeNet, VGGNet, ResNet, GoogleNet, ZFNet, and others [110]. The CNN frameworks have greatly influenced Vision research enabled by AI for potential applications.

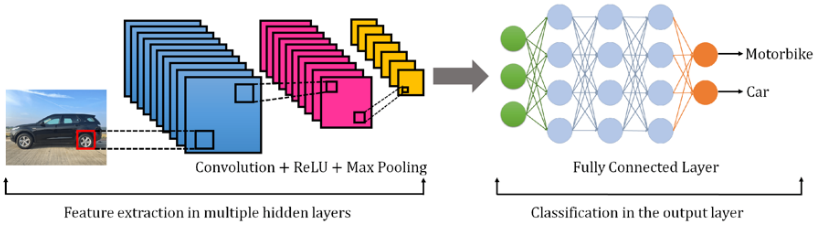
Motorbike
Car
Convolution + ReLU + Max Pooling
Fully Connected Layer
Feature extraction in multiple hidden layers
Classification in the output layer

Figure 10. General CNN architecture.

4.1.6 Recurrent Neural Network (RNN)

The connections between nodes in an RNN form a directed graph following a time series. [111]. Temporal dynamics are enabled by this characteristic. All inputs and outputs in CNNs are independent. Previous data may be needed to predict the following phrase or statement. Thus, past data must be remembered. Hidden layers help RNN solve this problem. The hidden state that remembers sequence information is the most important RNN feature [111]. RNN's memory stores all calculated data. It uses the same parameters for inputs and calculated data. To produce an appropriate output, it employs the same parameters for all inputs and jobs on all input or hidden layers. This feature simplifies parameters compared to other neural networks. The connections between nodes in an RNN create a directed graph following a time series [112]. Temporal dynamics are enabled by this characteristic. In typical neural networks, inputs and outputs are independent. Previous data may be needed to predict the following phrase or statement. Thus, past data must be remembered. Hidden layers help RNN solve this problem. The concealed state that retains sequence data is RNN's most important feature [113]. RNN's memory stores all calculated data. To produce an appropriate output, it employs the same parameters for all inputs and jobs on all input or hidden layers. This feature simplifies parameters compared to other neural networks. RNN architecture is presented in Fig. 11. Here is a small illustration of RNN's operation. Consider a DNN with 1 input, Three concealed, one output layer. Each layer will have unique weights and biases. Let's suppose hidden layer weights and biases are $w1, w2, w3w1, w2, w3$ and $b1, b2, b3b1, b2, b3$. These layers do not remember the prior output; therefore, they are independent. RNN turns independent activations into dependent ones by giving each layer the same biases and weights. It reduces complexity by memorizing previous outputs as input to the next hidden layer. These 3 layers can be combined into a single recurrent layer with identical weights and biases for all hidden layers. The following equation calculates the current state.

$ht=f\ (ht-1, xt)\ ht=fht-1, xt$ (13)

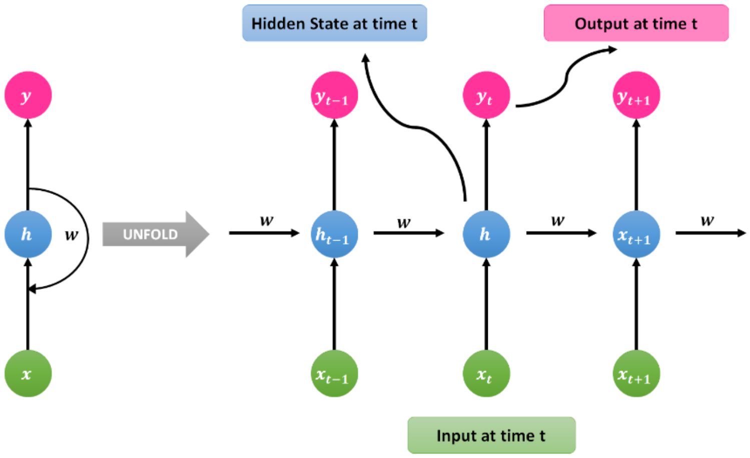

Figure 11. RNN architecture in general.

Here $htht$ represents the current state, $ht-1ht-1$ is the previous state and $xtxt$ is the neural network input.

The following expression applies a hyperbolic tangent (tanh) activation function.

$ht=\tanh(whhht-1+wxhxt)\ ht=\tanh whhht-1+wxhxt$ (14)

Here $whhwhh$ is Weight of current neuron and input neuron weight ($wxhwxh$). You can calculate output with this equation.

$yt=whyhtyt=whyht$ (15)

4.1.7 Generative Adversarial Networks (GAN)

GANs use two neural networks to produce artificial data that can be used in place of real data. GANs are commonly utilized in speech, picture, and video generation [114]. GAN was introduced by Ian Goodfellow at Montreal University in 2014. Facebook AI research director Yann LeCun named adversarial training the most promising ML topic in the previous decade. Because they can replicate any data distribution, GANs have many applications [115]. We can teach these networks to generate worlds like ours in speech, music, image, video, etc. Essentially, these networks are robot artists with astounding output. Deepfakes—GAN-generated fake media—is another technology. Gans have two models. The first

A generator model generates data similar to the required data. The generator resembles a human art forger who forges art. Discriminator is the second model. The model verifies if input data is from the original dataset or forged. Discriminators are like art experts who evaluate works for authenticity. GAN operation is shown in Fig. 12. GAN is easy to grasp utilizing universal approximators like ANN. Generators can be modeled using a neural network G (n, $\theta 1$). Gn, $\theta 1$. Its major task is mapping noise variables n to data space x. Another option is to simulate the discriminator using a second neural network ($D(x, \theta 2)$ Dx, $\theta 2$). It returns a 0–1 data authenticity probability. (0,1). In both cases, $\theta i \theta i$ represents neural network weights.

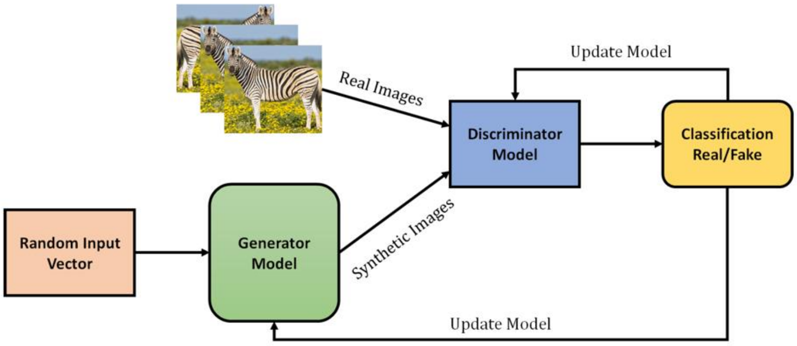

Figure 12. General Generative Adversarial Network architecture.

Thus, the discriminator trains to appropriately categorize input data, updating neural network weights to optimize the chance that any genuine data input x belongs to a real dataset. The generator is trained to produce realistic data. This also means that generator weights are tweaked to maximize the likelihood that each bogus image is classified as being from the original dataset. The generator and discriminator will hit a plateau after numerous training cycles. The generator provides realistic and artificial data, as well as the discriminator cannot distinguish between them. Generator and discriminator compete to optimize opposite loss functions during training [116]. The generator strives to maximize its likelihood of real output, whereas the discriminator tries to minimize it.

5. KEY DIFFERENCES IN DL-IIOT

We have highlighted the necessity of DL in numerous businesses, but implementing DL and getting usable and trustworthy results is difficult. It requires domain knowledge, statistical analysis, addressing issues and obtaining data-insight. Here are some of the biggest issues of DL-assisted IIoT networks

5.1. Complexity

A major challenge with DL models is complexity, which requires extra effort to fix [117],[118]. Complex models and big industrial datasets cause DL performance issues including the time-consuming training phase and the computationally intensive inference phase. Because IIoT devices are compact and mobile, they must offload intensive computations due to limited processing power, memory, and battery life. It might not always be feasible for a variety of reasons. Critical data should not be sent online for security. High transfer latency from cloud offloading of data and computations might not be appropriate for many time-critical real-time applications. Thus, effective DL algorithms that analyze data locally on IoT devices are crucial [119]. DL integration in IIoT sensors and devices has been attempted in recent years. Most of these work developed lightweight DL inference engines [120-123] , Libraries, frameworks, operating systems, etc., that can run on low-resource hardware such ARM Cortex-M MCUs and RISC-V-based PULP [124],[28]. DL models must be efficient, precise, and resource-efficient, as these efforts are still young. To efficiently learn industrial data features, a tensor train deep-compression (TTDC) model can address complexity [125]. This speeds up the DL model by compressing numerous parameters. DL's high computational complexity can be addressed by splitting models across IIoT and cloud systems [126] or to trim DL models for memory and computational savings [127]. How to split or prune models optimally depends on the model cut and accuracy needs, thus it's still unknown. Finally, accelerator hardware like DianNao, DaDianNao, and Eyeriss [128,129][130-139] are designed to maximize CNN and DNN performance, enabling native IoT DL applications. Because of the particular hardware and application, this strategy is not cost-effective and limited to a few crucial circumstances. Additionally, these strategies may not function as well with RNNs or GANs

5.2. Data Availability

industrial applications, overfitting affects model accuracy and efficiency due to a shortage of valuable training samples. The availability of valuable data raises many concerns. IIoT sensors and gadgets automatically record data. The same sensors may not capture the same data on the same instruments or devices due to non-uniform calibration, device age, or environment. Sensor data may also be lost due due to transmission-and-acquisition noise or inadequate connection. IIoT devices may not collect enough data for predictive maintenance. For instance, failure data is scarce, making it hard for DL algorithms to understand [140]. Thus, failure data may be useless. Thus, contextual data must be collected systematically, such as by considering calibration and aging [141].

5.3. Algorithm Selection

Many IIoT DL algorithms are popular and widely available [142]. These algorithms can function in any generic environment, however those for industrial applications should be chosen based on industry needs. Knowing which DL algorithm works best in a business is crucial [143]. Choosing the wrong DL algorithm might result in junk output, wasting time, effort, and money. Choosing a DL algorithm for an industrial setting is difficult. The IIoT has tagged data, making supervised learning algorithms better. If data is sparse and unlabeled, unsupervised DL methods are best. Therefore, the initial step is to identify DL algorithms, whether supervised or unsupervised suit the industrial setup better. Comparing the computational cost, complexity, and dependability of DL algorithms before applying them to an IIoT application is also necessary. CNN, RNN, and DBN DL algorithms require enormous training data and expensive hardware. Due of these issues, novel DL algorithms including deep transfer learning, deep reinforcement learning, and hybrid DL algorithms must be researched.

5.4. Data Selection

A DL algorithm must be intimately associated with training data to avoid “garbage in, garbage out” [144]. Avoiding selective bias and choosing data that accurately portrays the industrial process is crucial. Data selection relies on industry, data source availability, cost, and convenience. Each application has its own ideal data space, therefore reducing unnecessary data may be more efficient. By filtering data by specified criteria, analysts can compress it and save on computational, storage, and bandwidth expenses. DL models often learn on unprocessed inputs through multiple hidden layers to attain the desired performance at the cost of long training [145]. Thus, standard DL-based IIoT data-selection techniques are costly. Thus, core-set selection and active learning can be researched to help DL algorithms choose the most useful training examples.

5.5. Data Preprocessing

Converting raw data with Incomplete data, anomalies, and insignificant entries in a statistical form and DL algorithms can understand is the next critical step after picking the data [146], [147]. This step involves parsing, cleaning, and preparing data, which may include transforming raw data into numbers, scaling features to prevent dominance, and eliminating or replacing missing entries. IIoT architectures should contain automatic processes for iteratively cleansing data to enable smarter DL [148-153]. Data preprocessing increases data quality in DL-based IIoT networks to extract useful insights.

5.6. Data Labeling

We know that supervised DL algorithms are the easiest and best for IIoT. Unsupervised DL algorithms are harder to develop and may require many failed iterations and a protracted training procedure [154]. However, supervised DL algorithm data labeling is difficult and cannot be outsourced for complex tasks. Medical picture labeling and classification for diagnosis requires domain experts like doctors. Specialized medical specialists consider picture classification time-consuming. [155]. Data labeling is being reduced. Important progress toward this goal is SenseGAN [156], which considerably reduces labeled data using GANs. Another GAN-based labeling method is data augmentation [157]. Other DL problems in smart industrial applications include managing model and data versions, recreating models, etc. DL evolves, therefore its learning capacities change [158,159]. Should the group discover that the most recent models, features, and dataset are not properly described, integrating them into a DL setup might be a headache.

6. Conclusion

In order to link physical things to the internet for a variety of future industrial applications, IIoT is the most crucial component. Because IoT devices may turn items from application areas into internet hosts, their adoption has expanded dramatically over the previous ten years. However, because of security flaws, user privacy and security pose a significant difficulty. IoT security needs to be developed and looked into as a result. Deep learning-based IIoT networks and systems employ the IDS security mechanism. Next, we went over the various applications of DL-based IIoT, such as asset tracking, smart meters, smart grids, remote healthcare monitoring, predictive maintenance, and the mining, transportation, telecom, and agricultural sectors. We also discussed some of the difficulties with DL-based IIoT, such as the use of DL in IIOT security methods. Lastly, we used survey articles to determine the future paths of DL-based IIoT. The survey addressed the main flaws in the previous research and solved these issues by adding more data. The majority of surveys now in use don't provide a thorough explanation of typical IIoT design. This paper outlined the essential enabling technologies and protocols for the IIoT as well as its comprehensive seven-layer architecture. This survey covered mathematical underpinnings, reference designs, and the theories of popular DL algorithms. The absence of software and hardware implementation platform consideration in the current studies is one of their main flaws. A thorough explanation of hardware and software deployment frameworks is provided in the context of DL and IIoT in order to overcome this problem. Several possible use cases of DL technologies for the IIoT are described in order to assess the efficacy of DL for IIoT. In conclusion, this survey concludes by outlining the primary obstacles present in current DL-based IIoT systems.

References

[1] Yazdinejad, A., Dehghantanha, A., Parizi, R. M., Hammoudeh, M., Karimipour, H., & Srivastava, G. (2022). Block hunter: Federated learning for cyber threat hunting in blockchain-based iiot networks. *IEEE Transactions on Industrial Informatics*, *18*(11), 8356-8366.

[2] Abdel-Basset, M., Chang, V., Hawash, H., Chakrabortty, R. K., & Ryan, M. (2020). Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment. *IEEE Transactions on Industrial Informatics*, *17*(11), 7704-7715.

[3] Al-Abassi, A., Karimipour, H., Dehghantanha, A., & Parizi, R. M. (2020). An ensemble deep learning-based cyber-attack detection in industrial control system. *Ieee Access*, *8*, 83965-83973.

[4] Yazdinejad, A., Parizi, R. M., Bohlooli, A., Dehghantanha, A., & Choo, K. K. R. (2020). A high-performance framework for a network programmable packet processor using P4 and FPGA. *Journal of Network and Computer Applications*, *156*, 102564.

[5] Zhang, F., Kodituwakku, H. A. D. E., Hines, J. W., & Coble, J. (2019). Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. *IEEE Transactions on Industrial Informatics*, *15*(7), 4362-4369.

[6] Alert, D. 2016. Cyber-attack against ukrainian critical infrastructure. *Cybersecurity Infrastruct. Secur. Agency, Washington, DC, USA, Tech. Rep. ICS Alert (IR-ALERT-H-16-056-01)*.

[7] Hobbs, A. (2021). *The colonial pipeline hack: Exposing vulnerabilities in us cybersecurity*. SAGE Publications: SAGE Business Cases Originals.

[8] Yazdinejad, A., Parizi, R. M., Srivastava, G., Dehghantanha, A., & Choo, K. K. R. (2019, December). Energy efficient decentralized authentication in internet of underwater things using blockchain. In *2019 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.

[9] HaddadPajouh, H., Dehghantanha, A., Parizi, R. M., Aledhari, M., & Karimipour, H. (2021). A survey on internet of things security: Requirements, challenges, and solutions. *Internet of Things*, *14*, 100129.

[10] Wu, J., Yao, L., Liu, B., Ding, Z., & Zhang, L. (2020). Combining OC-SVMs with LSTM for detecting anomalies in telemetry data with irregular intervals. *IEEE Access*, 8, 106648-106659.

[11] Ingre, B., Yadav, A., & Soni, A. K. (2018). Decision tree based intrusion detection system for NSL-KDD dataset. In *Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 2 2* (pp. 207-218). Springer International Publishing.

[12] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Karimipour, H., Srivastava, G., & Aledhari, M. (2020). Enabling drones in the internet of things with decentralized blockchain-based security. *IEEE Internet of Things Journal*, *8*(8), 6406-6415.

[13] Wu, D., Jiang, Z., Xie, X., Wei, X., Yu, W., & Li, R. 2019. LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT. *IEEE Transactions on Industrial Informatics*, *16*(8), 5244-5253.

[14] Yazdinejad, A., Srivastava, G., Parizi, R. M., Dehghantanha, A., Choo, K. K. R., & Aledhari, M. 2020. Decentralized authentication of distributed patients in hospital networks using blockchain. *IEEE journal of biomedical and health informatics*, 24(8), 2146-2156.

[15] Bayrakdar, M. E. 2020. Cooperative communication based access technique for sensor networks. *International Journal of Electronics*, *107*(2), 212-225.

[16] Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Jhaveri, R. H., & Chowdhary, C. L. 2021. Performance assessment of supervised classifiers for designing intrusion detection systems: a comprehensive review and recommendations for future research. *Mathematics*, *9*(6), 690.

[17] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Zhang, Q., & Choo, K. K. R. 2020. An energy-efficient SDN controller architecture for IoT networks with blockchain-based security. *IEEE Transactions on Services Computing*, *13*(4), 625-638.

[18] Liu, Y., Garg, S., Nie, J., Zhang, Y., Xiong, Z., Kang, J., & Hossain, M. S. 2020. Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach. *IEEE Internet of Things Journal*, *8*(8), 6348-6358.

[19] Bayrakdar, M. E. 2020. Exploiting cognitive wireless nodes for priority-based data communication in terrestrial sensor networks. *ETRI Journal*, *42*(1), 36-45.

[20] Ijaz, M. F., Alfian, G., Syafrudin, M., & Rhee, J. 2018. Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest. *Applied sciences*, *8*(8), 1325.

[21] Yazdinejadna, A., Parizi, R. M., Dehghantanha, A., & Khan, M. S. 2021. A kangaroo-based intrusion detection system on software-defined networks. *Computer Networks*, *184*, 107688.

[22] Bayrakdar, M. E. 2020. Employing sensor network based opportunistic spectrum utilization for agricultural monitoring. *Sustainable Computing: Informatics and Systems*, *27*, 100404.

[23] Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K. K. R., & Parizi, R. M. 2020. An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic. *IEEE Internet of Things Journal*, *7*(9), 8852-8859.

[24] Abdel-Basset, M., Hawash, H., & Sallam, K. 2021. Federated threat-hunting approach for microservice-based industrial cyber-physical system. *IEEE Transactions on Industrial Informatics*, *18*(3), 1905-1917.

[25] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., & Karimipour, H. 2021. Federated learning for drone authentication. *Ad Hoc Networks*, *120*, 102574.

[26] Gao, P., Shao, F., Liu, X., Xiao, X., Qin, Z., Xu, F., ... & Song, D. 2021. Enabling efficient cyber threat hunting with cyber threat intelligence. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (pp. 193-204). IEEE.

[27] Liang, F., Yu, W., Liu, X., Griffith, D., & Golmie, N. 2020. Toward edge-based deep learning in industrial Internet of Things. *IEEE Internet of Things Journal*, *7*(5), 4329-4341.

[28] Khalil, R. A., Saeed, N., Masood, M., Fard, Y. M., Alouini, M. S., & Al-Naffouri, T. Y. 2021. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet of Things Journal*, *8*(14), 11016-11040.

[29] Zhu, S., Ota, K., & Dong, M. 2021. Green AI for IIoT: Energy efficient intelligent edge computing for industrial internet of things. *IEEE Transactions on Green Communications and Networking*, *6*(1), 79-88.

[30] Ullah, F., Naeem, H., Jabbar, S., Khalid, S., Latif, M. A., Al-Turjman, F., & Mostarda, L. 2019. Cyber security threats detection in internet of things using deep learning approach. *IEEE access*, *7*, 124379-124389.

[31] Zhang, Q., Yang, L. T., Chen, Z., Li, P., & Bu, F. 2018. An adaptive dropout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing. *IEEE transactions on industrial informatics*, *15*(4), 2330-2337.

[32] Binbusayyis, A., & Vaiyapuri, T. 2019. Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach. *Ieee Access*, *7*, 106495-106513.

[33] Binbusayyis, A., & Vaiyapuri, T. 2021. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. *Applied Intelligence*, *51*(10), 7094-7108.

[34] Christiansen, P., Nielsen, L. N., Steen, K. A., Jørgensen, R. N., & Karstoft, H. 2016. DeepAnomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors*, *16*(11), 1904.

[35] Kang, M. J., & Kang, J. W. 2016. Intrusion detection system using deep neural network for in-vehicle network security. *PloS one*, *11*(6), e0155781.

[36] Khalil, R. A., Saeed, N., Masood, M., Fard, Y. M., Alouini, M. S., & Al-Naffouri, T. Y. 2021. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet of Things Journal*, 8(14), 11016-11040.

[37] Cruz, T., Rosa, L., Proença, J., Maglaras, L., Aubigny, M., Lev, L., ... & Simões, P. 2016. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics*, *12*(6), 2236-2246.

[38] Grill, M., Pevný, T., & Rehak, M. 2017. Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences*, *83*(1), 43-57.

[39] Primartha, R., & Tama, B. A. 2017. Anomaly detection using random forest: A performance revisited. In *2017 International conference on data and software engineering (ICoDSE)* (pp. 1-6). IEEE.

[40] Vijayanand, R., Devaraj, D., & Kannapiran, B. 2018. Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. *Computers & Security*, *77*, 304-314.

[41] Dahiya, P., & Srivastava, D. K. 2018. Network intrusion detection in big dataset using spark. *Procedia computer science*, *132*, 253-262.

[42] Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G. G., & Chen, J. 2018. Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics*, *14*(7), 3187-3196.

[43] Hurrah, N. N., Parah, S. A., Sheikh, J. A., Al-Turjman, F., & Muhammad, K. 2019. Secure data transmission framework for confidentiality in IoTs. *Ad Hoc Networks*, *95*, 101989.

[44] Deebak, B. D., & Al-Turjman, F. 2020. A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks. *Ad Hoc Networks*, *97*, 102022.

[45] Dragos, A. 2017. Trisis malware: Analysis of safety system targeted malware. *Dragos version 1. 20171213*.

[46] Robertson, J., & Riley, M. 2014. Mysterious' 08 turkey pipeline blast opened new cyberwar era. *Bloomberg Business*.

[47] Lee, R. M., Assante, M. J., & Conway, T. the Baku-Tbilisi-Ceyhan (BTC) pipeline Cyber Attack.

[48] Lee, R. M., Assante, M. J., & Conway, T. 2016. Analysis of the recent reports of attacks on US infrastructure by Iranian Actors. *SANS Industrial Control Systems 4*, *9*.

[49] ITS Committee. 2013. IEEE standard for wireless access in vehicular environments-security services for applications and management messages. *IEEE Vehicular Technology Society*, *1609*(2)..

[50] McCarthy, J., Faatz, D., Urlaub, N., Wiltberger, J., & Yimer, T. 2021. *Securing the Industrial Internet of Things: Cybersecurity for Distributed Energy Resources* (No. NIST Special Publication (SP) 1800-32 (Withdrawn)). National Institute of Standards and Technology.

[51] Thamilarasu, G., & Chawla, S. 2019. Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*, *19*(9), 1977.

[52] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. 2017. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE access*, *6*, 1792-1806.

[53] Chawla, A., Lee, B., Fallon, S., & Jacob, P. 2019. Host based intrusion detection system with combined CNN/RNN model. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18* (pp. 149-158). Springer International Publishing.

[54] Wang, C., Wang, B., Liu, H., & Qu, H. 2020. Anomaly detection for industrial control system based on autoencoder neural network. *Wireless Communications and Mobile Computing*, *2020*(1), 8897926.

[55] Luo, T., & Nagarajan, S. G. 2018. Distributed anomaly detection using autoencoder neural networks in WSN for IoT. In *2018 ieee international conference on communications (icc)* (pp. 1-6). IEEE.

[56] Huang, J., Kong, L., Chen, G., Wu, M. Y., Liu, X., & Zeng, P. 2019. Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism. *IEEE Transactions on Industrial Informatics*, *15*(6), 3680-3689.

[57] Huda, S., Miah, S., Yearwood, J., Alyahya, S., Al-Dossari, H., & Doss, R. 2018. A malicious threat detection model for cloud assisted internet of things (CoT) based industrial control system (ICS) networks using deep belief network. *Journal of Parallel and Distributed Computing*, *120*, 23-31.

[58] Chen, N., Qiu, T., Mu, C., Han, M., & Zhou, P. 2020. Deep actor–critic learning-based robustness enhancement of Internet of Things. *IEEE Internet of Things Journal*, *7*(7), 6191-6200.

[59] Liu, M., Yu, F. R., Teng, Y., Leung, V. C., & Song, M. 2019. Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach. *IEEE Transactions on Industrial Informatics*, *15*(6), 3559-3570.

[60] Kim, J., Ban, Y., Ko, E., Cho, H., & Yi, J. H. 2022. MAPAS: a practical deep learning-based android malware detection system. *International Journal of Information Security*, *21*(4), 725-738.

[61] Panchal, A. C., Khadse, V. M., & Mahalle, P. N. 2018. Security issues in IIoT: A comprehensive survey of attacks on IIoT and its countermeasures. In *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)* (pp. 124-130). IEEE.

[62] Yu, X., & Guo, H. 2019. A survey on IIoT security. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)* (pp. 1-5). IEEE.

[63] Mohanta, B. K., Jena, D., Satapathy, U., & Patnaik, S. 2020. Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of things*, *11*, 100227.

[64] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. 2019. A survey on IoT security: application areas, security threats, and solution architectures. *IEEe Access*, *7*, 82721-82743.

[65] Bajic, E. 2018. Localisation et identification de ressources industrielles par l'Internet des objets. *GeSI-Revue des Départements de Génie Electrique et Informatique Industrielle*, (92), 19-27.

[66] Cheng, C. F., Chen, Y. C., & Lin, J. C. W. 2020. A carrier-based sensor deployment algorithm for perception layer in the IoT architecture. *IEEE Sensors Journal*, *20*(17), 10295-10305.

[67] Kaur, H., & Kumar, R. 2021. A survey on Internet of Things (IoT): Layer-specific, domain-specific and industry-defined architectures. In *Advances in Computational Intelligence and Communication Technology: Proceedings of CICT 2019* (pp. 265-275). Springer Singapore.

[68] HaddadPajouh, H., Khayami, R., Dehghantanha, A., Choo, K. K. R., & Parizi, R. M. 2020. AI4SAFE-IoT: An AI-powered secure architecture for edge layer of Internet of things. *Neural Computing and Applications*, *32*(20), 16119-16133.

[69] Abdullah, A., Kaur, H., & Biswas, R. 2020. Universal layers of IoT architecture and its security analysis. In *New Paradigm in Decision Science and Management: Proceedings of ICDSM 2018* (pp. 293-302). Springer Singapore.

[70] Mrabet, H., Belguith, S., Alhomoud, A., & Jemai, A. 2020. A survey of IoT security based on a layered architecture of sensing and data analysis. *Sensors*, *20*(13), 3625.

[71] Alidoosti, M., Nowroozi, A., & Nickabadi, A. 2020. Evaluating the web-application resiliency to business-layer DoS attacks. *ETRI Journal*, *42*(3), 433-445.

[72] Patnaik, R., Padhy, N., & Srujan Raju, K. 2021. A systematic survey on IoT security issues, vulnerability and open challenges. In *Intelligent System Design: Proceedings of Intelligent System Design: INDIA 2019* (pp. 723-730). Springer Singapore.

[73] Sodhro, A. H., Pirbhulal, S., & De Albuquerque, V. H. C. 2019. Artificial intelligence-driven mechanism for edge computing-based industrial applications. *IEEE Transactions on Industrial Informatics*, *15*(7), 4235-4243.

[74] Li, F., Shi, Y., Shinde, A., Ye, J., & Song, W. 2019. Enhanced cyber-physical security in internet of things through energy auditing. *IEEE Internet of Things Journal*, *6*(3), 5224-5231.

[75] Benkhelifa, E., Welsh, T., & Hamouda, W. 2018. A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems. *IEEE communications surveys & tutorials*, *20*(4), 3496-3509.

[76] Garg, S., Kaur, K., Kumar, N., Kaddoum, G., Zomaya, A. Y., & Ranjan, R. 2019. A hybrid deep learning-based model for anomaly detection in cloud datacenter networks. *IEEE Transactions on Network and Service Management*, *16*(3), 924-935.

[77] Qiu, M., Dai, H. N., Sangaiah, A. K., Liang, K., & Zheng, X. 2019. Guest editorial: Special section on emerging privacy and security issues brought by artificial intelligence in industrial informatics. *IEEE Transactions on Industrial Informatics*, *16*(3), 2029-2030.

[78] Tschofenig, H., & Baccelli, E. 2019. Cyberphysical security for the masses: A survey of the internet protocol suite for internet of things security. *IEEE Security & Privacy*, *17*(5), 47-57.

[79] Zhang, Y., Krishnan, V. V., Pi, J., Kaur, K., Srivastava, A., Hahn, A., & Suresh, S. 2019. Cyber physical security analytics for transactive energy systems. *IEEE Transactions on Smart Grid*, *11*(2), 931-941.

[80] Zhou, Z., Chen, K., & Zhang, J. 2015. Efficient 3-D scene prefetching from learning user access patterns. *IEEE Transactions on Multimedia*, *17*(7), 1081-1095.

[81] Liu, C. H., Lin, Q., & Wen, S. 2018. Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, *15*(6), 3516-3526.

[82] Hassija, V., Chamola, V., Gupta, V., Jain, S., & Guizani, N. 2020. A survey on supply chain security: Application areas, security threats, and solution architectures. *IEEE Internet of Things Journal*, *8*(8), 6222-6246.

[83] Alasmary, H., Khormali, A., Anwar, A., Park, J., Choi, J., Abusnaina, A., ... & Mohaisen, A. 2019. Analyzing and detecting emerging Internet of Things malware: A graph-based approach. *IEEE Internet of Things Journal*, *6*(5), 8977-8988.

[84] Celik, Z. B., McDaniel, P., Tan, G., Babun, L., & Uluagac, A. S. 2019. Verifying internet of things safety and security in physical spaces. *IEEE Security & Privacy*, *17*(5), 30-37.

[85] Rabieinejad, E., Yazdinejad, A., Dehghantanha, A., Parizi, R. M., & Srivastava, G. 2021. Secure ai and blockchain-enabled framework in smart vehicular networks. In *2021 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.

[86] Yazdinejad, A., Rabieinejad, E., Dehghantanha, A., Parizi, R. M., & Srivastava, G. 2021. A machine learning-based sdn controller framework for drone management. In *2021 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.

[87] Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, *7*, 100059.

[88] Jabbar, M. A., & Aluvalu, R. 2017. RFAODE: A novel ensemble intrusion detection system. *Procedia computer science*, *115*, 226-234.

[89] Muna, A. H., Moustafa, N., & Sitnikova, E. 2018. Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of information security and applications*, *41*, 1-11.

[90] Behera, S., Choubey, A., Kanani, C. S., Patel, Y. S., Misra, R., & Sillitti, A. 2019. Ensemble trees learning based improved predictive maintenance using IIoT for turbofan engines. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 842-850).

[91] Alhalabi, B., Gaber, M., & Basurra, S. 2020. Prune2edge: A multi-phase pruning pipelines to deep ensemble learning in iiot. *arXiv preprint arXiv:2004.04710*.

[92] Gu, T., Abhishek, A., Fu, H., Zhang, H., Basu, D., & Mohapatra, P. 2020. Towards learning-automation IoT attack detection through reinforcement learning. In *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)* (pp. 88-97). IEEE.

[93] Kurniawan, A. A., Santoso, H. A., Soeleman, M. A., & Fanani, A. Z. 2019. Intrusion detection system as audit in IoT infrastructure using ensemble learning and SMOTE method. In *2019 5th International Conference on Science in Information Technology (ICSITech)* (pp. 205-210). IEEE.

[94] Abdel-Basset, M., Hawash, H., & Sallam, K. 2021. Federated threat-hunting approach for microservice-based industrial cyber-physical system. *IEEE Transactions on Industrial Informatics*, *18*(3), 1905-1917.

[95] Liu, Y., Garg, S., Nie, J., Zhang, Y., Xiong, Z., Kang, J., & Hossain, M. S. 2020. Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach. *IEEE Internet of Things Journal*, *8*(8), 6348-6358.

[96] Chen, B., & Wan, J. 2019. Emerging trends of ml-based intelligent services for industrial internet of things (iiot). *2019 Computing, Communications and IoT Applications (ComComAp)*, 135-139.

[97] Ozanich, E., Gerstoft, P., & Niu, H. 2020. A feedforward neural network for direction-of-arrival estimation. *J. Acoust. Soc. Am.*, *147*(3), 2035-2048.

[98] Orimoloye, L. O., Sung, M. C., Ma, T., & Johnson, J. E. 2020. Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. *Expert Systems with Applications*, *139*, 112828.

[99] Zhang, N., & Sun, S. 2021. Multiview graph restricted Boltzmann machines. *IEEE Transactions on Cybernetics*, *52*(11), 12414-12428.

[100] Gu, L., Zhou, F., & Yang, L. 2020. Towards the representational power of restricted Boltzmann machines. *Neurocomputing*, *415*, 358-367.

[101] Deshwal, D., & Sangwan, P. 2020. A comprehensive study of deep neural networks for unsupervised deep learning. In *Artificial intelligence for sustainable development: Theory, practice and future applications* (pp. 101-126). Cham: Springer International Publishing.

[102] Chen, C., Ma, Y., & Ren, G. 2020. Hyperspectral classification using deep belief networks based on conjugate gradient update and pixel-centric spectral block features. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 4060-4069.

[103] Hong, C., Zeng, Z. Y., Fu, Y. Z., & Guo, M. F. 2020. Deep-belief-Networks based fault classification in power distribution networks. *IEEJ Transactions on Electrical and Electronic Engineering*, *15*(10), 1428-1435.

[104] Chu, H., Wei, J., Wu, W., Jiang, Y., Chu, Q., & Meng, X. 2021. A classification-based deep belief networks model framework for daily streamflow forecasting. *Journal of Hydrology*, *595*, 125967.

[105] Zhuang, X., Guo, H., Alajlan, N., Zhu, H., & Rabczuk, T. 2021. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. *European Journal of Mechanics-A/Solids*, *87*, 104225.

[106] Pawar, K., & Attar, V. Z. 2020. Assessment of autoencoder architectures for data representation. *Deep learning: concepts and architectures*, 101-132.

[107] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[108] Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[109] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[110] Boulila, W., Sellami, M., Driss, M., Al-Sarem, M., Safaei, M., & Ghaleb, F. A. 2021. RS-DCNN: A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification. *Computers and Electronics in Agriculture*, *182*, 106014.

[111] Zhang, Y., Wang, Y., & Luo, G. 2020. A new optimization algorithm for non-stationary time series prediction based on recurrent neural networks. *Future Generation Computer Systems*, *102*, 738-745.

[112] Boulila, W., Ghandorh, H., Khan, M. A., Ahmed, F., & Ahmad, J. 2021. A novel CNN-LSTM-based approach to predict urban expansion. *Ecological Informatics*, *64*, 101325.

[113] Kousik, N., Natarajan, Y., Raja, R. A., Kallam, S., Patan, R., & Gandomi, A. H. 2021. Improved salient object detection using hybrid Convolution Recurrent Neural Network. *Expert Systems with Applications*, *166*, 114064.

[114] Aggarwal, A., Mittal, M., & Battineni, G. 2021. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, *1*(1), 100004.

[115] Yoon, J., Drumright, L. N., & Van Der Schaar, M. 2020. Anonymization through data synthesis using generative adversarial networks (ADS-GAN). *IEEE journal of biomedical and health informatics*, *24*(8), 2378-2388.

[116] Fekri, M. N., Ghosh, A. M., & Grolinger, K. 2019. Generating energy data for machine learning with recurrent generative adversarial networks. *Energies*, *13*(1), 130.

[117] Li, H., Ota, K., & Dong, M. 2018. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE network*, *32*(1), 96-101.

[118] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. 2018. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12-22.

[119] Tang, J., Sun, D., Liu, S., & Gaudiot, J. L. 2017. Enabling deep learning on IoT devices. *Computer*, *50*(10), 92-96.

[120] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

[121] Le Minh, K. H., Le, K. H., & Le-Trung, Q. 2020. Dlase: A light-weight framework supporting deep learning for edge devices. In *2020 4th International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)* (pp. 103-108). IEEE.

[122] Louis, M. S., Azad, Z., Delshadtehrani, L., Gupta, S., Warden, P., Reddi, V. J., & Joshi, A. 2019. Towards deep learning using tensorflow lite on risc-v. In *Third Workshop on Computer Architecture Research with RISC-V (CARRV)* (Vol. 1, p. 6). ACM.

[123] Wang, X., Magno, M., Cavigelli, L., & Benini, L. 2020. FANN-on-MCU: An open-source toolkit for energy-efficient neural network inference at the edge of the Internet of Things. *IEEE Internet of Things Journal*, *7*(5), 4403-4417.

[124] Aghapour, E., Pathania, A., & Ananthanarayanan, G. 2021. Integrated ARM big. Little-Mali pipeline for high-throughput CNN inference. *Univ. Amsterdam, Amsterdam, The Netherlands, Tech. Rep. TCAD-2021-0385*.

[125] Roopaei, M., Rad, P., & Jamshidi, M. 2017. Deep learning control for complex and large scale cloud systems. *Intelligent Automation & Soft Computing*, *23*(3), 389-391.

[126] Robertson, C., Li, J., Ohira, R., Nguyen, Q. V. H., & Jo, J. 2019. Optimising Deep Learning Split Deployment for IoT Edge Networks. In *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)* (pp. 346-351). IEEE.

[127] Han, S., Mao, H., & Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

[128] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., & Temam, O. 2014. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. *ACM SIGARCH Computer Architecture News*, *42*(1), 269-284.

[129] Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., ... & Temam, O. 2014. Dadiannao: A machine-learning supercomputer. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture* (pp. 609-622). IEEE.

[130] Hussein, D. H., & Askar, S. 2023. Federated learning enabled SDN for routing emergency safety messages (ESMs) in IoV under 5G environment. *IEEE Access*, *11*, 141723-141739.

[131] Abdulazeez, D. H., & Askar, S. K. 2024. A novel offloading mechanism leveraging fuzzy logic and deep reinforcement learning to improve IoT application performance in a three-layer architecture within the fog-cloud environment. *IEEE Access*.

[132] Ibrahim, M. A., & Askar, S. 2023. An intelligent scheduling strategy in fog computing system based on multi-objective deep reinforcement learning algorithm. *IEEE Access*, *11*, 133607-133622.

[133] Abdulazeez, D. H., & Askar, S. K. 2023. Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing environment. *Ieee Access*, *11*, 12555-12586.

[134] Othman, M., Ali, D., & Abdullah, N. 2024. Deep Learning Based Security Schemes for IoT Applications: A Review. *The Indonesian Journal of Computer Science*, *13*(2).

[135] Ali, D., & Abdullah, N. 2024. Deep Learning in Medical Image Analysis Article Review. *The Indonesian Journal of Computer Science*, *13*(2).

[136] Pallathadka, H., Askar, S., Kulshreshta, A., Sharma, M. K., Widatalla, S., & Mudae, I. S. 2024. Economic and Environmental Energy Scheduling of Smart Hybrid Micro Grid Based on Demand Response. *International Journal of Integrated Engineering*, *16*(9), 351-365.

[137] Husain, B. H., & Askar, S. 2022. Smart resource scheduling model in fog computing. In *2022 8th International Engineering Conference on Sustainable Technology and Development (IEC)* (pp. 96-101). IEEE.

[138] Zhang, L., Askar, S., Alkhayyat, A., Samavatian, M., & Samavatian, V. 2024. Machine learning-driven detection of anomalies in manufactured parts from resonance frequency signatures. *Nondestructive Testing and Evaluation*, 1-23.

[139] Yang, Y., Patil, N., Askar, S., & Kumar, A. 2025. Machine learning-guided study of residual stress, distortion, and peak temperature in stainless steel laser welding. *Applied Physics A*, *131*(1), 44.

[140] Horrell, M., Reynolds, L., & McElhinney, A. 2020. Data science in heavy industry and the Internet of Things. *Harvard Data Science Review*, *2*(2).

[141] Sezer, O. B., Dogdu, E., & Ozbayoglu, A. M. 2017. Context-aware computing, learning, and big data in internet of things: a survey. *IEEE Internet of Things Journal*, *5*(1), 1-27.

[142] Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. 2020. Deep learning algorithms for bearing fault diagnostics—A comprehensive review. *IEEE access*, *8*, 29857-29881.

[143] Rymarczyk, T., Kłosowski, G., Kozłowski, E., & Tchórzewski, P. 2019. Comparison of selected machine learning algorithms for industrial electrical tomography. *Sensors*, *19*(7), 1521.

[144] Wang, Y., Pan, Z., Yuan, X., Yang, C., & Gui, W. 2020. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA transactions*, *96*, 457-467.

[145] Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., ... & Zaharia, M. 2019. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*.

[146] Jiang, F., Fu, Y., Gupta, B. B., Liang, Y., Rho, S., Lou, F., ... & Tian, Z. 2018. Deep learning based multi-channel intelligent attack detection for data security. *IEEE transactions on Sustainable Computing*, *5*(2), 204-212.

[147] Zhang, S., Yao, L., Sun, A., & Tay, Y. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, *52*(1), 1-38.

[148] Alvarez-Gonzalez, F., Griffo, A., Sen, B., & Wang, J. 2017. Real-time hardware-in-the-loop simulation of permanent-magnet synchronous motor drives under stator faults. *IEEE Transactions on Industrial Electronics*, *64*(9), 6960-6969.

[149] Askar, S., Zervas, G., Hunter, D. K., & Simeonidou, D. 2010. Classified cloning for QoS provisioning in OBS networks. In *36th European Conference and Exhibition on Optical Communication* (pp. 1-3). IEEE.

[150] Samann, F. E., Ameen, S. Y., & Askar, S. 2022. Fog computing in 5g mobile networks: a review. In *2022 4th International Conference on Advanced Science and Engineering (ICOASE)* (pp. 142-147). IEEE.

[151] Omer, S. M., Ghafoor, K. Z., & Askar, S. K. 2024. Lightweight improved yolov5 model for cucumber leaf disease and pest detection based on deep learning. *Signal, Image and Video Processing*, *18*(2), 1329-1342.

[152] Askar, S. 2017. SDN-based load balancing scheme for fat-tree data center networks. *Al-Nahrain Journal for Engineering Sciences*, *20*(5), 1047-1056.

[153] Askar, S., Zervas, G., Hunter, D. K., & Simeonidou, D. 2011. A novel ingress node design for video streaming over optical burst switching networks. *Optics Express*, *19*(26), B191-B196.

[154] Bi, Z. M., & Wang, L. 2010. Advances in 3D data acquisition and processing for industrial applications. *Robotics and Computer-Integrated Manufacturing*, *26*(5), 403-413.

[155] Zhou, L., Wu, D., Chen, J., & Dong, Z. 2017. When computation hugs intelligence: Content-aware data processing for industrial IoT. *IEEE Internet of Things Journal*, *5*(3), 1657-1666.

[156] Yao, S., Zhao, Y., Shao, H., Zhang, C., Zhang, A., Hu, S., ... & Abdelzaher, T. 2018. Sensegan: Enabling deep learning for internet of things with a semi-supervised framework. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, *2*(3), 1-21.

[157] Tschuchnig, M. E., Ferner, C., & Wegenkittl, S. 2020. Sequential iot data augmentation using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4212-4216). IEEE.

[158] Mehdiyev, N., Lahann, J., Emrich, A., Enke, D., Fettke, P., & Loos, P. 2017. Time series classification using deep learning for process planning: A case from the process industry. *Procedia Computer Science*, *114*, 242-249.

[159] Latif, S., Driss, M., Boulila, W., Huma, Z. E., Jamal, S. S., Idrees, Z., & Ahmad, J. 2021. Deep learning for the industrial internet of things (iiot): A comprehensive survey of techniques, implementation frameworks, potential applications, and future directions. *Sensors*, *21*(22), 7518.