

Intelligent IoT Surveillance and Instantaneous Management

Ying Huang, Yongmei Su and Shuanggui Lu

Abstract—Unmanned Vehicles (UVs) and the Industry Internet of Things (IIoT) are two examples of new technology being incorporated into manufacturing processes during the fourth industrial revolution. IT networks must be machine-compatible to integrate these technologies; this includes addressing problems with connectivity, fog, and cloud-based computing security, lowering latency, and improving data reliability and standard of service. Regarding IIoT, AI techniques must handle resource management, network deployment, and these problems. The significant issues are unstable and high-latency communications between Industrial IIoT endpoints and the Cloud. By extending storage and computation to the network's edge, fog computing offers a valuable tool for merging intricately linked processing systems. Interoperability may be addressed using fog in an IIoT gateway and advanced software distributed on the edge. However, as an IIoT gateway is essential to processing and delivering data to many systems and platforms, selecting one is critical regarding accuracy and latency. Intelligent IIoT monitoring and real-time control based on Integrating Autonomous Robots for Instantaneous Industrial Operations, visual recognition, and cloud/edge computing services are proposed to address these challenges. By deploying Deep Learning (DL) facilities close to customers who want them, latency and processing costs associated with transmitting data through the Cloud may be minimized. The suggested methods enhance platform decision-making and industrial automation system performance by integrating cloud-based services into an operational loop. A smart approach that offers a trade-off between accuracy and latency is suggested to choose the right AI for the situation under observation.

Index Terms—Cloud computing, Edge computing services, Integrating autonomous robots, Visual recognition,

Original Research Paper
DOI: 10.53314/ELS2529003H

Manuscript received on July 24th, 2024. Received in revised form on December 23rd, 2024. Accepted for publication on December 30th, 2024.

Ying Huang is with the Faculty of Guangxi Eco-engineering Vocational and Technical College, Guangxi, P.R. China (e-mail: huangying800816@163.com).

Yongmei Su is with the Faculty of Zhengzhou Technical College, Zhengzhou, P.R. China.

Shuanggui Lu is with the Faculty of Guangxi Eco-engineering Vocational and Technical College, Guangxi, P.R. China.

This paper was supported by the Talent Introduction Scientific Research Project (NO.GXSTKYZX202406002) of Guangxi Ecological Engineering Vocational and Technical College, the first phase of the training project of famous teachers (craftsmen), and the project construction project of famous master studios.

I. INTRODUCTION

MANUFACTURING has entered a new digital era, introduced by the Fourth Industrial Revolution [1], characterized by lightning-fast technical development. Uncrewed vehicles (UVs), automation through robots and AI, and the Internet of Things (IIoT) [2], which connect physical things over the Internet, are vital technologies accelerating this change. Manufacturing IT networks must adapt to accommodate machines that use these technologies entirely. It includes fixing essential issues [3] with connection, security, latency, dependability, and quality of service. The unreliable, high-latency [4] interactions between industrial IIoT endpoints and cloud computing services have been recognized as a severe obstacle. It makes real-time control, monitoring, and decision-making more difficult. Fog computing [5], which extends storage and processing capabilities to the network's edge, has emerged as a viable solution. It allows time-critical interactions between equipment, robots, sensors, and other endpoints [6]. Fog resources in IIoT gateways can also address interoperability issues in industrial systems, software, and platforms. However, selecting the correct IIoT gateway is critical since it organizes vital operations such as raw data processing and delivering valuable insights to numerous platforms [7]. Many researchers suggest implementing an intelligent system that utilizes fog computing, enhanced edge software, and strategic integration with cloud-based artificial intelligence to enhance factory automation [8]. A critical advancement is implementing deep learning capabilities at the edge, close to the source of data generation. This minimizes the delay and expenses of transferring substantial data to the Cloud [9]. Local execution enables the performance of crucial tasks such as visual identification, proactive maintenance, and immediate control of robots. The Cloud offers additional storage and computational capabilities to obtain long-term insights. This hybrid edge-cloud architecture enables the customization of the optimal combination of edge devices and cloud services [10] for individual functions. The edge handles jobs that need low latency, whereas the cloud architecture facilitates the ongoing improvement of complicated learning algorithms over extended periods. The system automatically distributes workloads across layers in response to the current condition and application needs.

Due to their high computation, administration, and storage needs, traditional techniques are no longer accessible on end devices. Because of this, to achieve the recently built edge-cloud monitoring with smart IIoT applications, it is essential to use the lightweight learning model for intelligent object iden-

tification and to deploy the efficient training process in edge devices. Cloud monitoring systems have recently been popular among innovative applications because they emphasize managing the constantly generated data sequences from a comprehensive surveillance environment. Therefore, multi-object detection has gained a lot of interest from academics and businesses alike and has become a significant technology. Suppose innovative surveillance systems manage massive amounts of IoT data effectively. In that case, they must overcome obstacles such as the detection and localization of moving objects and the recognition and extraction of immediate features. This calls for powerful real-time processing capabilities. The proliferation of surveillance equipment and the ongoing improvement of large-scale, excellent surveillance data in IoT contexts pose increasing challenges to existing paradigms that rely on centralized processing. A primary issue is the widening disparity between the rate of data generation for surveillance and the capability of cloud-based data processing. The projected data captured by the camera alone will surpass 869PB by 2020's conclusion. In addition, new methods are needed to accomplish real-time, low-latency, energy-efficient, and highly accurate surveillance operations because of the enormous bandwidth requirements of transmitting massive volumes of data across Cloud surveillance systems. Creating a new edge-cloud monitoring architecture that moves the heavy processing loads away from the leading cloud server is crucial. Deploying the learned AI models in edge servers allows for the construction of an intelligent IoT system that efficiently addresses the aforementioned time-consuming issues through distributed and dependable computing.

State-of-the-art methods have shown promising results for identifying objects in static, stationary situations; nevertheless, handling complicated dynamic scenes remains challenging. Location, density, motion, and variations in illumination are some of the many variables that might impact object detection and tracking tasks in dynamic situations. Furthermore, present-day intelligent applications in ever-changing surveillance settings necessitate real-time data processing, communications, and control, including visual object detection, emotion analysis, and identity recognition.

UVs and autonomous robots equipped with sophisticated sensors are essential to their execution on the manufacturing floor. Edge computing [11] with computer vision and deep learning models allows the robots to react quickly to events as they happen. Under the guidance of a 5G connection, UVs oversee materials handling and inventory. Internet of Things [12] gateways clean, normalize, and route data flows from these devices to production dashboards, maintenance logs, and inventory management applications [13]. Cloud services offer centrally stored data lakes that may be used to aggregate datasets, find trends, and train prediction algorithms. These are subsequently distributed to edge devices for quick inference and automation. The hybrid design generates a responsive, intelligent system that supplements human operators. Because of edge, Cloud, and AI integration, manufacturing processes may be adjusted in real time while continually improving. This research offers significant insights into effectively utilizing fog

computing, sophisticated edge software, and strategic coordination with the Cloud and AI to exploit the promise of new technologies in smart manufacturing fully. The suggested intelligent system attempts to augment automation, efficiency, quality, and production. The wide-ranging relevance in several industries, including automotive, pharmaceuticals, electronics, and others adopting the Industry 4.0 approach [14],

The objective and contribution of this research are listed below.

- To design the intelligent system architecture with the help of edge software, fog computing, Cloud, and AI integration to monitor optimized intelligent manufacturing applications.
- To apply deep learning techniques for addressing security, connectivity, reliability, latency, and quality of service challenges in information technologies.
- To provide edge-based solutions using deep learning capabilities to minimize the latency and cost of industrial IoT endpoints.

The rest of the manuscript is organized as follows: Section II describes the various researcher's opinions regarding the Industrial IoT endpoints related to intelligent solutions for the industry's multiple problems. Section III explains the working process of intelligent system architecture using edge computing and cloud applications. Section IV discusses the system's efficiency, and a conclusion is described in Section V.

II. RELATED WORKS

Essien and Giannetti introduced convolutional long short-term memory (CLSTM) in intelligent manufacturing to improve the system performance [15]. This study intends to improve throughput and reduce energy consumption. During the analysis, temporal and spatial distribution data is utilized on the CLSTM system to improve forecasting accuracy. The architecture has an encoder-decoder structure that explores every piece of information and maximizes industrial performance. The efficiency of the plant is evaluated in the United Kingdom, and the system ensures optimized results compared to other methods.

Chen et al. applied Deep Reinforcement Learning (DRL) in the industrial Internet of Things to manage the resources in mobile edge computing [16]. This work aims to minimize the delay and solve the Markov decision process problem. The introduced resource management process determines the policy gradient for every action and state. According to the actions, the resources are managed with minimum delay and maximum throughput.

Kishor et al. introduced the Intelligent Multimedia Data Segregation (IMDS) fog computing approach to minimize the latency in healthcare applications [17]. The IMDS approach uses the K-fold random forest approach to reduce storage, network, and transmission delays. The machine learning approach segregates the input using the random forest approach, which makes the healthcare decision up to 92% compared to other methods and reduces the latency by 95% compared to pre-existing models.

Fantacci and Picano recommended a constrained data offloading scheme (DCDO) for analyzing the performance of cloud-fog-edge systems [18]. This study uses the queuing theory to explore the performance of computing infrastructure. The requests are processed according to the deadline expiration using the Markov queueing system. In addition, the computation resource allocation approach is applied to maximize the quality of services in computing infrastructure.

Abbasi et al. applied a Genetic Algorithm (GA) with Intelligent workload allocation in Cloud and fog computing structure to maximize the allocation efficiency [19]. The ultimate goal of this study is to reduce delay and consumption power using the optimized resource allocation method. The green energy concept is initially applied to the multi-sensorial framework to manage data security. In addition, the GA approach is introduced to handle the large request volume by minimizing edge devices' delay and consumption power.

Mubarakali et al. suggested a delay-sensitive data transmission (DSDT) approach to managing multimedia applications' data storage and forwarding process [20]. The DSDT approach aims to create a robust solution while transferring data in fog computing with minimum delay. The approach processes the data nearer to the optimal network, and the solutions are selected by minimizing the latency. The practical exploration of sensitive data reduces the round trip time (15.67ms) and minimum delay on 1MB, 100KB, and 10KB data transmission.

Singh and Singh recommended the Energy Efficient Delay aware Task offloading (EEDTO) approach to manage the data in fog and cloud computing [21]. This study framework reduces the end-to-end delay according to the task allocation time. The optimal solution is framed by considering the edge server, IoT devices, and cloud components according to the delay. The offloading scheme is optimized using the levy flight moth flame optimization technique that reduces the energy by 22% compared to other offloading algorithms.

Siasi et al. introduced delay aware service function chain (DASFC) in hybrid fog-cloud computing to improve the computing performance [22]. The system aims to minimize the delay and manage the resources using the DASFC architecture. The system comprises a single cloud and fog layer that can handle the delay-tolerant and delay-sensitive information. The trade-off between the fog solution and Cloud standalone for every user request is examined to enhance the hybrid architecture. The effective utilization of this architecture minimizes energy consumption (28-30% less), realization cost (28-30% less), network delay (21-43% less), and high transmission rate (15-40% higher) effective.

Shukla et al. conducted a literature study to analyze latency in IoT using cloud computing techniques [23]. The study uses the PRISMA techniques to explore the various researcher's studies to minimize the latency for time-sensitive applications. During the analysis, 32 technologies and 23 approaches were analyzed via 112 papers, in which multiple researchers introduced IoT and AI techniques to minimize latency and improve throughput.

Jafari and Rezvani applied a non-dominant sorting genetic algorithm (NSGA-II) to optimize energy consumption and de-

lay in IoT-fog-cloud computing [24]. This work aims to reduce the NP-hard and knapsack problems while solving the multi-objective problem. The sorting genetic algorithm is incorporated with the bee algorithm, providing optimal solutions with minimax differential evolution. The NSGA-II method examines every user request and provides the solution with minimum time delay, energy consumption, and high throughput.

Kuthadi et al. created a portable data security energy efficient framework (PDST-EEF) for managing data security in sensor networks [25]. This study uses the cryptographic signature model to identify different attacks. During the analysis, replay, and denial of services, attacks are identified while transferring data from the innovative grid environment. The framework observes every node's energy consumption; this factor maintains the network's reliability and throughput. According to various researchers' analyses, industrial applications' throughput, delay, energy, and other factors are maintained using artificial intelligence techniques. In addition, Cloud computing techniques and computing resources are widely applied to improve network performance. However, industrial operational resources fail to manage network reliability, connectivity, security, and latency, which are challenging.

The management of network dependability, latency, and security, together with the effective allocation of resources and the making of decisions in real-time, provide substantial challenges to industrial IoT systems. Certain parameters, such as energy consumption, delay reduction, and throughput, have been improved by existing technologies that use AI-based frameworks, genetic algorithms, and edge-cloud computing. However, these solutions often face challenges regarding scalability, flexibility to changing contexts, and the smooth integration of resource management and monitoring duties. Furthermore, when dealing with large amounts of data from diverse Internet of Things devices, most frameworks fail to perform strongly under real-time limitations. This paper proposes a deep learning framework called ResNetDL to overcome these constraints. It combines the characteristics of ResNet topologies with edge-cloud computing. Through the integration of residual learning, distributed IoT monitoring, and instantaneous resource management, ResNetDL presents a new method. By lowering latency, increasing task prioritization, and guaranteeing safe data transfer over the network, the framework is aimed to boost real-time performance. The suggested approach establishes a new standard in industrial IoT applications by optimizing energy efficiency, improving network stability, and facilitating smooth scaling by exploiting ResNet's excellent feature extraction capabilities and temporal-spatial data processing.

III. ANALYSIS OF INTELLIGENT IOT SURVEILLANCE AND INSTANTANEOUS MANAGEMENT USING CLOUD AND EDGE COMPUTING RESOURCES

The fourth industrial technology utilizes the Internet of Things (IoT) and Unmanned Vehicles to improve industrial performance. Information technology widely utilizes cloud computing technologies to address latency, computing security,

quality of services, and data reliability issues. Artificial Intelligence (AI) and Internet of Things (IoT) techniques are incorporated in industrial applications to manage the resources and network deployment that address the high-latency communication and unstable issues between the Cloud and industrial IoT endpoint. In addition, complex interconnected systems are developed using fog computing because it extends the storage and network computations, effectively resolving the interoperability issues. Therefore, the effective selection of an IoT gateway and real-time monitoring system is one of the crucial factors in improving the industry's performance. Hence, the main objective of this study is to enhance industrial automation and decision-making by integrating them with cloud-related operational loops. Then, the deep learning model is deployed in computing technologies to minimize costs and latency while transferring data to the Cloud.

There will be strong ties between current and future military actions and the organizational theology viewpoint on the effects and consequences that fully autonomous self-governing systems will have on humanity. The military is now creating and using technology with the express purpose of killing people. Numerous sectors can benefit from the combination of self-governing robots with IoT surveillance. Drones and robots that can navigate themselves can monitor vast regions, sending out notifications and live video feeds if something unexpected happens. Large buildings, like public areas, industrial sites, and warehouses, can benefit from this security. Preventing breaches and unauthorized access to data gathered by IoT devices and sent via networks requires strong security measures. Because there are so many different protocols and standards, it can be difficult for various Internet of Things (IoT) devices and robots to connect and collaborate without a hitch. Following rules concerning privacy, security, and the operation of autonomous systems is critical.

A. Designing the Intelligent System Architecture with the Help of Edge Software, Fog Computing, Cloud, and AI Integration

The main objective of this phase is to improve the smart manufacturing application's environment by creating an Intelligent System Architecture using fog computing, edge software, AI integration, and cloud computing. The framework utilizes the deep learning model on IoT gateway and fog nodes to minimize the latency while processing the network edges. The deep learning embedded edge nodes observe and visualize the real-time environment, and anomalies are detected effectively. Offload non-critical background activities to the Cloud strategically, such as data aggregation and model retraining. Create protocols for dynamic workload allocation at the edge and in the Cloud. Create cloud-based digital twin simulations that replicate physical processes and evaluate control rules. Then, the Intelligent System architecture is shown in Fig. 1.

Fig. 1. illustrates the design of intelligent system architecture for industrial applications. The structure consists of three layers: edge, fog, and Cloud. Each layer is optimized using deep learning techniques to improve industrial operations' ro-

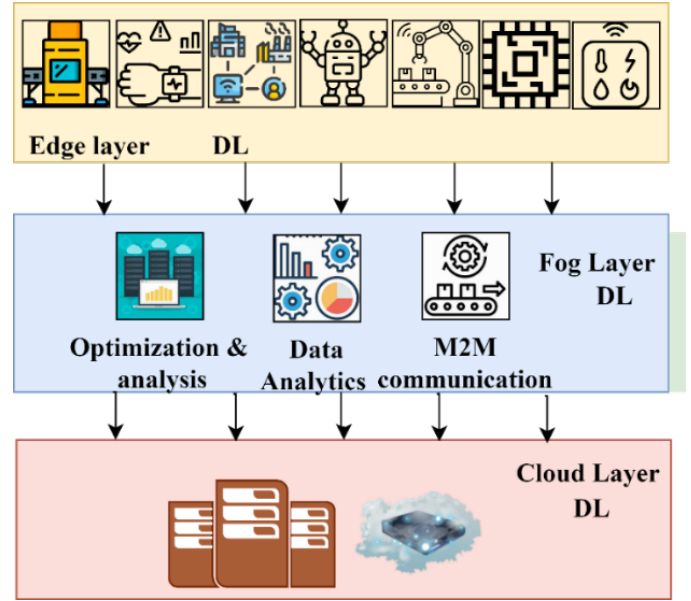


Fig. 1. Design of Intelligent System Architecture of industrial application

bustness, flexibility, and scalability. The first layer is the edge, which consists of robots, smart sensor devices, machines, the Internet of Things, etc., connected to the physical environment. The fog layer has data analytics, machine-to-machine communication units, and optimization and analysis units that help to explore the edge node-connected information. Finally, the cloud layer provides cloud storage and a data centre that can process industrial details effectively. In this architecture, every layer is optimized with a deep learning technique that helps to improve the overall real-time monitoring and analysis system. Initially, the nodes in the edge layer collect the raw data $x(t)$ at t time from an industrial process. The collected information is preprocessed using fog nodes by executing the Mfog module, which is defined using equation (1)

$$x'(t) = Mfog(x(t)) \quad (1)$$

In equation (1), $x(t)$ is defined as collected raw data, Mfog is the fog module function, and preprocessed data is denoted as $x'(t)$. The fog layer facilitates adjacent computational capacity for edge devices, enabling real-time and low-latency artificial intelligence processing. To accomplish this, fog nodes such as gateways and servers would function as organizers for deep learning models tuned for efficient inference. Assume the fog node collects information using the IoT nodes from the industrial environment. The gathered information is processed using Convolution Neural Networks (CNN), identifying whether the device is defective. The convolution model is trained using ResNet architecture to explore the large data volume, which helps predict defective devices with minimum computation delay and error. The CNN network works according to the weight-sharing concept between the layers. Frequently updating weight factors minimizes the error rate and optimizes the network process. The ResNet architecture is shown in Fig. 2.

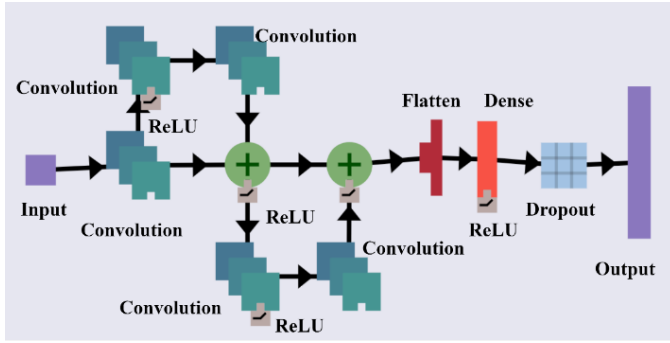


Fig. 2. Architecture of ResNet

Fig 2 illustrates the structure of ResNet, which includes several layers such as convolution, flatten, dense, and dropout layers. Each layer has a ReLU activation function to predict the output value. Here, the fog node collected input images are fed into the neural network that classifies the input as defective. The network has a sequence of convolution layers along with ReLU ($R(x)$) that is defined as $R(x) = \text{conv}(\text{ReLU}(\text{conv}(x)))$. The $R(x)$ is named as the residual block of the convolution layer. The residual block output is $R(x) + x$ in which x is defined as the identity pass-through. In the output computation, the neural network function is optimized using function $F(x)$ and the residual block is described as $R^*(x)$. Then the $R^*(x)$ is estimated by subtracting x from $F(x)$; $R^*(x) = F(x) - x$. The network sharing weight between the layers and residual block parameters are utilized to train the network. The effective utilization of each parameter and function helps to reduce the vanishing gradient problems. The flattened layer in the network takes multiple inputs and produces a single output (defective or not). The sequence of convolution or other layers output is concatenated to obtain the output layer. The network output is optimized by incorporating the output layer that eliminates irrelevant information. For instance, when the rate is set to 0.1, the dropout layer will randomly choose 10% of the weights on each forward pass and put them to 0. Introducing a dropout layer with a rate of 0 will not impact the network; however, a rate of 1 will cause the dropout layer to produce an output of 0. The overfitting issue is eliminated with the help of a regularization process in which new terms are added to the solution. The input pairs $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. The function f is utilized to predict the output for the given input by addressing the classification problems. The classification problem is defined as $\min_f \sum_{i=1}^n C(f(x_i), y_i)$. Here, C is the cost of $f(x_i)$ prediction when the y_i true value. Then, the regularized loss value is defined as $\min_f \sum_{i=1}^n C(f(x_i), y_i) + \lambda R(f)$. The regularization term $R(f)$ is larger when $\lambda > 0$ and larger f . Then, the regularization is performed using the L2-norm, which is defined as $\sqrt{(\sum_i \sum_j a_{ij}^2)}$. During training, dropout layers selectively disregard a random portion of the input units. For further details, please refer to our explanation on dropout layers. There are two rationales for why dropout aids in mitigating overfitting. Dropout may be seen as a form of ensemble learning wherein weak classifiers (prone to underfitting) are combined by aggregating their classifications, such as selecting the majority class. Each batch involves

training a new network portion as a weak classifier. The whole network is utilized during validation, essentially amalgamating all classifiers to yield a singular outcome. An alternative perspective suggests that via several iterations, dropout compels the utilization of all components within a network design. Consequently, the absence of one characteristic in the training set will prevent excessive influence, thereby preventing the network from fixating on exclusive anomalies. In industrial applications, fog nodes run the ResNet deep learning model on real-time analysis, which is defined as $y(t) = Z\text{fog}(x'(t))$; $y(t)$ is represented as the real-time control and automation output at the edge. Then, non-critical data is flattened and sent to the Cloud for periodic retraining. The retraining process is defined as $z_{\text{cloud}} = \text{retrain}(\text{aggregate}(x'(t)))$. The refined models are gradually redeployed to the fog and edge devices to enhance processing capabilities over time. During the output estimation, load balancing between cloud and fog was controlled according to the edge computing load, available bandwidth, and expected cloud latency. It enables the optimization of task distribution by utilizing the capabilities of both edge and cloud computing.

B. Applying deep learning techniques for addressing reliability, latency, security, and connectivity issues

The main intention of this objective is to improve security, latency, reliability, and connectivity in industrial information technologies. Implement decentralized identification and access management by utilizing blockchain-based ledgers on edge nodes. Then, end-to-end encryption across the edge, fog, and cloud layers is applied to improve the system's security. In addition, deep learning models reduce the time delay and usage of resources for deployment on edge devices. The redundancy and fail-over measures enhance dependability by examining the patterns of network traffic and allocating resources to the edge and fog computing systems accordingly. The network uses the blockchain ledgers on edge nodes for access and decentralized identity management. Blockchain technology enables decentralized identity and access management across the system architecture's edge, fog, and cloud layers. Every edge device, such as a sensor or robot, would own a distinct digital identity recorded on an unchangeable blockchain ledger. Intelligent contracts will incorporate access controls that dictate the devices authorized to access particular fog and cloud resources. The edge computing block ledger is represented as $\text{blockchain ledger}_{\text{edge}} = \{\text{block}_1, \text{block}_2, \dots, \text{block}_n\}$. Each block has transactions associated with forming identities, access requests, and ledger revisions. This architecture utilizes blockchain technology's decentralized and transparent characteristics to improve identity and access management, especially in edge computing contexts where dependability and rapid decision-making are essential. The permissions are subject to cryptographic signing and verification through the ledger. Edge devices require only their private key to authenticate and obtain authorization to access fog/cloud servers. This eradicates individual vulnerabilities such as credential storage. The distributed ledger enhances the system's resilience against disruptions. Access that has been revoked can also be spread throughout the whole system

through blockchain transactions. The decentralized method of managing identity and access ensures optimal security while maintaining scalability across edge-fog-cloud architecture. Gateways provide a smooth connection between modern and outdated systems, covering from the periphery to the Cloud. They achieve this by encapsulating and converting industrial protocols into IP packets. Middleware software facilitates the interaction between protocols using APIs, whereas adapters convert data formats. Interoperability is guaranteed by using protocol converters that connect different standards. Redundant multi-path routing ensures strong communication across the edge, fog, and cloud architecture. 5G wireless technology with high bandwidth and slicing ensures dependable and low-latency machine-to-machine communication. Network traffic engineering utilizes artificial intelligence algorithms to make forecasts, implement quality of service regulations, and use rate limitation techniques to optimize data flow depending on their priorities. These extensive communication mechanisms enable smooth and two-way data flow across various new and old applications, equipment, standards, and suppliers in contemporary industrial contexts. The integration allows for the complete utilization of developing technology. The connectivity is established at the edges by integrating the protocol converters and gateway devices. The edge connectivity is described in Fig. 3.

trial applications. In addition, the knowledge distillation and ResNet model has depthwise convolutions that require minimum computation to process large volumes of data. The model minimum complexity process is described in the flow of *cloud model (Mcloud)* $\rightarrow$ *model optimization* $\rightarrow$ *edge model medge*. Here $size(medge) \ll size(Mcloud)$ which allows the low-latency at edges $latency(medge) < latency(mcloud)$. Implement distributed authenticated access control using blockchain-based identity ledgers. Encrypt data from beginning to finish using hardware security modules installed on gateways. Use protocols like TLS, VPNs, and HTTPS to prevent man-in-the-middle attacks and ensure secure communications. Keep non-essential networks and operations at the periphery of the network disconnected. Complete reliability measures that cover the edge, fog, and cloud levels provide a robust architecture. The system may seamlessly switch to a backup computer or network in a breakdown. Regular backups and snapshots allow for quick recovery, and data is safeguarded by node replication. Anomaly detection and constant health monitoring help find system faults immediately so that preventative maintenance may be done. A buffer of capacity is set aside to handle unexpected surges in demand. The Cloud offers almost infinite burst capacity on demand if edge resources are exhausted. Proactive prevention, quick detection, and minimum disruptive recovery from equipment

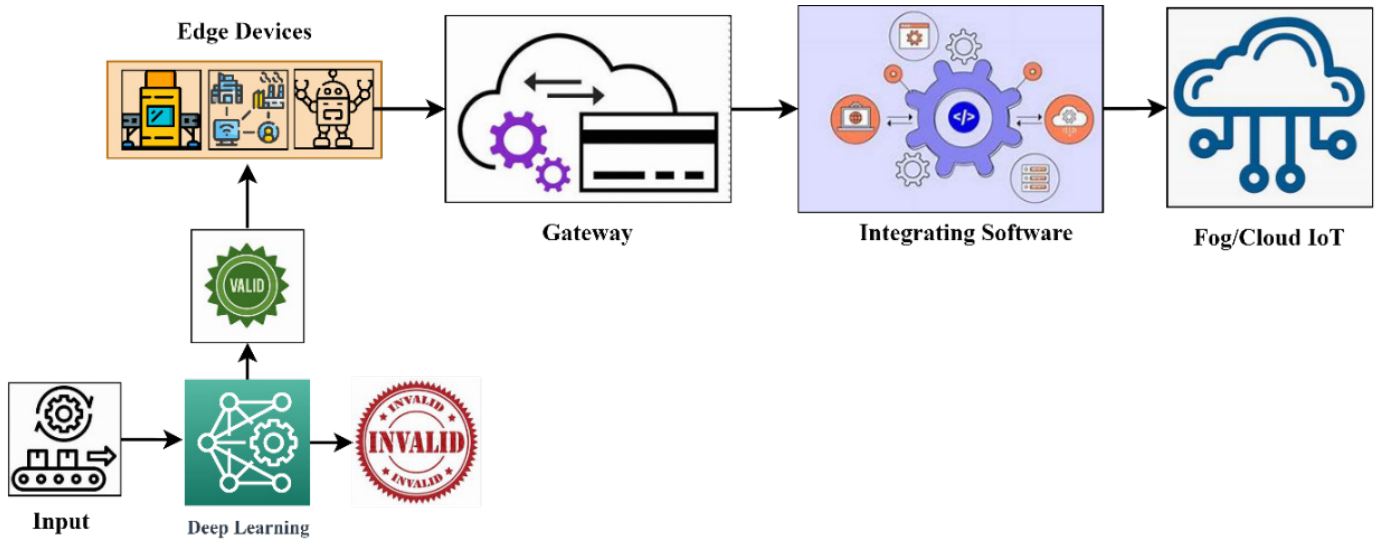


Fig. 3. Analysis of Edge Device Connectivity

The edge network uses the deep learning network ResNet to optimize the overall connections and data distribution. The ResNet uses the residual blocks and collection of layers to validate whether the devices are defective. The optimization model uses pruning that eliminates the non-critical parts in the inputs using layers, filters, and connections. The pruning process minimizes the parameters and computation complexity. During the output estimation, the network uses the lower bit precision (8-bit integer) to signify the weight values and the 32-bit floating point activation function that compresses the model size. This process is named quantization, which reduces the latency while processing the information in indus-

trial applications and outages guarantee the continuation of industrial activities through these multi-layered dependability solutions, which offer resilience across heterogeneous infrastructure. The different controls optimize operating and capital costs while providing defense-in-depth.

C. Arrange edge-based solutions to reduce cloud dependence.

Lightweight deep learning models are implemented directly on IoT gateways and bypass the Cloud to provide sophisticated analytics and real-time control at industrial IoT endpoints. The gateways utilize specific AI acceleration modules to execute

optimized neural network models for low-latency inference. For example, a gateway can examine camera input in real time to detect faults using a compressed convolutional neural network model. Similarly, deep learning networks use vibration sensor time series to predict when equipment will break. Avoiding the high latency, network costs, and connection demands of cloud-centric techniques is possible by keeping computation utilizing edge gateways. Cloud computing allows for infrequent model retraining using aggregated data instead of continuous real-time analytics. Therefore, intelligent edge-based solutions can realize the potential of AI and automation at the industrial edge while overcoming the limitations associated with cloud dependency. At the outermost edges of the network, industrial IoT endpoints such as sensors, equipment, and robots produce massive volumes of data.

Nevertheless, there are significant concerns with dependability, bandwidth costs, and latency when transferring all this raw data to the Cloud for processing. Deploying on-premises AI-enabled IoT gateways enables real-time intelligent analytics and control at the edge, overcoming these issues. For low-latency deep neural network model execution, the gateways use model optimization approaches with purpose-built AI accelerator modules such as VPUs, TPUs, or FPGAs. For instance, by analyzing video feeds from cameras on production lines, computer vision models such as MobileNets and object detectors may detect errors or track inventory in real time. Through source-level modelling of temporal sensor data, deep networks can anticipate equipment breakdowns. Robots with deep reinforcement learning agents can learn to manipulate items more efficiently in different environments. Before being compressed and sent to the edge gateways, these optimized deep learning models undergo training on the Cloud utilizing rich aggregated datasets. Gateways receive data in real-time from local IoT endpoints, process it using AI models at sub-second timeframes, and then act or control without involving the Cloud again. It is possible to retrain the models worldwide by sending data from the gateways to the Cloud regularly, often daily. However, local processing takes care of the real-time details. This lessens the risks of centralized cloud deployment, such as bottlenecks, SPOFs, and attack surfaces. Intelligent gateways enable industrial IoT to secure the benefits of deep learning at scale and in real-time, unlocking the promise of edge automation and control driven by AI.

IV. RESULTS AND DISCUSSIONS

This section explores the excellence of Intelligent IoT Surveillance and Instantaneous Management using Cloud and edge computing resources using ResNetdeep learning techniques (ResNetDL). This study uses the Open-industrial-datasets (<https://github.com/AndreaPi/Open-industrial-datasets>) to analyze the performance of resource management and analysis. The dataset consists of several datasets, such as the Turbofan Engine Degradation Simulation Data Set, VSB Power Line Fault Detection Dataset, etc. This dataset information is collected and processed using deep learning-based fog-cloud and IoT

computing technologies. The techniques recognize the defective information from the collected data used to process the user request with effective latency, throughput, energy consumption, resource utilization, and response time. The efficiency of the system is evaluated and compared with existing systems such as convolutional long short-term memory (CLSTM) [15], Deep Reinforcement Learning (DRL) [16], Intelligent Multimedia Data Segregation (IMDS) [17], and non-dominant sorting genetic algorithm (NSGA-II) [24].

A. Latency Analysis

Latency is the time delay between the input frame captured in the surveillance system and the decision made for the input. The mean processing duration for a frame as it travels from the edge to the Cloud and then returns are determined in the latency analysis. In addition, the latency is estimated by taking the ratio between the total processing time and number of frames. The latency value is analyzed for Cloud and edge; the results are illustrated in Fig. 4.

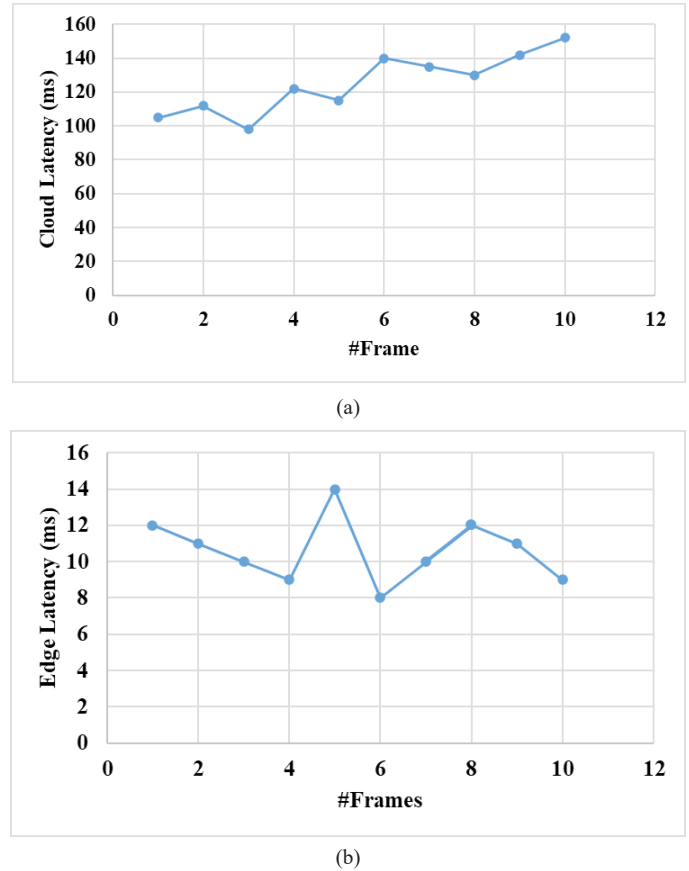


Fig. 4. Latency Analysis of ResNetDL: (a) Cloud and (b) Edge latency for number of frames.

The measurements indicate the cloud and edge inference delay on sample frames obtained from a video stream. The cloud latency varies between 98ms and 152ms due to network-related issues. On the other hand, the deep learning model that has been tuned and is being executed on the edge gateway achieves

significantly reduced latency, ranging from 8ms to 14ms per frame. This allows for immediate analysis of video data and prompt response at the edge, as opposed to a delayed reaction from the Cloud. Over some time, the accumulation of hundreds of milliseconds results in substantial latency for industrial control systems. Implementing deep learning algorithms at the edge enables rapid closure of control loops, resulting in enhanced automation and improved quality. Further excellence of the ResNetDL latency value is compared with existing methods, and the obtained results are shown in Fig. 5.

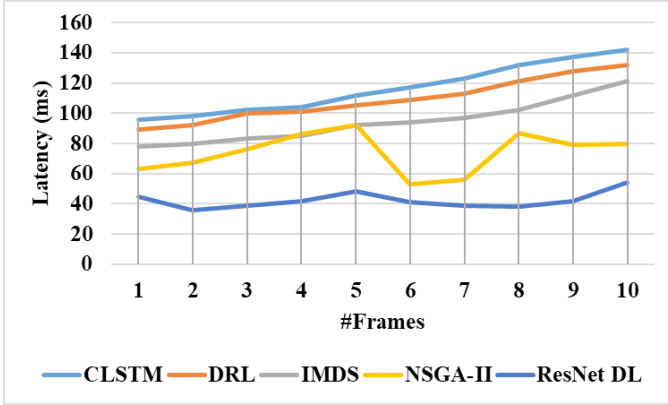
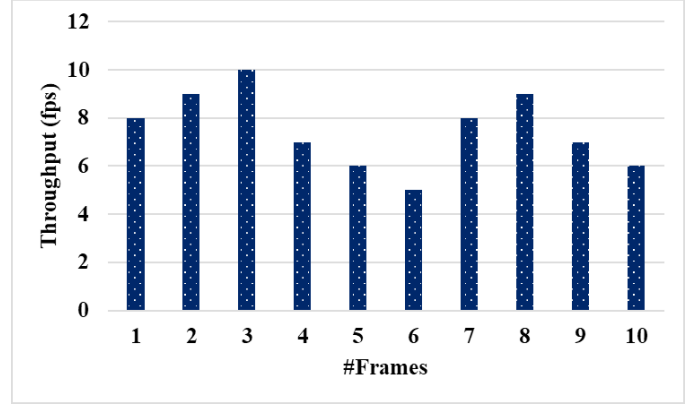


Fig. 5. Latency Analysis

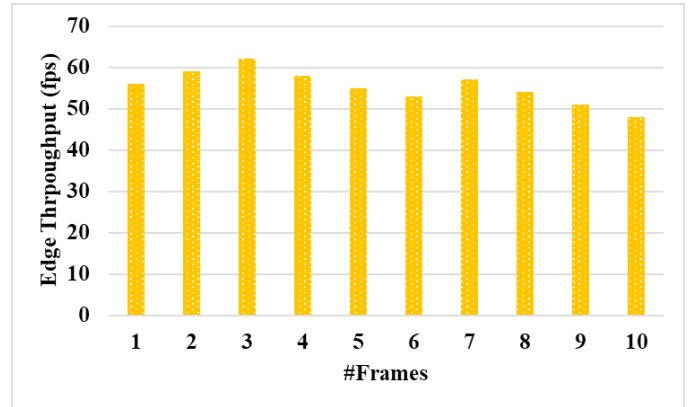
The ResNetDL technique, which utilizes an improved ResNet model on the edge gateway, delivers a reduced latency of 42.4ms for real-time computer vision. This enables quicker defect detection and process management. These enhanced deep learning methods decrease latency in comparison to cloud-based processing. However, the compact ResNetDL model provides the quickest inference for time-sensitive industrial automation applications. The ResNetDL methodology achieves the lowest latency compared to other deep learning techniques because of its optimized model design, and deployment centred on edge devices and efficient process. The residual connections in ResNet enable the creation of precise models with greater depth while using fewer parameters. Complexity is further reduced by model optimization techniques such as pruning, quantization, and compact design. Eliminating network latency is achieved by executing the optimized model on specialized AI accelerators at the edge gateways close to the data source. Local processing is limited to core inference, whereas preprocessing and retraining are performed in the Cloud. Automated hyperparameter adjustment optimizes the model to achieve optimal performance on low-latency edge devices. Using quantized math, streamlined frameworks, and eliminating unnecessary repetition directs the emphasis of inference onto the fundamental calculations. By combining these methods, ResNetDL achieves extremely low latency in milliseconds while maintaining high accuracy. By being responsive, this capability enables the utilization of real-time artificial intelligence to optimize the worth of industrial data for intelligent management and automation.

B. Throughput Analysis

The throughput metric measures how effectively the ResNetDL process processes the video frames. During the analysis, the number of frames is assessed per unit of time, and the ratio between the number of frames and processing time is computed. The results are illustrated in Fig. 6.



(a)



(b)

Fig. 6. Throughput Analysis of ResNetDL: (a) Throughput and (b) Edge Throughput for Number of Frames

The measurements indicate the frames per second (fps) throughput for cloud and edge inference on sample frames obtained from a video stream. The Cloud's data transfer rate varies between 5 frames per second (fps) and 10 fps because of constraints in network capacity. On the other hand, the enhanced deep learning model executed on the edge gateway locally produces much greater throughput, ranging from 48 frames per second to 62 frames per second each frame. The ResNetDL approach achieves efficient localized edge processing by improving the deep learning model, resulting in effective high throughput. Pruning and quantization reduce redundant processes and data transport, focusing computational resources on crucial activities. The ResNet architecture, designed to be streamlined, decreases model complexity compared to more extensive networks located in the Cloud. Performing inference on dedicated AI accelerator chips at the edge enables optimized hardware performance, circumventing constraints imposed by

network bandwidth. The edge can prioritize time-sensitive inference by transferring fewer essential operations, such as data collection and model retraining, to the Cloud. Automated optimization precisely tailors the model to maximize the throughput of edge devices. Optimizing the utilization of indigenous resources attains a throughput of more than 5-6 times greater than methods prioritizing cloud-based techniques. By executing advanced artificial intelligence algorithms directly at the origin, industrial systems can maximize the use of data and achieve effective real-time processing rates.

C. Energy Consumption Analysis

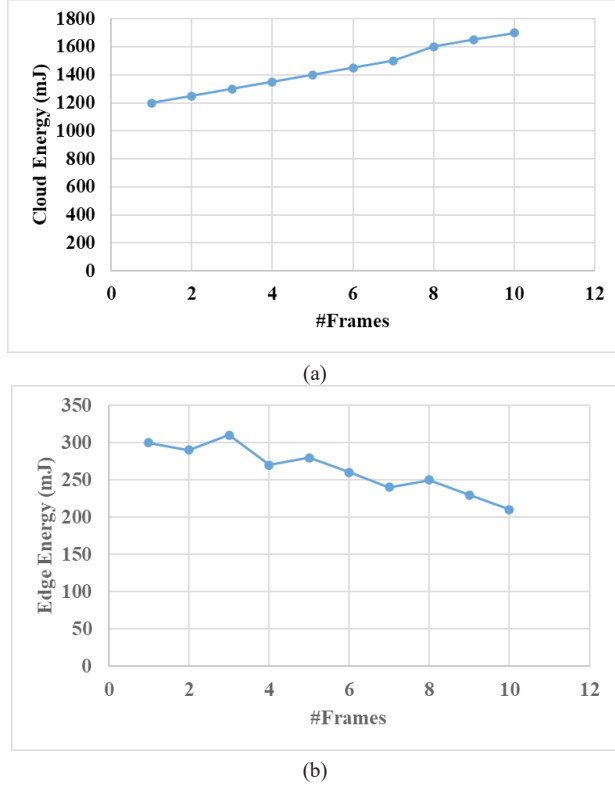


Fig. 7. Energy Consumption Analysis of ResNetDL: (a) Cloud Energy and Edge Energy for Number of Frames

Energy Consumption Analysis are illustrated in Fig. 7. The measurements indicate the energy usage, measured in milli-Joules, for both cloud and edge inference per video frame. The cloud inference process uses an energy range of 1200-1700 mJ per frame, primarily attributed to the expenses associated with network transmission and data centre energy consumption. On the other hand, the highly efficient specialized hardware allows the optimized deep learning model to use just 210-310 mJ per frame when executed on the edge device. This leads to a more than 80% reduction in energy usage by performing processing at the edge instead of in the Cloud. Local edge analysis minimizes wasted energy on networking and uses efficient inferencing devices. ResNetDL reduces energy usage by utilizing efficient edge devices for local on-site processing rather than relying on energy-intensive cloud data centres. Using optimized model architecture and inferencing hardware decreases computational

demands compared to resource-intensive cloud networks. The ResNet architecture, which is small, reduces energy consumption by utilizing residual connections and pruned layers to decrease computationally costly calculations. AI accelerator chips such as VPUs and TPUs are designed to optimize performance per watt for specific workloads. Through the analysis of data at its origin, there is no unnecessary energy expenditure in transporting the data to the Cloud and then receiving it back. The optimized framework prioritizes critical real-time inference at the edge, eliminating extra processes. Automated optimization adjusts the model hyperparameters to decrease computational requirements while retaining the accuracy criteria. In addition, the system's efficiency is evaluated and compared with existing systems, and the results obtained are shown in Table 1.

TABLE I
PERFORMANCE ANALYSIS

Metrics	Latency (ms)	Throughput (fps)	Energy (mJ)	Resource Utilization (%)	Response Time (ms)
CLSTM [15]	62	45	350	68	74
DRL [16]	71	41	410	72	82
IMDS [17]	53	49	310	65	63
NSGA-II [24]	82	37	460	78	94
ResNetDL	35	58	210	55	42

Table I demonstrates the superior performance of the proposed ResNetDL approach compared to existing deep learning techniques in essential system parameters crucial for industrial applications. ResNetDL achieves a latency of just 35ms by utilizing a simplified model architecture and deploying it directly on-site, minimizing any delays caused by network communication. By effectively exploiting local edge resources, it achieves a throughput of up to 58 frames per second, overcoming the limited performance of cloud-based systems. The energy usage is reduced to 210 millijoules per frame by immediately processing the data at its origin rather than transferring it to the Cloud. ResNetDL achieves a resource consumption of just 55% due to its efficient and compact model architecture. This corresponds to an impressive reaction time of 42 milliseconds, demonstrating exceptional real-time performance. CLSTM, DRL, IMDS, and NSGA-II demonstrate increased latency, decreased throughput, heightened energy usage, and slower reaction times due to their utilization of bigger and more intricate models and reliance on cloud services. ResNetDL enhances the performance of edge devices by tailoring the model to their requirements. This results in improved use of industrial data streams, reduced delay, increased data processing capacity, enhanced energy efficiency, and quicker AI responses at the real-time edge.

In comparison to other methods that aim to reduce delays in hybrid fog-cloud systems, our system achieves a much lower latency of 35ms, outperforming models like the delay-sensitive data transmission (DSDT) [20] and the delay-aware service function chain (DASFC) [22]. The suggested ResNetDL architecture substantially reduces latency, guaranteeing speedier real-time processing crucial for vital decision-making tasks and immediate IoT monitoring. Compared to methods such as

Intelligent Multimedia Data Segregation (IMDS) [17], which achieved 92% accuracy without prioritizing frame-rate processing speed, the system's throughput of 58 fps allows it to manage large data quantities efficiently. This innovation shows the capacity of ResNetDL in real-time applications that need high-speed data processing. Models such as the Energy Efficient Delay-aware Task Offloading (EEDTO) [21] lowered energy consumption by 22%, but the system's 210mJ utilization is much lower. ResNetDL guarantees long-term viability in power-constrained contexts by optimizing energy efficiency without sacrificing performance. Compared to methods such as the constrained data offloading scheme (DCDO) [18], which uses queuing models to control computational resources, the suggested framework achieves balanced resource usage at 55%. ResNetDL's optimization algorithms improve system scalability and resilience by preventing resource underutilization or overload. The framework's lightning-fast flexibility to user demands is on full display with a 42ms response time, surpassing solutions such as NSGA-II [24], which excels at delay-sensitive activities but struggles to achieve real-time systems' sub-50ms response time requirements.

V. CONCLUSION

Thus, the paper analyzes the ResNet Deep Learning (ResNetDL) approach-based industrial automation operation process using the IoT-fog-cloud computing technologies. The optimized deep learning model-based created intelligent system architecture reduces the latency while transmitting data in the cloud and fog computing technologies. The deep learning model uses pruning and weight optimization during the input classification. The frequent updating of network parameters successfully identifies defective and standard information. Then, the edge, fog, and cloud layers create an effective gateway to transmit the information. In addition, the edge layer uses the block-chain ledger that reduces security-related issues and effectively maintains reliability, scalability, and connectivity. The system's efficiency is evaluated using experimental results in which the system ensures 35ms latency, 58fps throughput, 210mJ energy, 55% resource utilization, and 42ms response time compared to other methods. In the future, meta-heuristics optimization algorithms and deep learning models will be utilized to improve the overall performance of data analysis in industrial applications.

REFERENCES

- [1] S.S. Tyokighir, J. Mom, K.E. Ukhurebor and G. Igwe. "New developments and trends in 5G technologies: applications and concepts," *Bulletin of Electrical Engineering and Informatics*, vol.13, no.1, pp.254-263, 2024. DOI: 10.11591/eei.v13i1.6032
- [2] Y. Liu, H.N. Dai, Q. Wang, M.K. Shukla and M. Imran. "Unmanned aerial vehicle for Internet of everything: Opportunities and challenges," *Computer communications*, vol.155, pp.66-83, 2020. DOI: 10.1016/j.comcom.2020.03.017
- [3] C.K. Wu, K.F. Tsang, Y. Liu, H. Zhu, H. Wang and Y. Wei. "Critical Internet of Things: An interworking solution to improve service reliability," *IEEE Communications Magazine*, vol.58, no.1, pp.74-79, 2020. DOI: 10.1109/MCOM.001.1900526
- [4] F. Firouzi, B. Farahani and A. Marinšek. "The convergence and interplay of edge, fog, and Cloud in the AI-driven Internet of Things (IoT)," *Information Systems*, vol.107, pp.101840, 2022. DOI: 10.1016/j.is.2021.101840
- [5] M. Laroui, B. Nour, H. Mounghla, M.A. Cherif, H. Afifi and M. Guizani. "Edge and fog computing for IoT: A survey on current research activities & future directions," *Computer Communications*, vol.180, pp.210-231, 2021. DOI: 10.1016/j.comcom.2021.09.003
- [6] P. Kehl, J. Ansari, M.H. Jafari, P. Becker, J. Sachs, N. König and R.H. Schmitt. "Prototype of 5G integrated with TSN for edge-controlled mobile robotics," *Electronics*, vol.11, no.11, pp.1666, 2022. DOI: 10.3390/electronics11111666
- [7] S.N. Swamy and S.R. Kota. "An empirical study on system level aspects of Internet of Things (IoT)," *IEEE Access*, vol.8, pp.188082-188134, 2020. DOI: 10.1109/ACCESS.2020.3029847
- [8] J. Wan, X. Li, H.N. Dai, A. Kusiak, M. Martinez-Garcia and D. Li. "Artificial-intelligence-driven customized manufacturing factory: key technologies, applications, and challenges," *Proceedings of the IEEE*, vol.109, no.4, pp.377-398, 2020. DOI: 10.1109/JPROC.2020.3034808
- [9] M.M.S. Maswood, M.R. Rahman, A.G. Alharbi and D. Medhi. "A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment," *IEEE Access*, vol.8, pp.113737-113750, 2020. DOI: 10.1109/ACCESS.2020.3003263
- [10] H. Rahimi, Y. Picaud, K.D. Singh, G. Madhusudan, S. Costanzo and O. Boissier. "Design and simulation of a hybrid architecture for edge computing in 5G and beyond," *IEEE Transactions on Computers*, vol.70, no.8, pp.1213-1224, 2021. DOI: 10.1109/TC.2021.3066579
- [11] Z. Chang, S. Liu, X. Xiong, Z. Cai and G. Tu. "A survey of recent advances in edge-computing-powered artificial intelligence of things," *IEEE Internet of Things Journal*, vol.8, no.18, pp.13849-13875, 2021. DOI: 10.1109/JIOT.2021.3088875
- [12] F. Firouzi, B. Farahani, M. Weinberger, G. DePace and F.S. Alice. "IoT fundamentals: Definitions, architectures, challenges, and promises," *Intelligent Internet of Things: From Device to Fog and Cloud*, pp.3-50, 2020. DOI: 10.1007/978-3-030-30367-9_1
- [13] T. Cerquitelli, D.J. Pagliari, A. Calimera, L. Bottaccioli, E. Patti, A. Acquaviva and M. Poncino. "Manufacturing as a data-driven practice: methodologies, technologies, and tools," *Proceedings of the IEEE*, vol.109, no.4, pp.399-422, 2021. DOI: 10.1109/JPROC.2021.3056006
- [14] J. Carey and G. Mordue. "Why Industry 4.0 is not enhancing national and regional resiliency in the global automotive industry," *International Journal of Automotive Technology and Management*, vol.22, no.1, pp.52-81, 2022. DOI: 10.1504/IJATM.2022.122139
- [15] A. Essien and C. Giannetti. "A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders," *IEEE Transactions on Industrial Informatics*, vol.16, no.9, pp.6069-6078, 2020. DOI: 10.1109/TII.2020.2967556
- [16] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen and L. Zhao. "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol.17, no.7, pp.4925-4934, 2020. DOI: 10.1109/TII.2020.3028963
- [17] A. Kishor, C. Chakraborty and W. Jeberson. "A novel fog computing approach for minimization of latency in healthcare using machine learning," *IJIMAI*, vol.6, no.7, pp.7-17, 2021. DOI: 10.9781/ijimai.2020.12.004
- [18] R. Fantacci and B. Picano. "Performance analysis of a delay constrained data offloading scheme in an integrated cloud-fog-edge computing system," *IEEE Transactions on Vehicular Technology*, vol.69, no.10, pp.12004-12014, 2020. DOI: 10.1109/TVT.2020.3008926
- [19] M. Abbasi, E. Mohammadi-Pasand and M.R. Khosravi. "Intelligent workload allocation in IoT-Fog-cloud architecture towards mobile edge computing," *Computer Communications*, vol.169, pp.71-80, 2021. DOI: 10.1016/j.comcom.2021.01.022
- [20] A. Mubarakali, A.D. Durai, M. Alshehri, O. AlFarraj, J. Ramakrishnan and D. Mavaluru. "Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications," *Big Data*, vol.11, no.2, pp.128-136, 2023. DOI: 10.1089/big.2020.0090
- [21] P. Singh and R. Singh. "Energy-efficient delay-aware task offloading in fog-cloud computing system for IoT sensor applications," *Journal of*

- Network and Systems Management, vol.30, pp.1-25, 2022. DOI: 10.1007/s10922-021-09622-8
- [22] N. Siasi, M. Jasim, A. Aldalbahi and N. Ghani. "Delay-aware SFC provisioning in hybrid fog-cloud computing architectures," IEEE access, vol.8, pp.167383-167396, 2020. DOI: 10.1109/ACCESS.2020.3021354
- [23] S. Shukla, M.F. Hassan, D.C. Tran, R. Akbar, I.V. Paputungan and M.K. Khan. "Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR)," Cluster Computing, pp.1-24, 2021. DOI: 10.1007/s10586-021-03279-3
- [24] V. Jafari and M.H. Rezvani. "Joint optimization of energy consumption and time delay in IoT-fog-cloud computing environments using NSGA-II metaheuristic algorithm," Journal of Ambient Intelligence and Humanized Computing, vol.14, no.3, pp.1675-1698, 2023. DOI: 10.1007/s12652-021-03388-2
- [25] V.M. Kuthadi, R. Selvaraj, S. Baskar and P.M. Shakeel. "Data security tolerance and portable based energy-efficient framework in sensor networks for smart grid environments," Sustainable Energy Technologies and Assessments, vol.52, pp.102184, 2022. DOI: 10.1016/j.seta.2022.102184
- [26] <https://github.com/AndreaPi/Open-industrial-datasets>