Privacy-Preserving Learning Model using Lightweight Encryption for Visual Sensing Industrial IoT Devices

B D Deebak, Seong Oun Hwang, *Senior Member, IEEE*

***Abstract*—Technological convergence in visual sensing with industrial IoT (VSI-IoT) can bring numerous advances to large-scale crowd management systems like visual crowdsensing. VSI-IoT has significant features, including sensing, computing, analyzing, and storing, to address the issues of bearing failures, such as unplanned outages, increased downtime, and reduced operational efficiency. By contrast, providing privacy to the IIoT environments is a challenging task. Thus, this paper presents a novel privacy-preserving learning (PPL) mechanism that senses the defect rate of bearing failures using lightweight model aggregation at edge computing systems to preserve the privacy features. This convergence model synthesizes shape features comprehensively to transform the feature vectors into predictive functions that examine the categorization models using a two-dimensional convolution neural network (2D-CNN). Using security analysis, we demonstrate that the proposed PPL can achieve better privacy protection and model accuracy to preserve the learning features without additional verifiability. Further, the examination results showed that the proposed 2D-CNN with BN and LN consumed less computation complexity to achieve better detection accuracy ($\approx$87.91.9% to $\approx$99.98%) and communication cost ($\approx$21.09*MB* to 23.92*MB*) over three bearing datasets (i.e., IMS-Rexnord, CWRU, and Paderborn) than other state-of-the-art approaches. Above all, the privacy preserving based AlexNet was implemented using CryptoNet and LoLa to show different sets of efficiencies such as processing time, privacy, and integrity checks to preserve system performance following time-sensitive application scenarios like supply-chain optimization.**

***Index Terms*—Industrial Internet of Things, Visual Sensing, Machine-To-Machine, Privacy-Preserving, Convolution Neural Network, Communication Cost.**

I. Introduction

IIoT applications exploit the core features of global networks to interrelate the connectivity of physical devices. It is an essential component of networking devices to outsource the massive amount of industrial data in smart manufacturing systems. Interconnected devices can collect data from various machinery operations, such as visual objects, advanced analytics, and cloud computing, to improve the quality of the desired process. Each process enables a better data transmission rate over a dedicated network to form system intelligence. Intelligent systems converge various technological assets, such as IoT, big data, machine learning, and cyber-physical systems, to handle massive industrial data. However, these systems can not organize several smart operations, such as sensing, process control, augmented workforce, process twin, and synchronization, to observe special events or occurrences of faults in a real-time manufacturing system. Ali et al. [1] designed a knowledge-based failure detection and prevention (K-FDP) model to eliminate production failure and reduce maintenance costs.

In K-FDP, the systems' main functionalities that detect manufacturing process changes include data collection and processing from various sensors [2]. However, automation processing discovers security threats in networking systems or industrial applications. Particularly, in VSI-IoT, the end-to-end processing controller relies on programmable devices and tools to leverage the sensing mechanism and flexible automation. As a result of this, device protection conducive to data consolidation and extraction demands proper privacy preservation to secure communication at the cloud center. The comprehensive studies have made it clear that encryption-based modeling with privacy preservation can enable efficient communication with massive visual sensing units to guarantee a high level of correctness and accuracy in any industrial network [3]. In consequence of this, visual sensing has attracted the researcher's attention to the advancements in IoT systems, including sensing, monitoring, tracking, and processing. However, to address privacy issues in data modeling and analysis, most existing studies have considered cryptography techniques such as differential privacy (DP) and homomorphic encryption (HE).

Pong et al. [4] confided in the HE system to secure the shared gradient in their deep learning system. Contrarily, the encryption based on this strategy incurs more computation and communication costs, potentially restricting its usage in industrial systems. Zhao et al. [5] exploited privacy technologies to take an industrial action between privacy preservation and modeling accuracy. Unfortunately, the methods utilized in the technologies cannot offer an acceptable trade-off when evaluated with complex learning models to intricate predictable patterns and relationships. In another work, Park and Lim [6] introduced federated learning-based privacy preserving (FLPP)) using homomorphic encryption to protect the application system against inference attacks. Regardless, the FLPP cannot gain a model with better interpretability to protect the target system against security threats while training the model with real-time test cases. To tackle various issues related to sensing and analyzing storage data, privacy-preserving categorization has been studied extensively in the past few years [7].

Owing to the trade-off efficiency between privacy and the usability of storage data [8], [9], the existing privacy-preserving mechanisms relied on third-party servers, which are undependable due to network and system vulnerabilities. Because industrial applications exploit IoT devices via third-party servers to collect and analyze sensitive data, protecting their data privacy demands a data-centric solution to classify the behavior of data using distributed learning with deep learning (DL) framework [10]. To process a large amount of industrial data using centralized learning and enable data sources to aggregate the hyperparameters to infer the behavior of the classification algorithms, most IoT devices employ edge intelligence. As a result of the successful representation model using behavioral classification, efficient IoT devices with limited computational resources involve complex convolution neural networks (CNNs) to operate the inference tasks associated with visual sensing units. Most existing works deal with cryptographic techniques with lightweight encryption to determine the scalar product in order to support neural network interference. However, transformation in neural networks causes a certain impairment in achieving inference accuracy due to discretized neural networks.

*Corresponding Authors (Seong Oun Hwang)

This work was supported by the Brain Pool Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and Information Communication Technology (ICT) under Grants (2022H1D3A2A02081848, RS-2024-00340882).

B D Deebak and Seong Oun Hwang are with the Department of Computer Engineering, Gachon Univesity, Gachon University, Seongnam 13120, South Korea e-mail: (deebak@gachon.ac.kr and sohwang@gachon.ac.kr).

A. Motivation

New challenges are arising in learning models with the technological growth of visual sensing in industrial IoT devices but not limited to privacy preserving. Because VSI-IoT has sensing and processing capabilities to converge distributed computing services with a visual sensor network to compute non-linear activation functions in large-scale application systems. Precisely, the research in artificial intelligence uses machine learning as a service (MLaaS) to perform extensive visual or image classification based on query processing to guarantee more precise activation in industrial application systems (automation and medical diagnosis). The system can constitute the semantics and contexts of the sensing devices with IIoT devices to provide high-level services. Alternatively, audio-visual sensor data is always unstructured and requires additional computation and processing power to extract real-time information. In consequence, interrogative cloud computing prefers proper training models and classification like CNN to serve a reliant inference on extensive computational data.

Zhang et al. [11] devised a privacy preserving neural network scheme (PPNNS) using proxy re-encryption and other key management systems to achieve a secure activation function. Despite this, the threat model applied in PPNNS cannot presuppose its pooling protocol with honest users to compute the activation function under ciphertext operation. Liu et al. [12] created a secure network inference system (SNIS) to preserve the private data stored in the cloud using a proprietary CNN model. Conversely, the SNIS neglects privacy preserving on the trained model and action data to endure better CNN inference since it cannot apply homomorphic encryption to leverage the computing tasks without associated data owners. Therefore, we precisely develop a privacy-preserving learning (PPL) model to manage a large set of applications indispensable to categorizing applications into different functionalities. The model can benefit application users, developers, and stores. As another option, securing privacy in VSI-IoT is still a challenging task to perform network analysis. For this reason, In this study, the PPL approach adopts four basic principles:

- Before uploading the applications, the developers assign a suitable category to the policy of the app store to achieve data protection and event classification;
- Upon the successful categorization of unshared data, the administrator reviews the documentation manually to verify the assignment strategies and computation resources of the learning models; and
- Considering the high-level visual factors via convolution neural network and semantic factors close to feature statistics, the PPL approach makes independent contributions to determine the factors influencing the service quality of IMS-Rexnord Bearing via a private identifier of each class.
- Achieving trade-off efficiency among the degree of security and privacy, the training applicant, i.e., the IoT edge, builds a robust network using edge computing that recreates a collaborative architecture to distinguish its private data while analyzing the bearing dataset.

B. Research Contribution

The proposed PPL prefers user devices to examine the collected data, which can easily expose the confidential information of the users. Therefore, technical strategies, such as segmentation and lightweight model aggregation, are employed to anonymize the data access. The major contributions are as follows:

1) We design a two-dimensional convolution neural network (2D-CNN) with binary and layer normalization to correlate the application functions and usage pattern, which can categorize the machinery defects precisely to analyze various beaming failures, such as failure at the inner race (F-IR), failure at the outer race (F-OR), and failure at the rolling element (F-RE).
2) We utilize an optimized PPL with lightweight aggregation to train the models on the edge computing system, which preserves the privacy between the edge device and server and permits the shared data to access the local aggregation to minimize communication cost-effectively.
3) We apply the extraction methods, including segmentation learning and symbolic-aggregate approximation for usage and shape features, which measure the data sequence in terms of the inner race, outer race, and rolling element with a suitable time interval.
4) We use lightweight homomorphic encryption to protect the PPL over the honest but curious parameters, which prevent the computing server from accessing the sensitive information of the edge devices.
5) We train the large-scale data (i.e., IMS-Rexnord, CWRU, and Paderborn) using the proposed 2D-CNN with batch normalization (BN) and layer normalization (LN) and other state-of-the-art approaches to analyzing the defects, which evaluate their metrics, such as accuracy, F1-Score, precision, recall, and communication cost, through preserving the privacy features.
6) We design a privacy preserving based AlexNet using the prediction models namely CryptoNet and LoLa to analyze the performance of the proposed 2D-CNN with lightweight encryption in order to assess a few significant metrics such as processing time, accuracy, latency, and privacy protection.

Section II discusses a generic architecture of the VSI-IoT with middleware integration and critical issues in privacy preserving classification and holomorphic encryption to represent the flow structures of the industrial systems including challenges addressed by holomorphic cryptosystems. Section III presents a VSI-IoT framework with privacy-preserving learning using 2D-CNN, which discusses the technical design, device configuration, threat model, model aggregation, lightweight encryption, and proposed 2D-CNN, including batch and layer normalization to handle the machinery defects. Section IV shows a formal security analysis using learning-based machinery systems and CPA to guarantee better verifiability with privacy protection. Section V demonstrates the evaluation results of various CNN models, such as proposed 2D-CNN and other state-of-the-art approaches. Section VI discusses the limitation issues of privacy preserving model using homomorphic encryption. Section VII summarizes the research work.

II. Background Research

This section discusses a generic architecture of the VSI-IoT to represent the flow structures of a multi-layer environment, including visual objects, middleware, and cloud application services.

A. VSI-IoT Architecture

IoT integrates intelligent components, including Device-To-Device (D2D) and visual systems, which provide connectivity between the smart devices in energy-efficient vehicular communication, homing systems, wearable systems, and smart building [20]. The system can support several applications to explore key functions, such as group interaction, administration, searching, and information retrieval. Since the VSI-IoT connectivity between physical devices has increased exponentially, the connected systems require mass storage capacity and computing power to process a massive amount of real-time data. To store and process massive data, cloud computing (CC) and edge computing (EC) have emerged as promising technological solutions in IIoT. Fig.1 shows the VSI-IoT architecture with the component views of middleware integration. The architecture supports several

TABLE I: Comparing the Strength and Limitation of Existing Privacy Preserving Learning Models.

Learning Model	Applied Technique	Strength ✅ and Limitation ❌
SAF [13]	Privacy Preserving with Multi-Party Computation	✅leverage the computation task without additional communication overheads. ❌consumes more computation efficiency to record the modeling data.
ELE [14]	Privacy Preserving with Lightweight Encryption	✅explores the energy-saving effect (E-SE) to improve the usage rate of users' equipment. ❌cannot preserve the confidentiality of the learning models due to the hidden risk of information leakage.
F-EDAM [15]	One-Time-Pad Homomorphic Technique with Data Aggregation	✅reduce the computation costs among the other real-time entities. ❌requires a few extra rounds of communication to execute the data aggregation scheme on networks.
FLS [16]	Federated Learning with Privacy Preserving	✅use holomorphic encryption to protect the modeling parameters of IoT devices. ❌cannot securely derive classification outcomes by using fully trained parameters.
EFL-BN [17]	Federated Learning with Batch Normalization	✅alleviate domain shifting without sharing sensitive data in the local feature distribution. ❌cannot securely derive classification outcomes by using fully trained parameters.
EC-FLM [18]	Edge Computing Enabled Federated Learning Model	✅integrate differential privacy with partition technique to protect the privacy of the learning model. ❌cannot reduce the privacy leak while increasing the transmission efficiency of the model.
UDLM [19]	Unified Deep Learning Model with Privacy Preserving	✅resist the inference attack against the learning models. ❌cannot protect the learning model against collusion attacks.

integrative solutions, such as security, safety, and privacy, to offer provisional services through a dedicated distributed system. The systematic flow transmissions of VSI-IoT architecture are as follows.

1) *Web of the Internet - Access Point* bridges a standalone unit within the local area network to communicate with an increasing range of connected devices wirelessly.
2) *Cloud Database - Data Center* enables numerous IoT applications to use as a storage computing machine to offload their voluminous processing and sharing capabilities.
3) *Web Server* provides infrastructure management for sensors, actuators, and resource access business models to deliver numerous services to modern automation systems.
4) *IoT Database - QoS Policy [Device Status]* shows the current state of the computing device to perform data exchange in order to prevent packet drop.
5) *IoT Database - QoS Policy [Publisher-Subscriber Model]* manages the complexity of point-to-point connect based on reading/writing operation to offer better device interaction with event-driven architecture.
6) *Homing Systems - Home Sensing Units* use their versatile features to read the parameters, such as humidity, pressure, gyroscope, object temperature, and GPS location.
7) *Wearable Systems - Wearable Sensing Units* utilize a short-span communication protocol, such as radio-frequency identification, near-field communication, and Bluetooth low energy (BLE), to test the smart sensing tag.
8) *Smart Building - Smoke Sensor/Door Sensing* include machine-to-machine communication to explore the features of the computing devices that establish communication between sensing devices and cloud services to initiate cautious action.
9) *Vehicular Systems - Motion/Light Sensing* have a state of linkability or unlinkability to broadcast the traceability path of a new computing device.

The hosted application services offer event-based data-centric communication to offer better interactivity between machines and infrastructure. Lastly, the architecture uses the middleware controller to understand the core activities of the publish-subscribe model, including dissemination, subscription, and push notification.

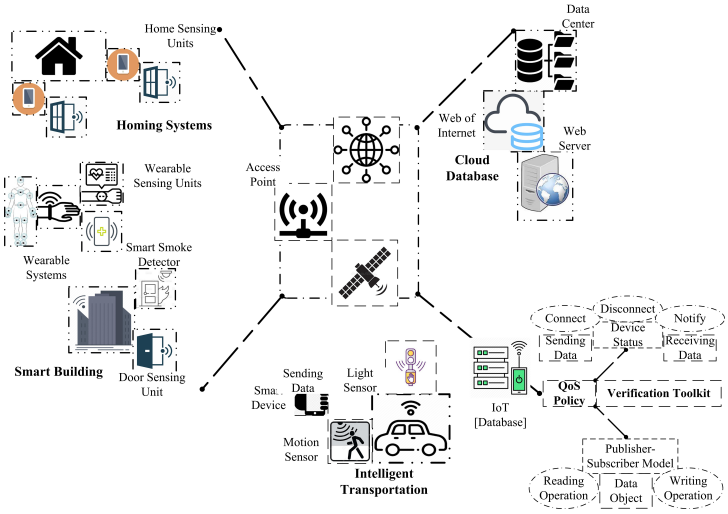


Fig. 1: Architecture of VSI-IoT with Component View of Middleware Integration.

B. Related Work

In this section, we discuss three significant phases namely privacy preserving, classification, and lightweight encryption to highlight their challenges while representing the activation function across a large-scale dataset contributed by several third parties.

Critical Issues in Privacy Preserving Classification and Homomorphic Encryption: Preserving the servers' learning models or users' private keys necessitates cryptographic techniques and activation functions to validate suboptimal performance while deforming different activation methods with lightweight encryption. Li et al. [13] constructed a server-assisted framework (SAF) to analyze the core structures of a non-interactive privacy preserving learning model with multi-party computation. However, the efficiency of the SAF is considerably low to satisfy the requirement of practical applications. Tian et al. [14] employed efficient lightweight encryption (ELE) to outsource the CNN inference privately. Despite that, the ELE scheme cannot preserve the confidentiality of the learning models due to the hidden risk of information leakage. Lyu et al. [15] applied a one-time-pad homomorphic technique to design an efficient fog-enabled data aggregation method (F-EDAM) with privacy preserving.

The F-EDAM generates a new key to match with message length which can unconditionally reduce the computation costs among the other real-time entities (fog, control center, and trusted authority). Despite this fact, the F-EDAM requires a few extra rounds of communication to execute the data aggregation scheme on networks [21] As a result, conventional strategies (support vector machine and artificial neural networks) capture all the properties of device activities into statistical features to investigate extracted data of multi-sensor systems. The study in [22] has discovered a hierarchical learning method after integrating its technique with a hidden Markov model. In order to stimulate the features of the time and frequency domain, the extended works have utilized a least-square support vector machine, resulting in the mitigation of privacy risk on the connected systems. Of late, the existing works have signified the use of recurrent neural networks

(RNNs) in classifying the temporal relation of multiple datasets with 2D data representation to train the model based on local data.

Yuanhang et al. [23] intended to devise a flow prediction model (FPM) using blockchain-based federated learning. Nevertheless, the FPM cannot prevent data poisoning attacks as it fails to conduct credible data sharing with collected user information. Shili et al. [24] developed crowdsourcing federated learning (CFL) to train the neural model with blockchain networks. The CFL utilizes a proxy re-encryption technique to achieve the property of privacy preservation. However, this learning method cannot machinate with the distribution mechanism fairly to enhance the credibility of parameter protection in the crowdsourcing platforms. Zhou et al. [16] created a federated learning scheme (FLS) with privacy preserving based on holomorphic encryption to protect the modeling parameters of IoT devices. Despite this, the FLS cannot protect the learning model against collusion attacks to prevent the servers from malicious nodes.

Li et al. [17] presented effective federated learning with batch normalization (EFL-BN) to alleviate domain shifting without sharing sensitive data in the local feature distribution. Regardless of this, the EFL-BN cannot securely derive classification outcomes by using fully trained parameters to probe the privacy of neural networks due to the direct disclosure of server training models. Zhang et al. [18] proposed edge computing enabled federated learning model (EC-FLM) to integrate differential privacy with partition technique. Conversely, the EC-FLM cannot reduce the privacy leak concerns while the modeling parameter exchanged between the participants and the central server to retrieve target information. Liu et al. [19] introduced a unified deep learning model (UDLM) with privacy preserving to prevent the leakage of user privacy in edge computing. The UDLM synchronizes federated learning with edge computing to keep the modeling data at the local storage of the edge device to resist the inference attack against the learning models.

However, the UDLM cannot obtain the aggregated model at the edge to exchange the computational parameters with the training process in order to preserve user privacy. The detailed analysis shown in Table I demonstrates that preventing unauthorized data access during computation demands confidential computing with a trusted environment to protect malicious access against applications or cloud service providers. To perform computation on encrypted data without decryption, lightweight homomorphic encryption as a cryptographic technique is applied to modern industrial applications. Since the application can compare data handling and security to maintain data locally with shared modeling parameters, the hardware-based solution (Intel SGX and Intel TEE) is not required to access the sensitive data during the training process as it cannot integrate with distributed learning to minimize high communication overheads. Fortunately, the applied strategy utilized in the PPL can effectively compute the encrypted data without decryption to assess the holomorphic operation equivalent to plaintext data without compromising the model and data privacy.

III. Proposed VSI-IoT Framework with

In this section, we discuss the structural overview of the VSI-IoT framework with privacy-preserving, threat model, learning using 2D-CNN, and edge computing to realize the efficiency of both training and classification in order to guarantee model privacy.

A. Preliminaries

The following provides explanatory descriptions of the VSI-IoT framework to fully understand the purpose of privacy preservation learning with lightweight model aggregation.

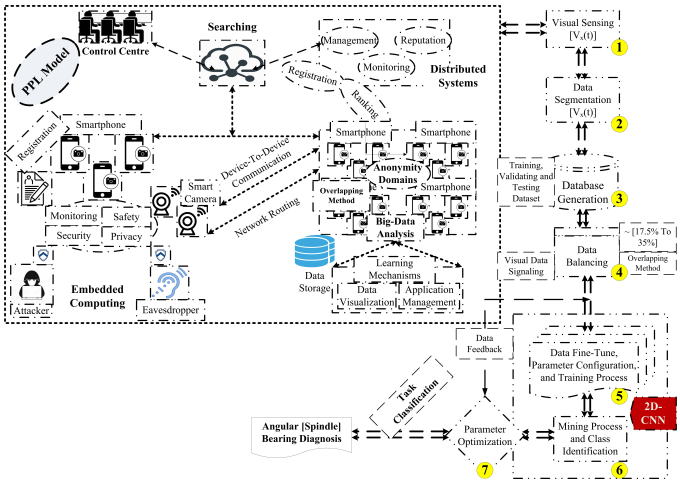


Fig. 2: Privacy Preserving Learning Model.

1) Privacy Preserving Learning Model

Fig. 2 shows the privacy peserving learning model. In this model, the end-users prefer to use a smart device with a safe ecosystem to examine privacy regulations. The proposed VSI-IoT has four key entities: a control center, distributed systems, embedded computing, and big-data analysis to provide end-to-end communication. The VSI-IoT uses D2D technology as a computing paradigm that utilizes a control center to realize the functional blocks, such as information retrieval, searching, and group administration. The system has dedicated Internet connectivity to examine the functionality of distributed systems, such as monitoring, management, and reputation. The monitoring function uses the registration process to authenticate the application services, whereas reputation holds the ranking function to manage the usage pattern. The significant roles of the key entities are as follows.

1) **Visual Sensing Devices:** A multi-range visual sensing device is considered to analyze short-range and long-range data transmission.
2) **Cloud Application Services and Visual Data Analytics:** Visual IoT considers an extensive operation to exchange industrial data via a dedicated communicated protocol, such as Bluetooth mesh, cellular, openthread, sigfox, and LoRaWAN.
3) **Middleware Layer:** The cloud-hosted middleware system aims to facilitate the communication scenario of machine-to-infrastructure that demands data pushing over communication protocols, such as HTTP, MQTT, and CoAP.
   - *Control Center* collects and transmits the visual data to the nearby computing server which could apply the data analytics to infer the data behaviors.
4) **Scenario of Large-scale Deployments:** Surveillance cameras are recommended to capture real-time data. Visual data is a prime element to meet the constraints of the application domains.
5) **Embedded Processing Units:** Initially, the traditional IoT devices are embedded with various capabilities, such as sensing, limited computation, communication, power, and storage. These scalar sensors connect to the sensors through the gateway to transmit the data.
   - *Embedded Computing* connects visual sensors in the IoT environment to process voluminous data on-board processing, ambient computing, and intelligence in order to create efficient data analytics in cloud environments.
6) **Network Connectivity in VSI-IoT:** With technological advancements, the IoT has transformed from the Internet of Things to the Internet of Everything. Thus, IoT networking requires more

embedding connectivity into physical things that can sense and transmit the required information from one node to another.

7) **Emerging Technologies of Cloud and Edge in VSI-IoT:** Of late, CC provides promising technological services to the IoT, including large-scale computing, analysis, and storage power.
8) **Distributed Systems** utilizes several autonomous units such as monitoring, management, and reputation across various computing devices to achieve a shared goal with an AI-supported pervasive environment.
9) **Big-data Analytics** often involves a complex process with a large amount of sensing data to uncover hidden patterns or correlations using strategic learning models.
10) **Anonymity Domain** uses third-party access to protect the information of the communication systems against privacy breaches in order to maintain ownership.
11) **Overlapping Method** applies a few adaptable models to interpret the key difference of piecewise function in an effort to safeguard model privacy.

2) Learning Model Using 2D-CNN

To manage the network workloads, the VSI-IoT framework uses system intelligence, driving computing services over edge networks. This idea reduces the complexity of a backbone network with possible storage/computation resources to circumvent the propagation delay [25]. Technical strategies, such as game theory and convex optimization, can also be employed using various test cases, including uncertain input, dynamic conditions, and temporal isolation to reveal the challenges of the wireless channel and security policies. The dynamic condition integrates the computing and communication systems to deal with the resources of the edge nodes. However, temporal isolation considers the Lyapunov optimization to achieve better resource optimization to meet the objective of edge computing systems. In consequence, the complexity of the networks is expected to grow by more than 2000 parameters [26]. Therefore, the proposed system uses AI techniques for better optimization, including physical, data-link, and traffic control. Reinforcement learning includes deep Q-Learning and deep reinforcement learning can also be applied to realize the difficulties of edge computing and caching.

The trained model develops effective resource management to gather the training data that may be either distributed or centralized through the knowledge of the learning agents (as shown in Fig.2). Fortunately, the deep learning frameworks verify the correctness of the results, i.e., target label via cloud server to minimize the target labels to protect their privacy with less computation and communication costs.

3) Threat Model

Let us assume that e_d is so credible, while the computing servers, including cloud and edge, are considered honest but curious (hBC). This assumption suggests that the computing device, i.e., the edge node, effectively wraps up its prediction process of deep learning. On the other hand, the modeling system may infer the private information of training data to collect the predictive outcomes of the data environment. Consequently, the proposed VSI-IoT generates samples similar to target labels to add bogus or erroneous information, which in turn applies a generator and discriminator to train the modeling parameters and to classify the models accurately drawn by the target labels.

To protect cloud servers against eavesdropping and collusion attacks, the proposed PPL adopts the concept of recoverable function with selective gradients. As a rule, each learning device initially shares its gradient vector G to associate data features with secret key s_k. Using a dedicated distributed system, the process involved in edge computing can group the privacy of end users' data via anonymity domain to examine the aggregated gradients at the cloud server. To probe privacy preservation among end users, the threat model additionally applies asynchronous encrypted selected gradient (AESG) with multi-key compliance supporting the boundless number of computing devices. The key assumptions are as follows.

1) We use the distributed parameters of the PPL to investigate the pattern of aggregated values and also utilize the appropriate machinery system via the cloud to preserve selected gradients by using lightweight homomorphic encryption; and
2) We allow adversary $\mathcal{A}_{\mathcal{D}}$ to guess or learn any sensitive information of a computing device to assess the privacy of the local and aggregated dataset in order to determine semantic security against chosen plaintext attack (CPA).

B. Learning the Workload Using Edge Computing

The VSI-IoT architecture integrates the learning strategies using edge computing to train data features locally, which optimizes the computing tasks of the collaborative models to update the shared data via lightweight local aggregation. Edge and cloud server accesses the uploaded parameters to converge the global model, which distributes the processed data using a flexible aggregation strategy to optimize the learning costs, including computation and communication. This learning model applies two-layer encryption-decryption to deal with the keys of the edge device and cloud server. In real-time, cloud server c_s uses an edge computing system including edge server e_s and edge device e_d to define the modeling parameters mp_i^l where $1 \leq i \leq N$.

$\{En_f, \beta\}$ includes a distributed dataset D_i^l under a computing server $l, w_i^l(t)$ to define the user parameters i.e., (i, U_R) These parameters represent layered output l_r and loss function represented by $F(w)$ to initialize the weights of the global parameters via a cloud server. This server uses a dedicated control center to broadcast the computing tasks participated by the local clients to aggregate the machinery information. The edge networks integrate with the cloud data centers to store and analyze a large volume of private data.

C. 2D-CNN - Training Design and Device Configuration

The proposed AI-supported model includes a network structure of 2D-CNN to examine device pattern recognition and fault detection systems. In this proposed model, a few computation layers were explored to minimize the execution time, which requires a single raw input, i.e., visual data, to test the characteristics of angular bearing failure. The major defects of the machinery process include F-IR, F-OR, and F-RE to analyze four levels of damage severity [7,14,21,28,40mm]. The machinery-bearing datasets such as IMS-Rexnord [27], CWRU [28], and Paderborn [29] are chosen to analyze the proposed model based on 2D-CNN. The chosen dataset works together with the convolution layer to offload the computing tasks, improving the communication efficiency of the system. The computing layers, including convolution and pooling, process the machinery data via a dedicated system to update the modeling parameters. This system selects the target labels randomly to calculate the encrypted data via different edge computing devices and minimize the computation costs of the data center. Algorithm 1 shows the descriptive flow of the convolution layer, which locally trains the data features to achieve privacy protection.

The applied algorithm uses a convolution layer with $m \times m \times D$, s, and p as the input matrix, stride, and padding values to generate the owners' encryption and decryption keys i.e., $\{En_{c,d}, 1 \leq d \leq D\}$ and $\{De_i, 1 \leq i \leq K\}$ where $En_{c,d}$ is a $m \times m$ random input matrix and De_i is a cross-matrix computation i.e., $(\frac{m-k+2p}{s})+1 \times \frac{m-k+2p}{s}+1$.

The simplified expression uses $Conv(En_{c,d}, i^{th})$ to represent the convolution operation i.e., for i^{th} kernel with $En_{c,d}$.

Algorithm 1 2D-CNN Preparation in Computing System

Require: Input size $m \times m \times D$m stride sm padding p, K kernels
Ensure: Encrypted keys $En_{c,d}$, $1 \le d \le D$, Decryption keys De_i, $1 \le i \le K$

▷ *For Edge System with Device Configuration.* ◁
Initialize the machinery with the target labels w_0
for $n \leftarrow 1$ to N **do**
  for each machinery $k \leftarrow 1$ to K in parallel. **do**
    Train the IMS-Rexnord Bearing Dataset Using 2D-CNN.
    Generate a random $m \times m$ matrices with $En_{c,d}$,$1 \le d \le D$;
    for $1 \le i \le K$ **do**
      for $1 \le d \le D$ **do**
        Obtain $En_{c,d}$ as input for applying the convolution on i^{th} kernel, thus the output can be written as $Conv(En_{c,d}, i^{th})$;
        d++;
      Set $De_i = \sum_{d=1}^{D} Conv(En_{c,d}, i^{th})$;
      i++;
    After all, CNN with p convolution layer and q fully connected layer, processed p as $\{En_{c,d}, De_i\}_{1 \le d \le D_p, 1 \le i \le K_p}$ and q as $\{En_f, \beta\}$ as a final set of owners' key $\{En_{key}, De_{key}\}$ and send the target labels to the computing system;

Privacy Preserving with Model Aggregation using fault Bearing Datasets: The maintenance systems consider a testing bank to examine the bearing faults (i.e., IMS-Rexnord, CWRU, and Paderborn) that occurred after the service lifetime $\approx 100 million RPM$. The testing bank consists of an AC motor, accelerometers, Rexnord bearing, acquisition card, and bearing condition to analyze the machinery features, namely rotation speed $\approx 2020 RPM$, loading factor $\approx 6050 lbs$, sampling frequency $\approx 20 kHz$, and fault diameter 0.177 to 0.355 mm. Data segmentation is preferred to augment the process of 2D-CNN input. To realize the data in a usable form, the rotation speed converts into revolutions per second (rps) using $RPM/60$. In addition, the segmentation length t_d considers the number of interested revolutions to estimate over the previous revolutions using $\#revol(s)/rps$. After the successful segmentation, the machinery process divides the dataset into seven features: moderate F-IR, severe F-IR, moderate F-OR, severe F-OR, moderate F-RE, and severe F-RE, with no failure, reported as 1, find the configurable parameters; 2, train the computation model; 3, discover an intelligent model to test/validate the visual data using a privacy-preserving model.

The modeling processes present an efficient privacy-preserving classification to train the visual sensing industrial network (VSIN), which uses an intermediate aggregator to minimize excessive communication (i.e., using convolution and fully connected layers). The basic objective is to apply a modeling strategy to define the local update τ_1 and to discover a global model using parameter aggregation τ_2. The optimized algorithm, i.e., Algorithm 2, iterates k entities to train the downloaded data using 2D-CNN. Subsequently, the modeling network generates data features to encrypt the training data to minimize the communication rounds via optimized deep learning. This local aggregation aggregates the features using the edge model $w^l(t)$ to modify the global model $w^g(t)$. In this modeling, the computing systems locally update the data via τ_1 and τ_2 to minimize the computation cost. Moreover, the system aggregates the modeling randomly, including local and global, to complete the parameter computation. Therefore, the local aggregation is defined as

$$w^l(t) = \frac{\sum_{i \in \theta} |D_i^l| w_i^l(t)}{|D_\theta^t|} \tag{1}$$

Upon successful aggregation, the learning-based machinery system uploads the local model to formulate a global aggregation. The aggregation function is as follows:

$$w^g(t) = \frac{\sum_{l=1}^{L} |D^l| w^l(t)}{|D|} \tag{2}$$

Algorithm 2 Privacy Preserving and Aggregation using 2D-CNN in Computing System.

Require: Given Input Data & Trained 2D-CNN
Ensure: Executed 2D-CNN Predicted Values

Set a Layered Input $\mathbf{M}$ = Data Input;
Set Computing Layer = Collected Values of Convolution and Fully-Connected Layers using 2D-CNN;
▷ *Performing Privacy Preserving during offloading the encrypted data to Edge System via IoT Device.* ◁
Assign Input = First Layer.
▷ *Aggregate Computing System with Local Modeling.* ◁
while Computing Layer is not null **do**
  if Layer = Convolution Layer **then**
    Execute Lightweight Encryption with $\mathbf{M}$ as Input.;
    Set $\mathbf{M}$ = Computed Output By Convolution Layer.;
  if Layer = Fully Connected Layer **then**
    Execute Lightweight Encryption with $\mathbf{M}$ as Input.;
    Set $\mathbf{M}$ = Computed Output By Fully Connected Layer.;
  Receiving Data r_d and Parameter Features p_f.;
  for each system $l \leftarrow 1$ to L in parallel. **do**;
    Train the IMS-Rexnord Bearing using 2D-CNN.;
    $w_i^l \longleftarrow w_i^l(t-1) - \gamma \Delta F_i(w_i^l(t-1))$.;
    if $t | \tau_1 = 0$ **then**
      $w^l(t) \longleftarrow$ Perform Local Aggregation
      if $t | \tau_1 \tau_2 \neq 0$ **then**
        for each entity $i \ln C^l$ in parallel **do**
          $w_i^l(t) \longleftarrow w^l(t)$
  ▷ *Aggregate Computing System with Global Modeling* ◁
  if $t | \tau_1 \tau_2 = 0$ **then**
    $w^g(t) \longleftarrow$ Global Aggregated Data
    for each entity $i \leftarrow 1$ to N in parallel **do**
      $w_i^l(t) \longleftarrow w^l(t)$

D. 2D-CNN Model with Lightweight Encryption

The proposed model deals with several normalization techniques, such as layer and batch normalization, to prompt the detection of visual sensing. The model has a topological mapping to process the visual sensing using CNN. The system modeling creates a route to process the machinery data, which may execute a command to collect the sensing data and estimate the defect rate using a CNN unit. The CNN unit applies two CNN-based models to train the machinery dataset [30]. The first model locates the signal length, filter size, and quantity to configure the machinery parameters, whereas the second model trains and validates the visual data to achieve a better performance ratio. The main contribution of the 2D-CNN is to find the faults at an early stage and yield higher detection accuracy over an efficient machinery process. The design architecture uses the original data by training the local attributes to provide privacy protection. Contrarily, the entities may train the data to act maliciously, resulting in data leakage [31].

As a result, protecting entities with data privacy demands a proper encryption mechanism, such as homomorphic encryption, to convert ciphertext into plaintext. In addition, this mechanism applies 2D-CNN to finish off various computations without compromising their security and accuracy. Traditional algorithms, such as ElGamal encryption, incur high computation efficiency and leak the private key with the communication entities [32]. Therefore, this paper utilizes Feldman's verifiable secret sharing (FVSS) to use a data-sharing technology and to generate the keys without any trusted third parties [33]. It is worth noting that the length of the keystream in the stream cipher is strictly limited due to collision in the internal state learning process to

produce the same key pair. As a result, the encryption scheme cannot be semantically secure because the keystream is distinguishable. Thus, in this paper, we use the FVSS mechanism to incorporate the features of verifiable double-key encryption to perform the following methods:

Method I: Generating the parameters - Consider three computing parameters p, q, and g where q represents a prime integer with a cyclic group G; p defines a large prime integer satisfying the property of $q|(p-1)$; and g is the key generator of G.

Method II: Generating the keys - Each entity e_i generates t^{th} round to define a random polynomial $f_i(x) = \sum_{j=0}^{t-1} a_{ij}.x^j$ which locally stores the private key using $s = z_i = q_{i0} = f_i(0)$. Considering $s_{ij} = f_i(0) \mod p$ as a shared one to compute and broadcast $\alpha_{ij} = g^{a_{ij}} \mod p$ where $i = 0,1,2,...,(t-1)$, P_j verifies the computation using $g^{s_{ij}} = \prod_{i=1}^{t} f_i(j) mod$ p where $j = 0,1,2,...,n$. If the verification is unsuccessful, then the shared data s_{ij} of the entities is invalid. If the shared data is assumed to be a valid one, then the collaborator finds a secret key s_k using an interpolation method so-called Lagrangian polynomial. The recoverable function is as follows:

$$s_k = \sum_{i=1}^{t} f_i(j) \prod_{i \le j \le t, j \ne i} \frac{i}{(i-j)} \tag{3}$$

The computing system broadcasts a key parameter α_{i0} to generate a global private key, which generates a global public key using

$$pk_i = \alpha_{i0} = g^{z_i} \mod p$$
$$pl = \prod_i pk_i = g^{\sum iz_i} = g^x \mod p \tag{4}$$

The learning-based machinery system chooses a random integer $r, r \in Z_p^*$ as a shared-key $s_k = r$ which uses a public key $pk = g^{s_k} \mod p$.

Encryption- considers the data features to encode the transmitted message t_m and compute its respective ciphertext(s) c_t1 and c_t2 using $g^{s_k} \mod p$ and $t_m.pk^{s_k} \mod p$.

Decryption- considers the data features to decode t_m using Eq.(5).

$$\frac{c_{t1}}{c_{t2}} = \frac{t_m.pk^{s_k}}{g^{s_k x}} = \frac{t_m.g^{x.s_k}}{g^{x.s_k}} \mod p \equiv t_m \tag{5}$$

In the decoding process, the machinery system aggregates data features to restore the transmission via a global private key x. The learning system transforms t_m into $m' = g^m \mod p$ to handle the encryption process. Using this transformation, homomorphic encryption can be successfully applied in deep learning as well. This function can be computed as follows:

$$E_{nc}(t_{m1}) * E_{nc}(t_{m2}) = g^{t_{m1}} key^{1} * .g^{t_{m2}} key^{2} \mod p$$
$$= g^{t_{m1}+t_{m2}} key^{1+2} \mod p \tag{6}$$

Using Eq.(6), the actual data feature can be recovered using Pollard's rho algorithm [34]. It is also worth noting that the entities use the targeted labels to carry out supervised learning and do not expect any privacy-labeled data. As a result, this paper assures that the training samples do not consider any privacy labeling to classify the features of this machinery dataset.

2D-CNN Classifier Model: To specify the effectiveness of classification models and evaluate the versatility of 2D-CNN in identifying bearing failures, the machinery dataset so-called IMS-Rexnord contains three different classes (i.e., F-IR, F-OR, and F-RE) and 534 visual images, whereas the CWRU considers three distinct classes with 12000 spectrograms and the Paderborn also has three faulty bearing conditions like CWRU and IMS with 2560 samples. These models have an over-fitting issue for data training over unseen data inadequately. Thus, to prevent this issue on a small dataset, data augmentation is preferred. The system may apply a differential

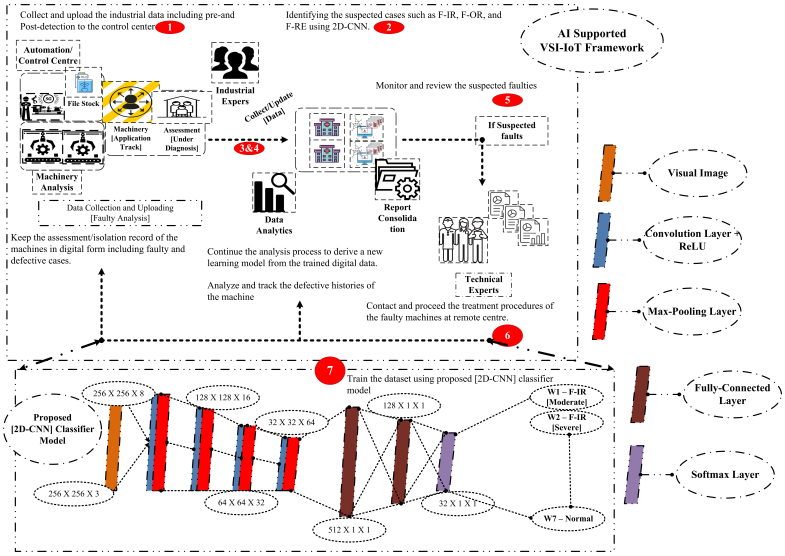


Fig. 3: AI-Supported Framework Using 2D-CNN.

transformation to enhance the sensing data and examine the defect rate, such as F-IR, F-OR, and F-RE. The proposed classifier simplifies the CNN-based modeling to prepare the dataset into 32 categories. Each model trains the prepared dataset to classify the visual images using a systematic architecture. The proposed classifier model considers a visual image, i.e., 256×256 with four convolutions, four max-pooling, and two fully-connected layers. The convolution layer utilizes the ReLU activation function that explores four output filters: 8,16,32,, and 64. The convolution layers include 5×5 strides, 3×3 strides, and 2×2 strides with a pool size of the max-pooling layer. In addition, it has two fully-connected layers, such as 512 channels and 128 channels, to analyze the fault rate using the ReLU activation function. To learn the loading factors of each class provided in IMS-Rexnord, CWRU, and Paderborn, we use 70% and 30% of images for testing and training, respectively to find the stages of the bearing defects before wearing into failure. The convolution network processes the softmax layer to complete the process of visual sensing, as outlined in Table II.

Machinery Datasets: The proposed model considers a CNN-based architecture to detect the status of the fault rate (i.e., of IMS-Rexnord [27], CWRU [28], and Paderborn [29]) which analyzes the visual images collected from the sensing unit. The potential challenges include semi- and fully-automatic machinery processes to prepare or collect a reliable dataset. The machinery process includes three faulty rates, such as F-IR, F-OR, and F-RE, to generate a dataset with 534 images, 12000 spectrograms, and 2560 samples to classify its internal defect. The proposed CNN-based network acts as a controller model to classify the preferred dataset into two categories, batch, and layer normalization.

The proposed privacy-preserving categorization reveals the AI-Supported VSI-IoT Framework that monitors, identifies, and analyses the potential threats in the IMS-Rexnord [27], CWRU [28], and Paderborn [29] Bearing dataset. Importantly, this prediction model observes the physical and social vulnerability like experimental measures to complete the aggregation model [35]. Fig.3 shows privacy-preserving Learning (PPL) using the 2D-CNN model that interconnects with the cloud network to provide functional interfaces, such as data collection, uploading, assessment, observation, and experts' decisions.

Isolation/Assessment Center – This center records the details of the isolated or assessment machines via the control center.

Data Observation – The observation center includes proper data

TABLE II: Configuration of the Proposed 2D-CNN Modeling

Layer Classifier	Output Shape	Kernel [Stride] Size
Input	256 × 256 × 3	-
Conv-2D	256 × 256 × 8	5 × 5-3 × 3
MaxPooling-2D	128 × 128 × 8	2 × 2-2
Conv-2D	128 × 128 × 16	5 × 5-5 × 5
MaxPooling-2D	64 × 64 × 16	2 × 2-2
Conv-2D	64 × 64 × 32	3 × 3-3 × 3
MaxPooling-2D	32 × 32 × 32	2 × 2-2
Conv-2D	32 × 32 × 64	3 × 3-3 × 3
Fully Connect-1	512	-
Fully Connect-2	128	-
Softmax	32	-

Hyper-Parameters	Values
Optimizer	SGD
Learning Rate	0.001
Batch-Size	32
Training Rounds	30
# Number of Entities	10
Aggregation Algorithm	2D-CNN

analysis and learning to build a suitable CNN model that provides a better real-time dashboard for industrial goods.

Industrial Experts – The experts monitor the activities of the suspected cases to infer the indicated failures.

Systematic Model– This networking infrastructure interconnects real-time entities over the Internet. It allows the VSI-IoT to:

1) Observe and upload the industrial data to investigate the suspected cases;
2) Maintain the industrial data over time to derive a specific analysis in order to record the possible vulnerabilities of the machines;
3) Communicate the predicted results to the industrial experts to inspect the vulnerabilities; and
4) Store the analyzed information to troubleshoot the machinery failures.

The intelligent privacy-preserving categorization framework includes three classical observations, such as F-IR, F-OR, and F-RE [36] to collect and upload the industrial data to the control center, including pre- and post-detection, as shown in Fig.3. Periodically, the user submits faulty information through the smart application, applying lightweight encryption to store the contact information of the infected machines in the controller database. Later, we utilize data analytics to consolidate the sensed data, whereby the storage of the digital records is regularized to infer the intrinsic features of the data using 2D-CNN models. Using the control center, the analytical information is presented on the real-time dashboard to construe the diagnostic procedure for the suspected machines. In this paper, we operate an effective technique, called normalization, to analyze the quality of the proposed AI model. This technique changes the behavior of layer distribution, which eventually adjusts the next layer to obtain a new distribution. As a result, the model can enhance the processing speed of the training set to achieve the global minima. It is worth noting that the technique can even perform a pre-processing step before executing the training process to improve the process of deep neural networks.

Since the analytical systems consider the feature, finding the values ranging from maximum to minimum to achieve a scope of the engineering process, the proposed 2D-CNN applies a logarithm scaling method to generate the feature values. To analyze the effectiveness of the proposed model, the techniques, such as BN and LN, were modified and trained after each of its convolutions. From Fig.4, batch normalization could not improve the performance ratio of the proposed 2D-CNN and controller model due to the size of the given bearing dataset, i.e., IMS-Rexnord [27], CWRU [28], and Paderborn [29]. Fortunately, layer

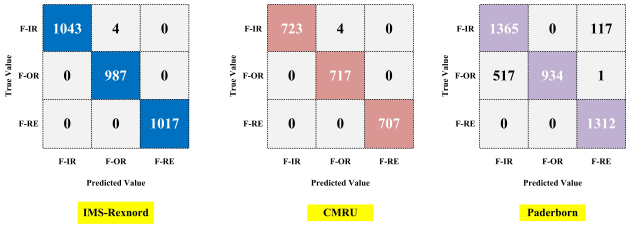


Fig. 4: Confusion Matrix of Bearing Dataset [27]–[29].

normalization with the proposed 2D-CNN effectively increases the accuracy of the trained and validated dataset. In addition, this normalization can even minimize the noisy level of the dataset, including training and validation, to make the model more reliable in comparison with other models, such as Proposed 2D-CNN and Proposed 2D-CNN+BN.

IV. Security Analysis

In this section, we consider two prominent factors protecting privacy and encrypting the selected gradients to secure the computing device or the participants against the cloud server or edge under the hBC. To guarantee the privacy of no leakage of secret information, the proposed PPL considers ciphertext distinguishability against chosen-ciphertext attack (CPA) as stated in [37], providing semantic security for selected gradients using lightweight encryption.

A. Privacy Analysis

Suppose the learning-based machinery system is honest but curious about its data features. This assumption demands a proper security requirement using the proposed PPC to achieve better verifiability with privacy protection. Most computing systems enforce the program to restrict the server access without additional verifiability. As a result, formal security proof is not necessary to protect the utilized data without compromising its privacy. The proposed PPC does not allow any entity to upload or share the local parameters with others to protect the privacy features. The trained model considers the learning-based machinery data locally to perform supervised learning, which associates with honest and semi-honest entities to operate the ciphertext directly. The proofs shown in Theorem 1 & 2 secure the privacy of the features.

Theorem 1: *The proposed PPC meets the privacy requirements of the featured parameters.*

Proof- In the course of training the privacy features, the computing systems use a specific layer to reveal additional information. As a result, the server cannot expose featured data because it is highly reliant on g^{t_m} to protect the privacy of the entities. The featured parameters utilize two possible ways to perform a decryption process, including additive and multiplicative operations. The former uses s_k to obtain entity information via g^{s_k} because it is preserved by a discrete logarithm problem, whereas the latter uses g^{s_k} and g^{μ} to evaluate the hardness of the computational Diffie-Hellman algorithm obtained by $g^{\mu.s_k}$. Moreover, the entity cannot receive any featured information before applying the encryption process. Thus, the proposed lightweight encryption can meet the privacy requirements of the featured parameters to hold Theorem 1.

Theorem 2: *The assumption makes clear that the fully homomorphic encryption can secure the minority features under the semi-honest model.*

Proof- Suppose a situation with a compromised or corrupted entity, i.e., Charlie. In the proposed PPC, Charlie obtains the encrypted data $E(G^{'})$ from Eve. Accordingly, Charlie applies encryption to recover the information or plaintext using the compromised private key. Although Eve uses a confusion operation on the row and column

of $E(G^{'})$, Charlies cannot infer the true ordering of $E(G)$. $E(G^{'})$ can easily confuse the encryption process using $E(\sigma_1)$,$E(\sigma_2)$,...,$E(\sigma_{t_m})$ and cannot infer any private information of the dataset owned by Eve. Further, while Charlie obtains the sensitive data I from Eve, the corresponding plaintext can be recovered by

$$\begin{aligned} SD(I) &= (\gamma^{(1)},\gamma^{(2)}\gamma^{(3)},...,\gamma^{(n)}) \\ &= [gap \times (q_{b_i}.x) - p_i) + p_i] \times Noise \end{aligned} \tag{7}$$

i,x are a few random numbers that cannot be deduced by Charlie to validate the data features, i.e., $SD(I)$. Moreover, $SD(I)$ can use random noise whereby Charlie cannot infer any additional information from Eve newly generated as p_{new}. Thus, even if Charlie is a dishonest or compromised entity, Eve's feature is preserved to secure the sampled data under the semi-honest model, as stated in Theorem 2.

B. Analysis Using CPA

In this analysis, we set up a few parameters $\{s_k,pk\}$ as a key pair of the learning devices or participants using lightweight homomorphic encryption to train the network over the verifiable dataset. By doing this, the verifiable learning model updates its encrypted weighted parameter within the computing space, handled by the edge or cloud server. After obtaining the encrypted model i.e., $E_M(-\alpha.G_{local}^{selective(i)})$ from the computing device or a participant, the associated cloud server verifies the integrity of the ciphertext operation and accordingly, finds its associative elements: $E_M(Wt_{global} + E_M(-\alpha.G_{local}^{selective(i)})$ in order to signify its equivalency $= E_M(Wt_{global} - \alpha.G_{local}^{selective(i)})$ signifying the randomly selected gradient uploaded to the cloud server. As a result of this, E_M ensures its equality with the property of homomorphic lightweight encryption to represent its dimension into $E_M(Wt_{global})$ and $E_M(-\alpha.G_{local}^{selective(i)})$ to keep updating the weights within the encrypted space $E_M(Wt_{global})$.

Theorem 3: *Achieving security against the cloud server - In this case, while the homomorphic lightweight encryption applied in the basic assumption is CPA-Secure, then the proposed PPL does not disclose any sensitive information about the aggregated dataset to the hBC cloud server.*

Proof- The local computing devices or participants associated with the PPL model can upload the encrypted gradient to the hBC cloud server. Given this, we claim that while utilizing homomorphic lightweight encryption in the proposed PPL is CPA-secure over the selected gradient, the sensitive information obtainable on the participant's data cannot be leaked to the hBc cloud server.

To prove the security for selected encrypted gradients as a resilient against the hBc cloud server, we consider a gaming event among the adversary A_D and a challenger C_H as specified in **Definition 1**. It is as follows.

Definition 1: CPA-Secure

Parameter Setup: C_H chooses a public attribute p_a and a pairing key of the computing device or participant $\{s_k,pk\}$ to compute the vectors over Z_p^*. Also, to play the game fairly, we have granted the defined parameters $\{pk,s_k\}$ access to A_D.

Task In-Challenge: At this stage, two plaintext messages m_1 and m_2 are chosen with the same length by A_D, and accordingly, submitted to C_H. To find $CT^{'} = E_k(pk,m_b)$, C_H randomly selects a guessing bit $g_b \in \{0,1\}$. On obtaining $CT^{'}$, C_H sends the key value to A_D to find the guessing bit $g_b^{'}$. Hence, lightweight homomorphic encryption claims that it can be secure against chosen ciphertext attacks, if the predictive outcome $PO_{A_D{}^{CPA}}(\lambda) = |P_r[g_b^{'} = g_b] - \frac{1}{2}$ is negligible in the given derivative λ.

Further, in the given assumption, the decisional-based verifiable secret sharing (D-VSS) considers $\{pl,pk,s\}$ to define a feature key without the knowledge of trusted third parties. As a result of this, we obtain $S \leftarrow Z_p^{(m,pl)}$ to compute three vectors $s \leftarrow Z_p^{m\times 1}$, $z \leftarrow Z_{0,g}^{pl\times 1}$, and $r \leftarrow Z_{(0,g)^P m\times 1}$ over the set of integers $\langle Sx + r\rangle$ to define the polynomial probabilistic time (PPT) algorithm P_{PT}:

$$PO_{A_D}^{CPA(pl,pk,s)}(\lambda) = |P_r[P_{PT}(S,r) \rightarrow 1]|. \tag{8}$$

The D-VSS assumption neglects $PO_{A_D{}^{CPA(pl,pk,s)}}(\lambda)$ under λ function when the aggregated dataset does not perform any encryption. Also, it is worth noting that the computation results of encryption are uniformly randomized and cannot be distinguished by using random integer values while the selected gradients are already applied using D-VSS encryption.

Claim - ***Under the D-VSS assumption, the DVSS-based lightweight encryption is CPA-secure. Particularly, for any*** A_D***, there is an algorithm*** P_{PT} ***with the same execution time such that*** $PO_{A_D}^{CPA}(\lambda) \leq (l+1).PO_{A_D}^{CPA(pl,pk,s)}(\lambda)$.

We also apply the same formal security analysis as stated in [38] to prove that the sharing vectors within S are fully secure under the randomization of matrix P to achieve privacy protection on the learning model. At first, we initialize the mapping process to start the original game addressed in Section III.D in order to assign $P \leftarrow Z_p^{(pl\times l)}$ matrix with A_D randomly. In particular, we find the defect rate using $P = (pl.R - S.A) \in Z_p^{(pl\times l)}$, where R and S are the generated matrices to incorporate the feature of verifiable double-key encryption on the given dataset model. This randomized key encryption process makes P indistinguishable from A_D in achieving data protection. As a result of this, the secret encrypted vectors under this assumption hold better data and model privacy within the column matrix S.

However, it is still challenging to ensure that the conditional probability with $gcd(p,q) = 1$ to find the event occurrence in the given matrix P to the desired D-VSS form. By estimating the value $pl^{-1}.P = R + (-pl^{-1}.A).S \in Z_p^{(pl\times l)}$, we can determine the randomized matrix $A^{'} = -P^{-1}.A \in Z_p^{(pl\times pl)}$ to contribute the overall feature vectors over ciphertext $\mathbf{c}^*$. In addition, we exhibit $P^{'} = pl^{-1}.P \in Z_p^{(pl\times pl)}$ to ensure randomness under the D-VSS assumption, thereby we prove the randomization matrix P to compute the vectors over Z_p^* as stated in **Definition 1**. ■

The strategy implicated in randomized matrix P enhances the security of the lightweight encryption to mitigate the risk of potential threats from A_D. This technique employs a learning transformation strategy over double-key encryption to exhibit the characteristic of the given matrices R and S whereby the randomized matrix P appears to be indistinguishable from any A_D to infer the random noises. As a result, we claim that the proposed PPL with lightweight encryption can protect the integrity of the data vector to improve its resiliency against the attempts of A_D.

TABLE III: Comparison of Performance and Computation Overheads on the IMS-Rexnord Bearing Dataset[27]

	IMS-Rexnord Bearing Dataset [27]								
Models	Categories	Training Dataset	Testing Dataset	Training Time (sec)	Testing Time (sec)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed 2D-CNN	5	1700	390	149.96	0.3763	91.9	89.31	91.67	90.31
Proposed 2D-CNN+BN	5	1700	390	148.69	0.3499	94.9	92.33	94.71	91.67
Proposed 2D-CNN+LN	5	1700	390	119.84	0.2730	97.67	96.87	98.11	94.61
FLS [16]	5	1700	390	191.14	0.4421	86.9	86.17	88.81	84.91
EFL-BN [17]	5	1700	390	197.27	0.4488	88.9	88.66	90.43	84.69
EC-FLM [18]	5	1700	390	201.12	0.4721	84.9	83.09	86.22	81.36
UDLM [19]	5	1700	390	203.67	0.4937	82.35	80.19	83.47	78.77

V. Experimental Results

This section applies the proposed 2D-CNN, 2D-CNN with BN, LN, and other existing models [16]–[19] to evaluate the standard metrics,

TABLE IV: Comparison of Performance and Computation Overheads on the CWRU Bearing Dataset[28]

	CWRU Bearing Dataset[28]								
Models	Categories	Training Dataset	Testing Dataset	Training Time (sec)	Testing Time (sec)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed 2D-CNN	5	1398	574	151.21	0.3467	94.29	93.18	94.94	95.87
Proposed 2D-CNN+BN	5	1398	574	147.33	0.3131	97.7	96.91	97.31	93.74
Proposed 2D-CNN+LN	5	1398	574	113.44	0.2170	99.98	98.12	98.51	96.91
FLS [16]	5	1398	574	183.77	0.4371	91.4	91.78	93.93	89.31
EFL-BN [17]	5	1398	574	187.85	0.4288	92.65	93.19	95.62	88.45
EC-FLM [18]	5	1398	574	191.51	0.4643	91.67	88.7	93.57	87.9
UDLM [19]	5	1398	574	195.94	0.4798	87.61	85.9	90.79	83.93

TABLE V: Comparison of Performance and Computation Overheads on the Paderborn Bearing Dataset[29]

	Paderborn Bearing Dataset[29]								
Models	Categories	Training Dataset	Testing Dataset	Training Time (sec)	Testing Time (sec)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed 2D-CNN	5	2560	953	167.19	0.6143	87.91	85.61	88.14	86.91
Proposed 2D-CNN+BN	5	2560	953	171.47	0.6891	92.8	91.1	91.06	89.9
Proposed 2D-CNN+LN	5	2560	953	137.61	0.3965	96.8	95.37	97.91	95.59
FLS [16]	5	2560	953	213.8	0.8453	82.57	83.19	86.3	80.47
EFL-BN [17]	5	2560	953	217.96	0.8971	81.9	83.1	84.27	86.87
EC-FLM [18]	5	2560	953	247.67	0.9568	77.9	76.5	79.15	75.34
UDLM [19]	5	2560	953	267.89	0.9974	71.8	69.81	73.34	68.5

such as Accuracy, F1-Score, precision, recall, and Time per Epoch. A proper investigation was performed with a few dedicated software tools namely Tensorflow, Keras, and Torch to examine the classifier models over the given datasets i.e. IMS-Rexnord Bearing [27], CWRU [28], and Paderborn [29]. The tools were operated by Windows 11 pro-64-bit equipped with Core i9 and 32GB RAM, which can substantially scale the training system to handle a large number of computing nodes and perform the local training effectively. While training the data, a few dedicated virtual clients were connected over a local network to predict or identify the faulty diagnostic rate of the learning-based machinery system, including F-IR, F-OR, and F-RE.

Standard Metrics – The evaluation metrics, including accuracy and F1-Score, were chosen to measure the performance efficiencies of the proposed 2D-CNN, such as batch and layer normalization and other existing models [16]–[19]. This metric can mathematically be expressed as follows:

Accuracy considers several classifiers to compute the number of instances classified correctly to the available number of instances.

$$Accuracy = \frac{(T_P + T_N)}{(T_P + T_N + F_P + F_N)} * 100 \tag{9}$$

The F1-Score can be calculated using the measurements of precision and recall to determine the accuracy of the models.

$$Fl\text{–}Score = \frac{2 \times (Precision * Recall)}{(Precision + Recall)} \tag{10}$$

The Precision determines the percentage of machinery defects, i.e., F-IR, F-OR, and F-RE, that correctly classify the faulty one among the retrieved instances returned by relevant classes.

$$Precision = \frac{T_P}{T_P + F_P} * 100 \tag{11}$$

Recall determines the percentage of actual machinery defects to classify the faulty state among the retrieved instances processed by the machinery system.

$$Recall = \frac{T_P}{T_P + F_N} * 100 \tag{12}$$

The testing rig selects its healthy state ≈ 121155 data points. The proposed 2D-CNN and other existing models consider equal dimensions for the sequence of tensors. Assuming that a $12KHz$ sampling rate with rotation speed $\approx 1750rpm$ needs ≈ 411 points per revolution, thus this paper chooses a half revolution ≈ 205 points to obtain better training and testing accuracy. Furthermore, the data points of the healthy state can also be considered to branch off the sampling data and observe the vibration behavior of the machinery system in terms of F-IR, F-OR, and F-RE. The testing and training accuracy for ≈ 50 epochs process forward and backward networks to learn model 1.0000 and 0.9973 respectively. As a result, the computing time of ≈ 50 epochs was $57sec$.

Initially, the influential ratio, i.e., a local update, was considered to examine the aggregation factors, including local and global models. Interestingly, to optimize the learning mechanisms, τ_1 and τ_2 are determined to be more consistent in aggregating the local models. The learning result shows that while increasing the local gradient, the performance models can handle their derivative functions cautiously to minimize their communication cycles with a remote server. The proposed 2D-CNN, including BN and LN, can achieve better accuracy within a short period than other existing models [16]–[19] to lower their communication costs (as shown in Fig.5). Furthermore, the experimental results showed that the proposed 2D-CNN could even update its local gradient value further without affecting the model performance to maintain the cost efficiency of the server.

Initially, the bias value was set to 0 randomizing the sample values from the normalized distribution (0,0.022). To assess the performance of the proposed 2D-CNN, 2D-CNN with BN, 2D-CNN with LN, and other existing models [16]–[19], the simulation environment was set with 10 edge devices, 1 edge servers, and a centralized server. This environment considered the dataset, i.e., IMS–Rexnord Bearing, to analyze the performance of the classifications along with computation overhead. Table III lists the performance and computation overheads of the proposed 2D-CNN, including BN and LN and other privacy-preserving approaches on the IMS-Rexnord Bearing Dataset [27]. The proposed 2D-CNN with BN and LN cautiously handles the inherent features of the dataset to infer the behaviors of F-IR, F-OR, and F-RE, solving the issues of data heterogeneity in terms of F-IR, F-OR, and F-RE, which examine three different classes to optimize the feature categories. Importantly, the examination result shows that the proposed 2D-CNN, including BN and LN, achieves better classification performance (i.e., 91.9%, 94.9%, and 97.67%) than the other existing privacy-preserving approaches [16]–[19].

Similarly, Table IV shows the performance and computation overheads of the proposed 2D-CNN, including BN and LN and other privacy-preserving approaches on the CWRU bearing dataset [28]. In order to compare the performance and computation overhead, the proposed 2D-CNN with BN and LN and other existing privacy-preserving approaches [16]–[19] utilized the sampled spectrogram images as an input to train them in the same 2D-CNN classifier model. The examination results reveal that the proposed 2D-CNN with BN and LN achieves a better trade-off between classifier hit rate and false-alarm rate to distinguish the degree of separability among the classes (i.e., F-IR, F-OR, and F-RE) to gain superior classification performance (i.e., 94.29%, 97.7%, and 99.98%) than the other approaches [16]–[19]. Lastly, Table V demonstrates the proposed 2D-CNN, including BN and LN and other privacy-preserving approaches on the performance and computation overhead over Paderborn [29] to learn the optimized cell structure which uses the 2D-CNN classifier model like IMS-Rexnord Bearing [27] and CWRU [28] to share the normalized structure with reduction cell while existing with the max-pooling operation. The reduction cell structure utilized a few selective parameters like rotation speed $1800RPM$, torque $0.75Nm$, and axial force $1000N$ over Paderborn [29] to optimize the appropriate samples whereby noise disturbance in the given sampling data was eliminated. The observational results show that the proposed 2D-CNN with BN and LN acquires a more suitable performance ratio (i.e., 87.91%, 92.8%, and 96.8%) than the other existing approaches [16]–[19] to

sort out the performance issue sourced from its structural restriction.

TABLE VI: Estimated p-Values using Paired T-Test

	IMS-Rexnord Bearing Dataset [27]		CWRU Bearing Dataset [28]		Paderborn Bearing Dataset [29]	
Models	Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)
Proposed 2D-CNN vs FLS [16]	0.014912	0.025679	0.013912	0.024323	0.017317	0.027654
Proposed 2D-CNN vs EFL-BN [17]	0.014512	0.025314	0.013691	0.024172	0.017476	0.027773
Proposed 2D-CNN vs EC-FLM [18]	0.014269	0.025198	0.013742	0.024296	0.017532	0.027831
Proposed 2D-CNN vs UDLM [19]	0.014112	0.025037	0.013137	0.024171	0.017237	0.027762
Proposed 2D-CNN+BN vs FLS [16]	0.014633	0.025439	0.013091	0.023125	0.017657	0.027871
Proposed 2D-CNN+BN vs EFL-BN [17]	0.014467	0.025287	0.013097	0.023133	0.017677	0.027887
Proposed 2D-CNN+BN vs EC-FLM [18]	0.014197	0.025107	0.013031	0.023191	0.017616	0.027817
Proposed 2D-CNN+BN vs UDLM [19]	0.014021	0.024971	0.013091	0.023113	0.017621	0.027843
Proposed 2D-CNN+LN vs FLS [16]	0.014291	0.025317	0.013247	0.023159	0.017691	0.027891
Proposed 2D-CNN+LN vs EFL-BN [17]	0.014247	0.025168	0.013017	0.023015	0.017591	0.027797
Proposed 2D-CNN+LN vs EC-FLM [18]	0.014043	0.025003	0.013041	0.023107	0.017619	0.027823
Proposed 2D-CNN+LN vs UDLM [19]	0.013981	0.024877	0.012891	0.022987	0.016983	0.027794

A. Statistical Analysis with T-Test

Table. VI presents a few missing predictions of the learning-based machinery system, such as F-IR, F-OR, and F-RE. The computational values of the matrix relate the metric space of two distinctive classes to draw out the testing model using 2D-CNN. In addition, in the testing scenario, data pre-processing, selection, and manipulation were not considered, but the modeling features determined the strength of the machinery system, including a healthy and faulty state. As a result, the proposed 2D-CNN, including BN and LN, achieves better classification performance than other existing privacy-preserving approaches [16]–[19] consider a paired sample $T-Test$ to provide validation statistically, which observes the measuring designs of the machinery system. The parametric procedure considers a significant level of hypothesis <0.05 on the given dataset [27] to eliminate the variation among the sampling defects, such as F-IR, F-OR, and F-RE. Moreover, the observation considered two sampling proportions that classify the testing hypotheses, accuracy and F1-Score, to show the competitive features of the proposed 2D-CNN with BN and NC over other existing approaches [16]–[19].

Assume $\hat{p_1}$ and $\hat{p_2}$ be the accuracy and F-Scores from the machinery classifier 1 and classifier 2, and n is the number of bearing samples with healthy and faulty states. The bearing samples classify the parameters s_1 and s_2 to obtain suitable findings less than <0.05. This can be functionally derived as: $\hat{p_1}=\frac{s_1}{n}$ and $\hat{p_2}=\frac{s_2}{n}$. Accordingly, the testing statistic is obtained that is as follows:

$$T_S=\frac{(\hat{p_1}-\hat{p_2})}{\sqrt{\frac{2\hat{p}(1-\hat{p})}{n}}} \tag{13}$$

where (13) $\hat{p}=\frac{(s_1+s_2)}{2n}$ This analysis considered accuracy and F1-Scores of the proposed 2D-CNN with BN and NC with other existing approaches to verify whether the classifier $\hat{P_2}$ is better than the classifier $\hat{p_1}$ to signify their competitive solutions over the sampling features. The typical hypothesis includes:

- $H_0: p_1 == p_2$ [Hypothesis is NULL]
- $H_a: p_1 < p_2$ [Alternate hypothesis claims that the newer one is better than the existing one]

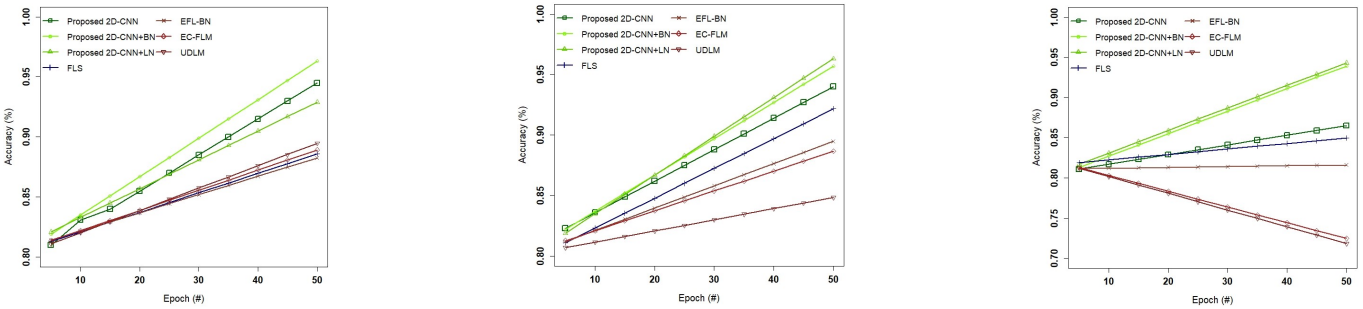
B. Analyzing Communication Cost Efficiency

In this analysis, the execution time adopts the concept of segmentation learning, which builds a suitable framework on the given dataset [27] to examine the communication efficiencies of the proposed 2D-CNN, including BN and LN, over other existing privacy-preserving approaches [16]–[19]. The segmentation learning utilizes a strategy of a mini-batch stochastic gradient to employ a batch size of 10 over the communication clients, i.e., edge devices, to update the modeling accuracy locally. According to [17], $\approx 100ms$ is set to be a communication latency between the edge and centralized server, whereas $\approx 10ms$ is set to be a transmission latency between the edge device and edge server. On the other hand, the edge device needs $\approx 1ms$ to complete its local update. Similarly, for each training round, the communication costs were calculated to compare the execution time of the modeling accuracy. The proposed 2D-CNN, including BN and LN and other existing models, uses identifier dimension 64 to find the communication costs of the edge devices. While this study examined the cost efficiencies of the models, the proposed 2D-CNN with BN and LN minimized the communication latency of the edge devices to gain better performance efficiency than other existing models [16]–[19].

The proposed 2D-CNN with BN, LN, and other existing models analyzed the training strategies to investigate the augmentation parameters, such as visual image, split ratio including testing and training, and elemental loss. While conducting experimental analysis, the key parameters were examined over dynamic/static configuration on the dataset (i.e., IMS-Rexnord Bearing [27], CWRU [28], and Paderborn [29]) to probe its transmission latency. The programming tool, Python-V3.5, and Keras frameworks, such as Tensorflow-2.1, were used to analyze the modeling accuracy and computing time. This strategical approach reveals that the proposed PPC could combine three models effectively to aggregate the learned features constructed by normalized Gaussian distributed data. The loss function was optimized by applying the stochastic gradient descent (SGD) with a ~ 0.9 momentum. A learning rate of [0.002] was allowed to achieve higher performance over several classes (i.e., F-IR, F-OR, and F-RE). The proposed 2D-CNN with BN and LN shows less communication latency than other existing models to optimize the feature categories and preserve the prediction ratio in identifying the system's defects at an early stage (as shown in Fig.6).

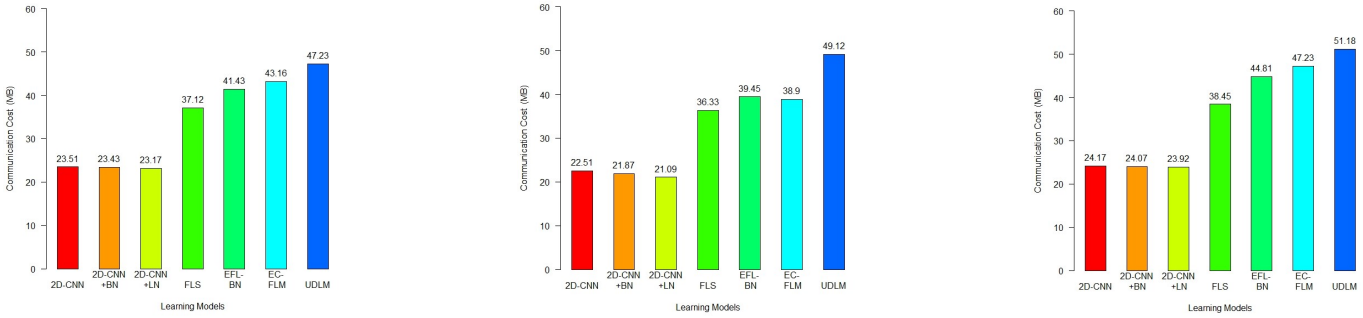
C. Performance Testing

To evaluate the performance of the 2D-CNN classifier model, two additional metrics namely scalability and verification overhead were chosen. A classical neural network so-called multi-layer perceptron (MLP) was considered to provide proof of correctness over the aggregated results. The experimental testbed utilized a few dedicated libraries including TensorFlow and Keras to adopt the functional features of CNN. Additionally, the battery-operated IoT and edge devices were kept reserved using Raspberry Pi 3 [Model B] with 1.2 GHz and Macbook Pro with 3.1 GHz which were later connected over a dedicated WiFi connection to examine the local aggregation model obtained through a group of users. Moreover, the bearing datasets including IMS-Rexnord Bearing [27], CWRU [28], and Paderborn [29] utilized an architecture of AlexNet to analyze its security parameter $\lambda \approx 160$ associated with input data of the convolutional and fully- connected layers. Lastly, the privacy-preserving approaches [16]–[19] including the proposed 2D-CNN with lightweight encryption employed the prediction models such as CryptoNets and LoLa to examine the prediction accuracy of the bearing datasets (i.e., IMS-Rexnord, CWRU, and Paderborn). The prediction models applied a 4096 prediction request to operate the communication via dedicated IoT and edge devices which parallelize the operation to examine the throughput ratio of the proposed 2D-CNN with lightweight encryption and other relevant approaches [16]–[19] over an hour.



(a) IMS-Rexnord Bearing (b) CWRU Bearing (c) Paderborn Bearing

Fig. 5: Accuracy (%) of the proposed 2D-CNN, including BN and LN and other privacy-preserving approaches in diagnosing the bearing failures over dataset [27]–[29].



(a) IMS-Rexnord Bearing [27] (b) CWRU Bearing [28] (c) Paderborn Bearing [29]

Fig. 6: Communication Cost (MB) of the proposed 2D-CNN, including BN and LN and other privacy-preserving approaches in optimizing the features over the prediction of system defects.

Protection Efficiency and Scalability The numerical analysis of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches applies a floating point operation **FLOP** to represent the functions such as addition and multiplication. The general convolution layer uses input size $n \times n \times D$, stride values s, padding size p, and kernel matrix $k \times k$ to construct a fully-connected layer where m is the dimension of the input matrix and T is the number of neurons. In order to build the blocks using CNN, the pooling region sets $q \times q$ size, consolidating the features of the bearing datasets (i.e., IMS-Rexnord, CWRU, and Paderborn) through the the2D-CNN with lightweight encryption model. The study analysis employed an uncompressed AlexNet to process the inference request of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches [16]–[19]. Each request requires 2.27 billion **FLOPs** to generate the encryption and decryption key which is proportionate to FaceNet (1.6 billion **FLOPs**) and ResNet (3.6 billion **FLOPs**).

While the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches utilized the AlexNet to establish the connection between IoT and edge devices, the proposed 2D-CNN with lightweight encryption consumed 112*ms* to process the request of the IoT devices whereas the other privacy preserving approaches [16]–[19] utilized 121*ms* 129*ms*, 127*ms*, and 136*ms*, respectively. It is worth noting that a set of computed keys (i.e., encryption and decryption) cannot be used more than once, however, the owner of the IoT device can determine more than 2096 keys in 5 minutes for the complex architecture of AlexNet. As a result, the IoT devices associated with 2D-CNN with lightweight encryption can efficiently handle the CNN requests generated by AlexNet to preserve the offloading data to the edge devices. Table VII, VIII, and IX show the system efficiencies of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches on AlexNet for the bearing datasets [27]–[29]. While applying the 2D-CNN with lightweight encryption on AlexNet, the expected computation time is considerably shortened to $\frac{1}{94}$ for IoT devices.

With this achievement, resource-constrained IoT devices can save more power consumption to extend the lifetime of the battery. As shown in Table VII, VIII, and IX, the edge device so-called a dedicated laptop can operate the process of data encryption efficiently while selecting the layer of CNN to offload the computing tasks to the control center. In practice, the proposed 2D-CNN with lightweight encryption achieves better privacy protection than other privacy preserving approaches since it has independent modules to customize different convolutional and fully-connected layers to extract the significant features (i.e., F-IR, F-OR, and F-RE) of the bearing datasets including IMS-Rexnord, CWRU, and Paderborn. In addition, the proposed lightweight encryption and other privacy-preserving approaches fed the unencrypted data on the edge computing device to compare the execution of privacy preserving in the dedicated 2D-CNN classifier model. The experimental results demonstrate that the proposed lightweight encryption accelerates the processing capacity of IoT devices (i.e., IMS-Rexnord 38.91×, CWRU 40.73×, and Paderborn 39.41×) much faster than the other existing approaches [16]–[19].

Especially, in practical scenarios, the execution times of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches on AlexNet are more consistent, since the dimension of the ciphertext is the same as the plaintext. Since the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches

TABLE VII: System efficiencies of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches on AlexNet on the IMS-Rexnord Dataset[27]

Layers	IoT without Task Offloading (sec)	IMS-Rexnord Bearing Dataset[27]									
		Proposed 2D-CNN with Lightweight Encryption		[16]		[17]		[18]		[19]	
		Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed
Conv [1]	14.01	0.819	13.87 ×	0.931	11.76 ×	0.992	11.87 ×	0.987	11.42 ×	0.941	12.91 ×
Conv [2]	45.67	0.469	68.18 ×	0.519	62.13 ×	0.527	61.32 ×	0.512	63.9 ×	0.533	61.22 ×
Conv [3]	21.93	0.216	72.81 ×	0.292	67.21 ×	0.297	66.31 ×	0.247	63.32 ×	0.261	69.71 ×
Conv [4]	33.65	0.371	95.27 ×	0.317	84.26 ×	0.333	87.62 ×	0.331	87.77 ×	0.319	87.15 ×
FC [1]	3.61	0.071	83.72 ×	0.057	82.79 ×	0.063	82.27 ×	0.069	82.23 ×	0.059	81.24 ×
FC [2]	2.25	0.041	67.13 ×	0.031	63.2 ×	0.037	64.8 ×	0.035	64.13 ×	0.029	62.77 ×
Non-Linear	2.91	1.87	NA	1.21	NA	1.46	NA	1.33	NA	1.39	NA
Total Cost	124.03	3.857	38.91 ×	3.357	33.61 ×	3.68	37.7 ×	3.51	35.21 ×	3.53	35.47 ×

TABLE VIII: System efficiencies of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches on AlexNet on the CWRU Bearing Dataset[28]

Layers	IoT without Task Offloading (sec)	CWRU Bearing Dataset[28]									
		Proposed 2D-CNN with Lightweight Encryption		[16]		[17]		[18]		[19]	
		Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed
Conv [1]	13.21	0.897	14.17 ×	0.977	12.43 ×	0.997	12.13 ×	0.959	12.03 ×	0.982	12.07 ×
Conv [2]	48.91	0.411	69.12 ×	0.557	61.11 ×	0.578	61.72 ×	0.537	60.9 ×	0.501	60.06 ×
Conv [3]	23.67	0.227	73.19 ×	0.287	68.61 ×	0.261	68.91 ×	0.289	64.02 ×	0.279	68.11 ×
Conv [4]	35.13	0.411	97.23 ×	0.381	89.34 ×	0.354	88.9 ×	0.387	89.07 ×	0.398	88.97 ×
FC [1]	4.81	0.092	88.14 ×	0.061	81.27 ×	0.069	82.02 ×	0.073	82.73 ×	0.069	83.97 ×
FC [2]	3.55	0.049	69.56 ×	0.037	62.17 ×	0.032	63.12 ×	0.039	63.71 ×	0.031	62.07 ×
Non-Linear	3.11	1.97	NA	1.33	NA	1.52	NA	1.47	NA	1.39	NA
Total Cost	132.39	4.057	40.73 ×	3.63	36.79 ×	3.81	38.47 ×	3.75	37.72 ×	3.65	36.93 ×

TABLE IX: System efficiencies of the proposed 2D-CNN with lightweight encryption and other privacy-preserving approaches on AlexNet on the Paderborn Bearing Dataset[29]

Layers	IoT without Task Offloading (sec)	Paderborn Bearing Dataset[29]									
		Proposed 2D-CNN with Lightweight Encryption		[16]		[17]		[18]		[19]	
		Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed	Total Processing Time (sec)	Speed
Conv [1]	16.42	0.857	14.92 ×	0.935	13.3 ×	0.941	13.41 ×	0.953	13.68 ×	0.981	13.94 ×
Conv [2]	50.37	0.472	71.67 ×	0.513	61.45 ×	0.519	61.59 ×	0.543	61.92 ×	0.507	60.52 ×
Conv [3]	25.41	0.291	75.01 ×	0.242	67.21 ×	0.259	67.49 ×	0.264	67.92 ×	0.237	68.05 ×
Conv [4]	36.54	0.474	98.41 ×	0.357	91.23 ×	0.368	91.87 ×	0.391	91.93 ×	0.341	91.18 ×
FC [1]	5.27	0.093	89.67 ×	0.071	83.43 ×	0.074	83.97 ×	0.079	84.19 ×	0.061	81.4 ×
FC [2]	2.87	0.039	71.34 ×	0.033	64.56 ×	0.036	64.87 ×	0.031	61.98 ×	0.032	63.28 ×
Non-Linear	3.73	1.69	NA	1.27	NA	1.12	NA	1.47	NA	1.23	NA
Total Cost	140.61	3.91	39.41 ×	3.42	34.67 ×	3.32	33.69 ×	3.731	37.85 ×	3.39	33.97 ×

on AlexNet employ device connection between IoT and edge, an additional metric so-called processing cost is essential to assess the latency associated with data encryption. Therefore, the system testbed utilizes a dedicated wireless channel with a transmission rate of $10MB$ between IoT and edge devices. It is worth noting that in real-time practice, the computing devices like IoT and Edge may establish their connection via wired or wireless channels with higher transmission rates to assess the latency associated with the processing data. As shown in Table VI, the processing time on AlexNet is merely about $\frac{1}{53}$ as compared to the AlexNet without 2D-CNN with lightweight encryption. The observation results reveal that the upcoming 6G era can apply this mechanism to improve the transmission rate of the application system in order to optimize the network performance.

From Table VII, VIII, and IX, it is evident that the proposed 2D-CNN with lightweight encryption achieves faster execution speeds $\approx$ 38.91 $\times$ to 40.73$\times$ for the given datasets (i.e., IMS-Rexnord, CWRU, and Paderborn). To ensure scalability, Table X, XI, and XII are illustrated which show the processing time of the proposed 2D-CNN with lightweight encryption and other existing approaches over the convolution and fully connected layers with/without privacy protection. The experimental results show that the proposed 2D-CNN with lightweight encryption achieves better processing costs $0.0992sec$, $0.0946sec$, and $0.111sec$ over the execution of AlexNet requests than other existing privacy-preserving approaches (for the bearing datasets i.e., MS-Rexnord, CWRU, and Paderborn). In addition, while the integrity check was turned on, the results show that the proposed 2D-CNN with lightweight encryption still achieves better cost efficiencies $0.0925sec$, $0.094sec$, and $0.105sec$ than other privacy-preserving approaches. The above comparative results can even validate the scope of scalability for the proposed 2D-CNN with lightweight encryption in real-time scenarios. The practical evidence reveals that the proposed 2D-CNN with lightweight encryption preserves or enhances cost efficiencies while the complexity of the convolution and fully connected layers increase proportionally. Above all, considering the AlexNet, the proposed 2D-CNN with lightweight encryption achieves a better transmission rate than other existing approaches to provide a promising solution of scalability regardless of its complex CNN architecture.

TABLE X: Integrity check over the convolution and fully connected layers with and without privacy protection on AlexNet via IMS-Rexnord Bearing Dataset [27]

Layers	IMS-Rexnord Bearing Dataset[27]									
	Proposed 2D-CNN with Lightweight Encryption		[16]		[17]		[18]		[19]	
	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)
Conv [1]	0.017	0.013	0.023	0.027	0.027	0.033	0.031	0.035	0.025	0.029
Conv [2]	0.0503	0.0472	0.0513	0.0519	0.0591	0.0579	0.0539	0.0532	0.0571	0.0563
Conv [3]	0.012	0.013	0.013	0.015	0.017	0.019	0.021	0.020	0.015	0.017
Conv [4]	0.0143	0.014	0.0154	0.0156	0.0163	0.0164	0.0169	0.0168	0.0171	0.0173
FC [1]	0.0037	0.0033	0.0043	0.0041	0.0051	0.0053	0.0057	0.0058	0.0047	0.046
FC [2]	0.0019	0.002	0.0023	0.0025	0.0021	0.0022	0.0027	0.0028	0.0024	0.0025
Total Cost	0.0992	0.0925	0.1093	0.116	0.1266	0.256	0.114	0.134	0.121	0.168

TABLE XI: Integrity check over the convolution and fully connected layers with and without privacy protection on AlexNet via CWRU Bearing Dataset[28]

Layers	CWRU Bearing Dataset[28]									
	Proposed 2D-CNN with Lightweight Encryption		[16]		[17]		[18]		[19]	
	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)
Conv [1]	0.015	0.014	0.021	0.024	0.023	0.027	0.029	0.031	0.024	0.028
Conv [2]	0.0501	0.0491	0.0507	0.0511	0.0571	0.0576	0.0527	0.0531	0.0561	0.0562
Conv [3]	0.011	0.012	0.014	0.014	0.016	0.017	0.019	0.021	0.023	0.022
Conv [4]	0.0137	0.0138	0.0149	0.0151	0.0157	0.0159	0.0161	0.0163	0.0165	0.0167
FC [1]	0.0031	0.0032	0.0042	0.0041	0.0045	0.0047	0.0049	0.0051	0.0053	0.054
FC [2]	0.0017	0.0019	0.0021	0.0023	0.0025	0.0026	0.0029	0.0029	0.0031	0.0032
Total Cost	0.0946	0.094	0.145	0.119	0.111	0.119	0.125	0.129	0.128	0.180

D. Observation with Prediction Models

This section shows the prediction scenario of the bearing datasets (MS-Rexnord, CWRU, and Paderborn) using CryptoNets and LoLa. The scenario applies a core operation of matrix-vector multiplications in a neural network to contain the weights or vector values attributed to convolution or fully connected layers. Each core operates the vectors in a different form to produce another representation of the outputs like a convolution vector (row-major). Using this technique, the convolution layer transforms the location of the data vectors which use kernel mapping to repeat the process n times. Moreover, the layer uses $\sum w_j.m^j$ to compute a set of output vectors using r operators (i.e., multiplications and additions), where w_j is a 2×2 i.e., $\{w_{1,1},w_{1,2},w_{2,1},w_{2,2}\}$ and m^j is the convolution representation for r messages. In general, unencrypted weights use BFV support (i.e.,

integer arithmetic) to perform relatively cheaper operations in order to multiply the messages using the scalar. Considering this operation, the bearing datasets and their feature vectors (i.e., F-IR, F-OR, and F-RE) use SIMD (single instruction multiple data) supported by CryptoNets to represent their required pixels in the message form. As a result, the bearing datasets (MS-Rexnord, CWRU, and Paderborn) assign a unique array of 32 pixels which make 784 and 25 messages as the input to the prediction models (CryptoNets and LoLa).

To compare with relevant privacy-preserving approaches [16]–[19] using the same scale value, the prediction models including CryptoNets and LoLa shared the same network settings using AlexNets and YASHE cryptosystems. Using the same network settings, the prediction models evaluated the neural networks (i.e., convolution and fully connected layers) with predefined sets of strides and activation functions to feed the layers with output neurons. Table XIII shows the efficiencies of the prediction models with AlexNets, 2D-CNN with lightweight encryption, and other privacy preserving approaches on bearing datasets (MS-Rexnord, CWRU, and Paderborn). Since the prediction models (CryptoNets and LoLa) used large sizes of input on AlexNet, the models preferred polynomial computation to overcome the issue of multiplication overflow. Thus, it is evident that the models utilized a high degree polynomial to rationalize the computation roots in order to determine the cost of each convolution layer. The cross-examination of the processing requests reveals that the prediction models with AlexNet and other privacy preserving approaches hold $\approx$12 to 18*minutes* to operate the first convolution layer. In consequence, this operational cost may exceed further while the other convolution layers are executed in parallel to determine the execution time of the computational parameters (i.e., encryption, convolution, and decryption) to compute a few significant metrics such as accuracy and latency.

Finally, the experimental results show that the proposed 2D-CNN with lightweight encryption attains better cost efficiencies i.e., encryption $\approx$0.014 to 0.019*sec*, convolution $\approx$0.0127 to 0.0132*sec*, decryption $\approx$0.21 to 0.31*sec*, accuracy $\approx$97.95% to 98.56%, and latency $\approx$2.6 to 2.8*sec*, respectively to process the requests into batches in order to minimize the cost of computation of each inferential request. Although the subsequent operation develops a few additional costs to collect and generate the request into batches in this time-sensitive scenario, the proposed 2D-CNN with lightweight encryption achieves better computational performance than other privacy preserving approaches.

VI. Discussion and Limitation

To address two major limitation factors namely noise handling and growth of message size, in this study, the growths of both noise and message size are solved using lightweight encryption based on the support of CNN inference with privacy preserving categorization over encrypted data. The proposed idea utilizes Feldmans' verifiable secret sharing to generate the keys without any trusted authorities whereby the connected devices like IoT can securely offload the inference tasks to speed up the system trade-off including computation and communication. Meanwhile, the function interface utilized the prediction models (i.e., CryptoNets) to guarantee effective encryption over the representation of features (F-IR, F-OR, and F-RE) in order to perform fault diagnosis via a small portion of the activation function. This systematic approach gains several technical merits: 1. Can deal with a large image source, even if the representation is limited in range; 2. Can obtain higher accuracy using the shallow network as compared to linear predictor where the predictor acts as a translator to speed up the evaluation with the given homomorphic encryption; and Can utilize the activation function with the limited range using low-degree polynomial to add a normalized layer in advance of batch normalization associated with the non-linear layer.

TABLE XII: Integrity check over the convolution and fully connected layers with and without privacy protection on AlexNet via Paderborn Bearing Dataset[29]

Layers	Paderborn Bearing Dataset[29]									
	Proposed 2D-CNN with Lightweight Encryption		[16]		[17]		[18]		[19]	
	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)	With Privacy (sec)	Without Privacy (sec)
Conv [1]	0.021	0.019	0.027	0.026	0.031	0.029	0.030	0.029	0.026	0.025
Conv [2]	0.0519	0.0503	0.0519	0.0521	0.0597	0.0593	0.0589	0.0587	0.0599	0.0596
Conv [3]	0.017	0.015	0.019	0.017	0.021	0.018	0.027	0.025	0.029	0.031
Conv [4]	0.0146	0.0144	0.0159	0.0157	0.0161	0.0160	0.0173	0.0177	0.0181	0.0179
FC [1]	0.0039	0.0037	0.0047	0.0043	0.0057	0.0054	0.0061	0.0059	0.0067	0.066
FC [2]	0.0021	0.0023	0.0025	0.0024	0.0029	0.0028	0.0033	0.0032	0.0037	0.0035
Total Cost	0.111	0.105	0.121	0.118	0.136	0.131	0.143	0.140	0.143	0.203

Nevertheless, in [39], a privacy preserving classification was introduced using a deep neural network to protect device privacy. However, this classification technique consumes more computation and communication costs while testing the scenario using CryptoNets. Moreover, the existing approaches [16]–[19] cannot process any data instinctively to provide any private prediction while preserving the privacy of the computing device like IoT. Therefore, in this paper, privacy-preserving learning is integrated with lightweight encryption to demonstrate the practical feasibility based on offloading solutions in time-sensitive scenarios like supply-chain optimization.

VII. Conclusion

To improve the effectiveness of privacy assurance, particularly in categorizing the fault categories using the prediction models (i.e., CryptoNet and LoLa) without any trusted third-party, we introduce privacy-preserving learning (PPL) with lightweight encryption. This devised strategy applies local update and model aggregation on the edge computing systems, including devices and servers using the 2D-CNN model, to optimize the learning features of the VSI-IoT framework whereby the features of the machinery defects, i.e., F-IR, F-OR, and F-RE, were precisely extracted to detect the faults at an early stage. The two-stage models, including the proposed 2D-CNN+BN and 2D-CNN+LN, use a nonlinear function to examine the beaming failures and test their robustness with two levels of severity called moderate and severe. Moreover, the proposed 2D-CNN+BN and 2D-CNN+LN can even handle sensitive information via PPL to examine the key features of any automation system, which minimizes the processing time between the edge device and server. Using formal security analysis, we prove that the proposed PPL can guarantee data privacy and model accuracy to protect the server access without extra verifiability. Lastly, the experiment result shows that the proposed 2D-CNN with BN and LN achieves a better accuracy ($\approx$92.9% to $\approx$96.3%) and communication cost ($\approx$23.17 to 23.51*MB*) than other existing models. In the future, the potential features of blockchain technology, such as decentralization, security, and immutability, will be explored to develop a seamless lightweight framework with adaptive feature learning, which can quickly investigate the learning models to offload the computing resources in Industrial IoT networks.

References

[1] M. Ali, J.-S. Pan, S.-M. Chen, and M.-F. Horng, "Modern "advances in applied intelligence"," in *27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE*, Springer, 2014, pp. 3–6.

[2] S. Roy, P. Pranav, and V. Bhattacharjee, "Securing the internet of things: Current and future state of the art," *Smart Healthcare Analytics in IoT Enabled Environment*, p. 227, 2020.

TABLE XIII: Efficiencies of the prediction models with AlexNets, 2D-CNN with lightweight encryption, and other privacy preserving approaches on Bearing Dataset[27]–[29]

Parameters	IMS-Rexnord Bearing Dataset[27]						
	Proposed 2D-CNN with Lightweight Encryption	With CryptoNet	With LoLa	[16]	[17]	[18]	[19]
Encryption (sec)	0.014	0.365	0.216	0.571	0.597	0.465	0.643
Convolution (sec)	0.0127	647.921	413.33	789.12	723.61	712.65	809.17
Decryption (sec)	0.21	0.291	0.231	0.378	0.392	0.411	0.457
Accuracy	98.31%	93.46%	94.35%	93.65%	93.85%	91.95%	94.25%
Latency (sec)	2.7	235	2.6	3.6	3.8	3.4	4.2

(a) IMS-Rexnord

Parameters	CWRU Bearing Dataset[28]						
	Proposed 2D-CNN with Lightweight Encryption	With CryptoNet	With LoLa	[16]	[17]	[18]	[19]
Encryption (sec)	0.013	0.316	0.254	0.432	0.478	0.416	0.518
Convolution (sec)	0.0123	617.127	386.18	672.28	696.34	624.8	756.14
Decryption (sec)	0.27	0.371	0.31	0.397	0.421	0.407	0.428
Accuracy	98.56%	94.68%	94.97%	94.71%	94.57%	93.16%	94.66%
Latency (sec)	2.6	247	2.8	3.9	4.2	4.1	4.6

(b) CWRU

Parameters	Paderborn Bearing Dataset[29]						
	Proposed 2D-CNN with Lightweight Encryption	With CryptoNet	With LoLa	[16]	[17]	[18]	[19]
Encryption (sec)	0.019	0.412	0.356	0.504	0.546	0.522	0.584
Convolution (sec)	0.0132	812.413	488.212	736.24	758.36	716.24	794.56
Decryption (sec)	0.31	0.387	0.362	0.436	0.478	0.454	0.521
Accuracy	97.95%	91.35%	95.7%	93.92%	93.47%	93.42%	92.95%
Latency (sec)	2.8	296	3.2	4.1	4.3	4.2	4.7

(c) Paderborn

[3] Q. Zhang, C. Xin, and H. Wu, "Privacy-preserving deep learning based on multiparty secure computation: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 412–10 429, 2021.

[4] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*, Springer, 2017, pp. 100–110.

[5] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1486–1500, 2019.

[6] J. Park and H. Lim, "Privacy-preserving federated learning using homomorphic encryption," *Applied Sciences*, vol. 12, no. 2, p. 734, 2022.

[7] D. Dhinakaran, S. Sankar, D. Selvaraj, and S. E. Raja, "Privacy-preserving data in iot-based cloud systems: A comprehensive survey with ai integration," *arXiv preprint arXiv:2401.00794*, 2024.

[8] J.-W. Lee, H. Kang, Y. Lee, *et al.*, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30 039–30 054, 2022.

[9] L. Zhang, J. Xu, P. Vijayakumar, P. K. Sharma, and U. Ghosh, "Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system," *IEEE Transactions on Network Science and Engineering*, 2022.

[10] J.-W. Lee, H. Kang, Y. Lee, *et al.*, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30 039–30 054, 2022.

[11] X. Zhang, X. Chen, J. K. Liu, and Y. Xiang, "Deeppar and deepdpa: Privacy preserving and asynchronous deep learning for industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2081–2090, 2019.

[12] X. Liu, Y. Zheng, X. Yuan, and X. Yi, "Securely outsourcing neural network inference to the cloud with lightweight techniques," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 620–636, 2022.

[13] T. Li, J. Li, X. Chen, Z. Liu, W. Lou, and Y. T. Hou, "Npmml: A framework for non-interactive privacy-preserving multi-party machine learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 6, pp. 2969–2982, 2020.

[14] Y. Tian, L. Njilla, J. Yuan, and S. Yu, "Low-latency privacy-preserving outsourcing of deep neural network inference," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3300–3309, 2020.

[15] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "Ppfa: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[16] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10 782–10 793, 2020.

[17] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[18] J. Zhang, Y. Zhao, J. Wang, and B. Chen, "Fedmec: Improving efficiency of differentially private federated learning via mobile edge computing," *Mobile Networks and Applications*, vol. 25, no. 6, pp. 2421–2433, 2020.

[19] G. Liu, C. Wang, X. Ma, and Y. Yang, "Keep your data locally: Federated-learning-based data privacy preservation in edge computing," *IEEE Network*, vol. 35, no. 2, pp. 60–66, 2021.

[20] Z. Huang, "Hybrid device-to-device and device-to-vehicle networks for energy-efficient emergency communications," *IEEE Systems Journal*, 2023.

[21] J.-N. Liu, J. Weng, A. Yang, Y. Chen, and X. Lin, "Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 247–257, 2019.

[22] C. Wang, Y. Xu, H. Liang, W. Huang, and L. Zhang, "Woody: A post-process method for smartphone-based activity recognition," *IEEE Access*, vol. 6, pp. 49 611–49 625, 2018.

[23] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.

[24] S. Hu, J. Li, Q. Zhao, C. Zhang, Z. Zhang, and Y. Shi, "Blockdl: Privacy-preserving and crowd-sourced deep learning through blockchain," in *2021 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2021, pp. 1–7.

[25] H. Tran-Dang and D.-S. Kim, "Frato: Fog resource based adaptive task offloading for delay-minimizing iot service provisioning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 10, pp. 2491–2508, 2021.

[26] D. Zhang and B. Wei, *Smart sensors and devices in artificial intelligence*, 2020.

[27] J. Chuya-Sumba, L. M. Alonso-Valerdi, and D. I. Ibarra-Zarate, "Deep-learning method based on 1d convolutional neural network for intelligent fault diagnosis of rotating machines," *Applied Sciences*, vol. 12, no. 4, p. 2158, 2022.

[28] I. Mukherjee and S. Tallur, "Light-weight cnn enabled edge-based framework for machine health diagnosis," *IEEE Access*, vol. 9, pp. 84 375–84 386, 2021.

[29] C. Zhang, Z. Xiao, and Z. Sheng, "A bearing fault diagnosis method based on a convolutional spiking neural network with spatial–temporal feature-extraction capability," *Transportation Safety and Environment*, vol. 5, no. 2, tdac050, 2023.

[30] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Network*, vol. 34, no. 3, pp. 50–56, 2020.

[31] M. Sajjad, Z. A. Khan, A. Ullah, *et al.*, "A novel cnn-gru-based hybrid approach for short-term residential load forecasting," *Ieee Access*, vol. 8, pp. 143 759–143 768, 2020.

[32] B. Chen, L. Wu, H. Wang, L. Zhou, and D. He, "A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5813–5825, 2019.

[33] B. Applebaum, E. Kachlon, and A. Patra, "Verifiable relation sharing and multi-verifier zero-knowledge in two rounds: Trading nizks with honest majority," *Cryptology ePrint Archive*, 2022.

[34] D. J. Bernstein, T. Lange, and P. Schwabe, "On the correct use of the negation map in the pollard rho method," in *International Workshop on Public Key Cryptography*, Springer, 2011, pp. 128–146.

[35] A. Nauman, Y. A. Qadri, M. Amjad, Y. B. Zikria, M. K. Afzal, and S. W. Kim, "Multimedia internet of things: A comprehensive survey," *IEEE Access*, vol. 8, pp. 8202–8250, 2020.

[36] A. A. Shah, N. A. Bhatti, K. Dev, and B. S. Chowdhry, "Muhafiz: Iot-based track recording vehicle for the damage analysis of the railway track," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9397–9406, 2021.

[37] G. Oded, *Foundations of cryptography: Volume 2, basic applications*, 2009.

[38] Y. Aono, T. Hayashi, L. Trieu Phong, and L. Wang, "Efficient key-rotatable and security-updatable homomorphic encryption," in *Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing*, 2017, pp. 35–42.

[39] H. Chabanne, A. De Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network," *Cryptology ePrint Archive*, 2017.