Journal of Ambient Intelligence and Humanized Computing (2023) 14:4659–4674
https://doi.org/10.1007/s12652-022-04367-x

ORIGINAL RESEARCH

A differential approach and deep neural network based data privacy-preserving model in cloud environment

Rishabh Gupta[1] · Ishu Gupta[2] · Deepika Saxena[1] · Ashutosh Kumar Singh[1]

Received: 20 December 2021 / Accepted: 30 July 2022 / Published online: 26 September 2022

Abstract
Data outsourcing has become indispensable to allow information sharing among multiple parties. The users do not fully trust the cloud platform since it is operated by a third party. Preserving privacy while sharing the data among different parties is a challenging task; therefore, users apply the differential privacy mechanism to protect their data. However, such protection mechanisms suffer from the problem of degradation of learning results. In this paper, the authors address the degradation of the learning results due to noise injection into user's data through ϵ-differential privacy. A differential approach and deep neural network based data privacy-preserving model is proposed, which injects noise at an appropriate position by exploiting the properties of the Laplace transform to maintain the accuracy level. The experiments are conducted over Steel Plates Fault, Spambase, Banknote Authentication, and Monk Problem datasets for deep neural network classifier to evaluate the model's efficiency in accuracy, precision, recall, and F1-score terms. The achieved results show that the proposed model ensures high accuracy, precision, recall, and F1-score up to 99.75%, 99.72%, 99.72%, and 99.72%, and improvement up to 19.34%, 30.67%, 29.39%, and 32.11%, respectively, as compared to existing approaches.

Keywords Privacy preservation · Deep neural network · Optimization · Differential privacy · Cloud computing

1 Introduction

With the rapid cloud computing usage, numerous applications and data are shifting from local machines to cloud servers for storage and computation (Li et al. 2018a). The sensitive data is essential to be shared among other parties for effective utility. On the other hand, users are reluctant to share their data with the cloud for analysis and storage (Anand and Singh 2022). Since a third party handles it Saxena and Singh (2020), Singh and Kumar (2019), and data might be misused, as well as data users lose control of their data (Manimuthu and Murugaboopathi 2021; Shen et al. 2018). According to a Cisco survey, 76% of owners have no clue about their data utilization by other parties (Cisco 2021). Other organizations may acquire outsourced data from the cloud for various purposes (Ali et al. 2015), which does not ensure that data will not leak after it has been received (Hauer 2015). Due to these reasons, data protection has become a vital problem for any organization. As a result, a data protection mechanism is needed that can preserve the privacy of sensitive data.

The two major techniques such as cryptography and differential privacy, are used to protect data for privacy purposes before uploading it to the cloud platform (Wei et al. 2016; Zaghloul et al. 2019; Sandhia and Raja 2021). However, a few models (Gong et al. 2020; Li et al. 2018b; Gupta et al. 2020) are also reported that combine differential privacy and encryption techniques to preserve data privacy in the cloud environment. The deep neural network is playing a key role in data classification, which further helps in preserving privacy and security (Wang et al. 2020). Cryptography is efficient for data privacy (Singh and Singh 2022), but it becomes cumbersome to perform the computation over the encrypted data, and

✉ Ishu Gupta
ishugupta23@gmail.com

Rishabh Gupta
rishabhgpt66@gmail.com

Deepika Saxena
13deepikasaxena@gmail.com

Ashutosh Kumar Singh
ashutosh@nitkkr.ac.in

[1] Department of Computer Applications, National Institute of Technology, Kurukshetra, Haryana, India

[2] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan

data is compromised once the key is leaked (Zhangjie et al. 2018). In differential privacy technique, noise is injected into original data to preserve privacy. But, it degrades the accuracy of preserved data classification compared to the non-privacy preserving counterparts (Li et al. 2019). The reason for degradation is the degree of noise and the position at which noise is added to the user's data. To address the challenges mentioned above, we propose a Differential approach and Deep neural network based data Privacy-Preserving Model (DD-PPM) for data protection through storage, analysis, and secure sharing in the cloud environment.

1.1 Our contributions

The key contributions of DD-PPM are as follows:

- A model for preserving the privacy of the data is proposed, which injects noise at an appropriate position by exploiting the properties of the Laplace transform to maintain the accuracy level.
- An optimized noise injective mechanism is proposed that reduces the impact of perturbation while protecting the data.
- DD-PPM transforms the actual data into the synthetic form, which is used to share and perform the classification tasks to enhance the computation's efficiency and usage.
- Implementation and performance evaluation is carried out using multiple data sets, and a comparison with existing works is performed to prove the efficiency of the proposed model.

The rest of this paper proceeds as follows. Section 2 describes the related work. An overview of the proposed model DD-PPM is presented in Sect. 3, followed by the privacy-preserving of data in Sect. 4. The data classification steps and illustration of the proposed model are defined in Sects. 5 and 6. The operational design and computational complexity are described in Sect. 7. The experimental results of DD-PPM are entailed in Sect. 8. Section 9 shows the proposed model's features, statistical, and security analysis. Finally, the proposed work with the direction for future work is summarized in Sect. 10. Table 1 shows a list of notations used in this paper with their definitions.

Table 1 Notations and their descriptions

CO	Classifier owner	DA	Data analyst
UE	Untrusted entity	D_i	Actual data
CM	Classification model	DU_{id}	Data users
CA	Classification accuracy	$\hat{D}_{t'}^{N}$	Training data
$\hat{D}_i^N$	Preprocessed data	FS	F1-Score
Δf	Absolute distance	ϵ	Privacy budget
Pr	Probability	CS	Cloud server
ϑ	Scaling parameter	t''	Testing objects
S	Random mechanism	f	Query function
t'	Training objects	$\hat{D}_{t''}^{N}$	Testing data
P	Precision	N_i	Noise
D_i^N	Synthetic data	R	Recall
σ	Standard deviation	μ	Mean

2 Related work

2.1 Cryptographic approach

A basic scheme based on multi-key fully homomorphic encryption (MK-FHE) mechanism was introduced in Li et al. (2017). In this scheme, an advanced model was developed based on the fully homomorphic encryption (FHE) and the double decryption mechanism to learn encrypted data in the cloud. Yonetani et al. (2017) proposed a doubly permuted homomorphic encryption (DPHE) based privacy-preserving framework to learn visual classifiers securely over distributed private data. It enabled multiparty protected scalar products with reduced computational cost. A secure outsourcing scheme for a classifier owner to delegate a remote server and provide privacy-preserving classification services for users is presented in Li et al. (2018c). CryptoDL is a system proposed by Hesamifard et al. (2018) for applying deep neural network algorithms over encrypted data. They established neural network techniques while considering the existing limitations of homomorphic encryption schemes. A privacy-preserving outsourced classification in cloud computing (POCC) framework to support the arbitrary number of multiplication and addition operations on ciphertexts securely is presented in Li et al. (2018d). It protects sensitive data and queries using fully homomorphic encryption based on Gentry's scheme. Gao et al. (2018) proposed a privacy-preserving Naive Bayes classifier scheme to avoid information leakage under the substitution-then-comparison (STC) attack. A privacy-preserving classification mechanism was designed by adopting a double-blinding technique and reduced the communication and computation overhead. To train the model over the multi-key encrypted data, a privacy-preserving deep learning model, namely PDLM, is presented in Ma et al. (2018). The model based on stochastic gradient descent (SGD) was trained using a privacy-preserving calculation toolkit in a privacy-preserving manner. PDLM reduced the storage overhead. Hassan et al. (2019) proposed a privacy-preserving machine learning scheme that enables all participants to check the encrypted data's correctness publicly.

A unidirectional proxy re-encryption (UPRE) scheme was used to minimize computational costs. Data privacy was protected using a differential privacy scheme.

2.2 Differential privacy approach

Zhang and Zhu (2016) focused on a class of regularized empirical risks to reduce the machine learning problems. The dual variable perturbation and primal variable perturbation methods were proposed to protect the distributed learning algorithms over the network. Out of these methods, the primal variable perturbation slightly outperformed the dual variable perturbation in balancing the trade-off between privacy and accuracy. The dual variable perturbation outperformed the primal variable perturbation in the performance of learning. Li et al. (2018e) proposed a data protection scheme that preserves Naive Bayes' privacy over data contributed by multiple providers. The ϵ-differential privacy was used for data protection. To preserve the privacy of SGD, Phuong (2019) proposed two systems, namely the Server-aided Network Topology (SNT) system and the Fully-connected Network Topology (FNT) system based on the connection with SNT and FNT server. The constructed systems used the weight parameters rather than the gradient parameters and achieved an accuracy similar to SGD. These systems are efficient in terms of computation & communication and effective in terms of accuracy. To preserve the privacy of learning model parameters, Wei et al. (2020) devised a noising before model aggregation federated learning (NbAFL) framework. Gong et al. (2020) proposed three privacy budget dynamic allocation strategies to mitigate the consumption and prevent privacy leakages. The homomorphic encryption scheme was used to encrypt the parameters, and the algebraic operation was performed to update the parameters in the encrypted form. The differential privacy was applied to preserve the confidentiality of data. A privacy-preserving machine learning with multiple data provider (PMLM) scheme with improved computational efficiency and data analysis was presented in Li et al. (2018b). The authors adopted public-key encryption with a double decryption algorithm (DD-PKE) to encrypt the owner's data with their public keys and ϵ-differential privacy for cloud data privacy. The encrypted data is transformed into synthetic data, which enhances the computation's accuracy and efficiency. The experiments were performed, and the results showed that the existing proposed work is more secure. A machine learning and probabilistic analysis-based model, namely MLPAM, was presented in Gupta et al. (2020). The proposed model helps the various owners to share their data safely with other entities for different purposes by using encryption, machine learning, and probabilistic approaches. A mechanism has been provided to reduce the risk associated with the leakage for prevention coupled with detection. MLPAM has achieved the secured communications protocol. Table 2 provides an overview of the literature review.

The major downfall of the existing works is that the models injected the noise into entire data and/or protected it using several encryption approaches followed by the deep neural learning-based classification, which degraded accuracy and/or increased computation cost. In previous models, there was a single owner and/or a single untrustworthy entity. Unlike prior works, DD-PPM injects noise into specific parts of data to keep them private and uses a deep neural network to do classification tasks.

3 Proposed model

The proposed architecture (Fig. 1) comprises of four entities: Data users (DU_{id}), Data analyst (DA), Cloud server (CS), and Classifier owner (CO), which are stated as follows in terms of intercommunication and vital flow of information:

1) DU_{id}: An entity that produces, stores, and computes it on the cloud platform. Before sending it to the cloud, DU_{id} passes to DA to upgrade its performance. DU_{id} is considered untrusted entity because they do not leak its own data but may disclose the data of other users.
2) *DA*: An entity that collects all the data from DU_{id} for utilization purposes. To ensure data protection, DA injects the noise before sharing it to the cloud. DA is deemed a trusted entity in the proposed model.
3) *CS*: An entity that acquires all synthetic data from DA and offers storage, computing, and sharing services to DA and CO. CS is treated as an untrusted entity in the model as it follows the protocol strictly but is curious to learn the information.
4) *CO*: An entity that acquires synthetic data from CS. CO has a classification model (CM) that performs the classification tasks over the received data from CS. In the proposed model, CO is viewed as an untrustworthy entity.

Let the data user $\mathbb{DU} = \{DU_1, DU_2, \cdots, DU_n\}$ owns data $\mathbb{D} = \{D_1, D_2, \cdots, D_n\}$, where data D_i belongs to $\mathbb{D}$ that may be of any sort or size. To enhance the utilization of data, each $DU_1, DU_2, \cdots, DU_n$ shares $D_1, D_2, \cdots, D_n$ to DA. Due to the sensitive data, DA does not aspire to reveal $\mathbb{D}$ to unauthorized parties such as CS and CO. As a result, $\mathbb{DA}$ procures synthetic data $\mathbb{D}^{\mathbb{N}} = \{D_1^{N_1}, D_2^{N_2}, \cdots, D_n^{N_n}\}$ by injecting noise $\mathbb{N} = \{N_1, N_2, \cdots, N_n\}$ into sensitive data $\{D_1, D_2, \cdots, D_n\}$ using ϵ-differential privacy to make the data private before sharing (details discussed in Sect. 4). DD-PPM utilizes the differential privacy mechanism to produce the noise $N_1, N_2, \cdots, N_n$, since it is the most appropriate mechanism for preventing privacy concerns

Table 2 Tabular sketch of the literature review

Model/scheme/framework	Workflow and implementation	Outcomes	Drawback
A privacy-preserving scheme for deep learning on encrypted data Li et al. (2017)	• Secure multi-party computation protocol encrypts data	• More effective because of the online phase of owners and server	High computation cost due to more bandwidth demand
	• The experiments are carried out to determine human emotions	• According to the security analysis, this scheme is more secure	
A data protection framework for visual learning Yonetani et al. (2017)	• A homomorphic cryptosystem is utilized to secure data	• Up to 89.70% accuracy is achievable	DPHE performs either addition or multiplication operations at a specific duration
	• Caltech101, Life-logging, and Flickr8k dataset were used for experiments	• Low cost and high throughput for training visual classifier	
A framework for distributing the outsourced classification service Li et al. (2018c)	• The naive bayes and hyperplane decision-based classifiers are performed over numerous datasets for experiments	• Acquire high-level privacy for outsourced classifier	Deep learning's intricate computations make multiparty computing difficult to expand
	•The classifier is protected by adopting a private information retrieve protocol	• More practical and less overhead in communication	
A framework for using deep neural network methods to secure data Hesamifard et al. (2018)	• For continuous activation functions, a polynomial approximation was applied	• Provide high accuracy up to 94.20%	High computation cost due to homomorphic encryption
	• MNIST and CIFAR-10 datasets were used for the experiments	• More feasible to train neural networks over encrypted data	
A secure outsourced classification framework for sharing the data Li et al. (2018d)	• A fully homomorphic encryption approach is used to encrypt data	• More effective data utilization	High computational overhead because of fully homomorphic encryption
	• The experiments are carried out using the naive bayes classifier	• This framework safely trains a classification model according to the security analysis	
An efficient and privacy-preserving mechanism for Naive Bayes classifier Gao et al. (2018)	• Data was protected with the additive homomorphic encryption algorithm	• Less communication cost because most analyses are offline	This scheme is unable to achieve truth that protects privacy
	• The experiments were performed in C++ language over Breast Cancer Wisconsin and Statlog Heart datasets	• More efficient at retaining sensitive data safe from malicious servers	
A cloud-based deep learning model for preserving the data privacy Ma et al. (2018)	• To protect training data, a public-key cryptosystem with distributed two trapdoors was utilized	• Achieve high accuracy up to 91.61%	Due to the usage of two trapdoors, the public-key cryptosystem has a high computation cost
	• LeNet deep learning evaluated model's efficiency using Torch7 nn packages	• More effectively train the model while retaining privacy	
A data protection scheme based on additive homomorphic encryption Hassan et al. (2019)	• ElGamal and differential privacy techniques are used to protect data	• UPRE reduces communication costs	The scheme is ineffective and insecure
	• This scheme is evaluated using the Java pairing-based cryptography library	• Improve the accuracy because employing differential privacy	
A distributed machine learning algorithm for data protection Zhang and Zhu (2016)	• The training data was distributed across the network using the alternate direction method	• Reduce machine learning issues through regularised empirical risk	Increase the communication overhead due to the dynamic privacy budget
	• The experiments were carried out using a logistic regression algorithm	• Make a balance between privacy and accuracy	

Table 2 (continued)

Model/scheme/framework	Workflow and implementation	Outcomes	Drawback
A Naive Bayes learning scheme for data privacy Li et al. (2018e)	• To protect training tasks, differential privacy and homomorphic encryption were adopted	• Paillier decryption operation reduces computation costs	It does not satisfy the differential privacy in the local setting and preserves individual privacy with encryption technique
	• The experiments were performed over Balance Scale, Breast Cancer Wisconsin, and SPECT Heart datasets	• The security analysis proves that the scheme is more practical	
A multilayered neural network driven framework for sharing encrypted network weights Phuong (2019)	• Symmetric encryption was adopted for input data security	• Maintain the deep learning's value in terms of accuracy	It does not consider the output's privacy
	• This framework was assessed using Breast Cancer, Skin/NonSkin, and MNIST datasets	• This framework is more efficient in terms of computing according to security analysis	
A differential privacy driven framework for federated learning protection Wei et al. (2020)	• To inject noise into learning parameters, the Gaussian mechanism was applied	• Up to 95.53% accuracy is achievable	Trade-off between privacy and FL performance
	• The experiments are carried out using ReLU and softmax activation functions on feed-forward neural network	• Reduce the leakage's risk in terms of prevention and detection	
A multi-party deep learning framework for protecting the model's parameters Gong et al. (2020)	• Additive homomorphic encryption and differential privacy were combined to mitigate privacy breaches	• Achieve a trade-off between the privacy and training efficiency	Low efficiency due to a homomorphic encryption technique
	• The framework is assessed using convolutional neural network and paillier algorithm 3 library	• Acquire high accuracy up to 96.91%	
A privacy preserving machine learning scheme for the data protection Li et al. (2018b)	• Differential privacy and double decryption were applied to protect data	• Up to 92.62% accuracy is achievable	The noise was injected in an inefficient manner, resulting in lower accuracy
	• This scheme is evaluated over Abalone, Wine, Cpu, Glass, and Krkopt data sets	• Due to differential privacy, it increases computing efficiency	
A secure data sharing model based on machine learning and probabilistic analysis Gupta et al. (2020)	• The ciphertext-policy attribute-based encryption and differential privacy were utilized to protect the data	• Achieve high accuracy up to 92.89%	It is ineffective for communicating and managing data across several environments
	• The random noise was injected into the data	• The security analysis proves that no actual data will be accessed during the transfer	

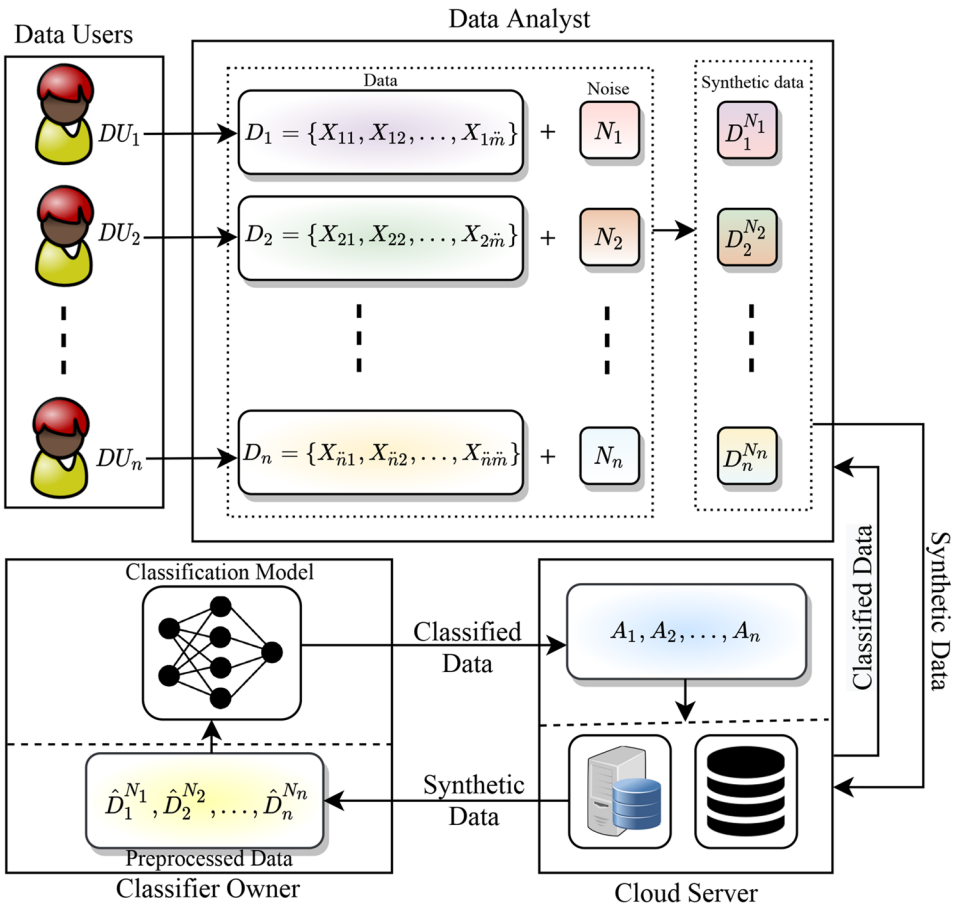


Fig. 1 DD-PPM architecture

based on auxiliary information. It injects $N_1, N_2, \cdots, N_n$ into D_1, $D_2, \cdots, D_n$, respectively. DA transfers $D_1^{N_1}, D_2^{N_2}, \cdots, D_n^{N_n}$ to CS for storage and sharing. Whenever, *CS* needs performing the classification tasks, it passes data $D_1^{N_1}, D_2^{N_2}, \cdots, D_n^{N_n}$ to CO. CO has CM and provides the classification services. DU_1, DU_2, $\cdots$, DU_n can make the query via CS that passes it to CO to perform the classification tasks over it. The achieved results from CM are delivered to related entities: DU_1, $DU_2, \cdots, DU_n$ through CS.

4 Privacy-preserving of data

Data analyst (DA) injects the noise (N_1, N_2, $\cdots$, N_n) into data (D_1, D_2, $\cdots$, D_n), respectively, using ϵ-differential privacy. Before injecting the noise, DA determines the location (*loc*), where the minimum value occurs in the instance of the data $D_1, D_2, \cdots, D_n$ using Eq. (1).

$$loc_i = g(D_i) \tag{1}$$

where $i = 1, 2, \cdots, n$ and g is a function to find location. A random function S is referred to as ϵ-differential privacy if for any combination of the dataset D and its neighbor $D^{'}$, and for all $\Omega \subseteq \text{Range}(S)$ are described using Eq. (2):

$$Pr[S(D) = \Omega] \leq exp(\epsilon) \times Pr[S(D^{'}) = \Omega] \tag{2}$$

where Pr[.] is the probability function applied to the mechanism S and presents the privacy revealing risk. The privacy budget is denoted by ϵ, which is a predefined privacy parameter. It is determined by statistical analysis results and the individual's information to keep concealed. The value of ϵ is reduced to obtain a higher level of privacy protection. A numeric query function (f) must be used to map a data set D into real space with d-dimensional, it is denoted as f: D $\rightarrow R^d$. The sensitivity of f for each combination of dataset D and $D^{'}$ is assigned using Eq. (3):

$$\Delta f = \max_{D,D^{'}} \| f(D) - f(D^{'}) \|_{Q_1} \tag{3}$$

where norm is denoted as $\| . \|_{Q_1}$. The sensitivity Δf is only dependent on the query f types. It determines the highest gap between query results on neighboring data sets. For any function f: D $\rightarrow R^d$, the Laplace mechanism F is described using Eq. (4). The mechanism F takes dataset D as input, and $\epsilon >$ o, a numeric query function f, and calculates the results. It is based on the sensitivity of f and acquires the statistical noise from a Laplace distribution. Afterward, the obtained noise is injected into the datasets.

$$F(D) = f(D) + (\text{Lap}_1(\vartheta), \text{Lap}_2(\vartheta), \ldots, \text{Lap}_d(\vartheta)) \quad (4)$$

where the noise $\text{Lap}_t(\vartheta)$ (t ∈ [1, d]), and ϑ belongs to R^+ comes from a Laplace distribution, whose the probability density function is calculated using Eq. (5).

$$N = \frac{1}{2\vartheta} \cdot \left(\exp\left(\frac{-|D|}{\vartheta} \right) \right) \quad (5)$$

where N is the noise vector. The noise $N_1, N_2, \cdots, N_n$ is produced by taking the sample from the Laplace distribution with scaling parameter $\vartheta = \Delta f/\epsilon$. The parameter ϑ is under the control of ϵ. The generated noise $N_1, N_2, \cdots, N_n$ is injected into the corresponding $D_1, D_2, \cdots, D_n$ as $D_i^{N_i} = D_{loc_i} + N_i$, where $i \in [1, n]$. *DA* sends the synthetic data $D_1^{N_1}, D_2^{N_2}, \cdots, D_n^{N_n}$ to CS that stores and shares it to CO to perform the classification task.

5 Data classification

CO preprocesses data $D_1^{N_1}, D_2^{N_2}, \cdots, D_n^{N_n}$ by using the normalization function to acquire the preprocessed data $\hat{\mathbb{D}}^{\mathbb{N}} = \{\hat{D}_1^{N_1}, \hat{D}_2^{N_2}, \cdots, \hat{D}_n^{N_n}\}$ by applying Eq. (6), where μ, and σ are the mean and the standard deviation of the training samples, respectively.

$$\hat{D}_i^N = \frac{(D_i^N - \mu_i)}{\sigma_i} \quad (6)$$

The preprocessed data $\hat{D}_1^{N_1}, \hat{D}_2^{N_2}, \cdots, \hat{D}_n^{N_n}$ is divided into training data $\hat{\mathbb{D}}_{t'}^N = \{\hat{D}_{t',1}^{N_1}, \hat{D}_{t',2}^{N_2}, \cdots, \hat{D}_{t',n'}^{N_{n'}}\}$ and testing data $\hat{\mathbb{G}}_{t''}^N = \{\hat{D}_{t'',1}^{N_1}, \hat{D}_{t'',2}^{N_2}, \cdots, \hat{D}_{t'',n''}^{N_{n''}}\}$. The training data $\hat{D}_{t',1}^{N_1}, \hat{D}_{t',2}^{N_2}, \cdots, \hat{D}_{t',n'}^{N_{n'}}$ is used to train CM with a deep neural network (DNN), while the testing data $\hat{D}_{t'',1}^{N_1}, \hat{D}_{t'',2}^{N_2}, \cdots, \hat{D}_{t'',n''}^{N_{n''}}$ is used to measure the accuracy of CM. During the testing process, data objects $\hat{D}_{t'',1}^{N_1}, \hat{D}_{t'',2}^{N_2}, \cdots, \hat{D}_{t'',n''}^{N_{n''}}$ are given to *CM* to assign class labels. Thus, *CM* evaluates $\hat{D}_{t'',1}^{N_1}, \hat{D}_{t'',2}^{N_2}, \cdots, \hat{D}_{t'',n''}^{N_{n''}}$ and provides the results as a class label vector $\mathbb{A} = \{A_1, A_2, \cdots, A_{n''}\}$. In DD-PPM, the feed-forward DNN classifier is deployed, consisting of one input layer with n nodes, two hidden layers having p nodes, and one output layer with q nodes, as shown in Fig. 2. The input layer neurons acquire preprocessed input data $\hat{D}_1^{N_1}, \hat{D}_2^{N_2}, \cdots, \hat{D}_n^{N_n}$, which is given to hidden layers represented as $H_p = \{\theta_1 \hat{D}_1^{N_1}, \theta_2 \hat{D}_2^{N_2}, \cdots, \theta_p \hat{D}_n^{N_n} + b\}$, where θ, and b are weight and bias, respectively.

Table 3 Synthetic data

Users	Λ_1	Λ_2	Λ_3	Λ_4	Λ_5
DU_1	3.6216	8.6661	−2.2187	−0.44699	0
DU_2	4.5459	8.1674	−1.6756	−1.4621	0
DU_3	0.32924	−4.0187	4.5718	−0.9888	0
DU_4	0.9124	−1.5223	2.8342	−1.1598	1
DU_5	1.4311	2.9374	−0.3698	0.6221	1
DU_6	3.2032	5.7588	0.4724	−0.6125	0
DU_7	1.5356	9.1772	−0.7336	−0.7353	0
DU_8	1.2247	8.7779	−4.4035	−0.8064	0
DU_9	3.9899	1.1038	2.3946	0.8629	1
DU_{10}	1.8993	7.6625	0.1539	−1.0175	1
DU_{11}	−1.5768	10.8430	2.5462	−2.7926	0
DU_{12}	3.4040	8.7261	−3.0030	−0.5724	0
DU_{13}	4.6765	−3.3924	3.4896	1.4771	0
DU_{14}	2.6719	3.0646	0.8747	0.5861	1
DU_{15}	0.8035	2.8473	4.3439	1.6524	1
DU_{16}	1.4479	−4.0257	8.3428	−2.1086	0
DU_{17}	5.2423	11.0272	−3.8934	−4.1013	0
DU_{18}	5.7867	7.8902	−1.3663	−0.4870	0
DU_{19}	0.3292	−4.7317	4.5718	−0.9888	1
DU_{20}	3.9362	10.1622	−3.8235	−3.5545	1

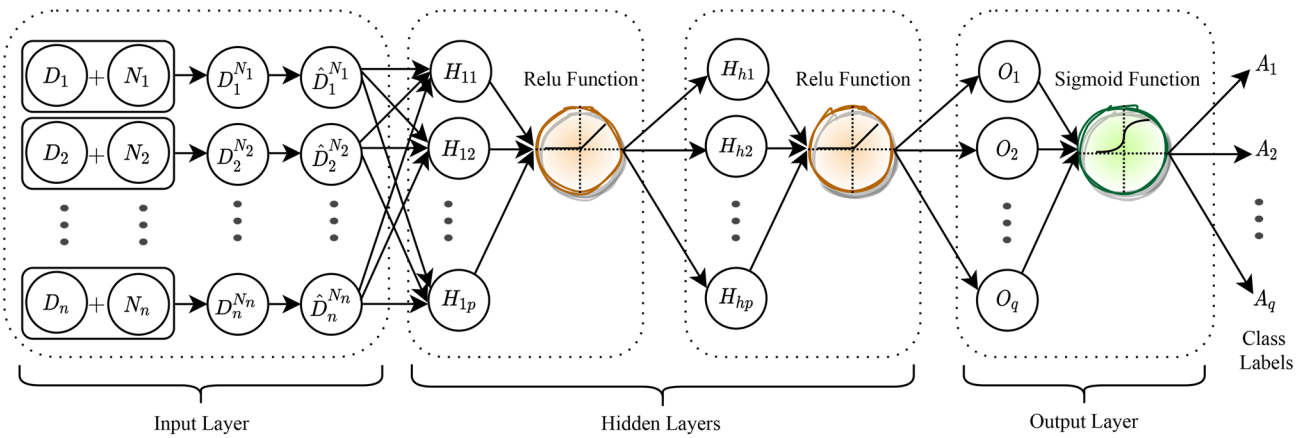


Fig. 2 DNN Classifier for shared data

The hidden layers neurons use rectified linear unit (ReLU) function f_1 by applying Eq. (7) as an activation function in the respective neural network.

$$\varphi = \max(0, \hat{D}^N) \tag{7}$$

The results of hidden layers are provided to the output layer as $\{\theta_1 h_1, \theta_2 h_2, \cdots, \theta_q h_p + b\}$, where b is the bias. The sigmoid function is used as an activation function f_2 at the output layer by applying Eq. (8).

$$\phi = \frac{e^{\hat{D}^N}}{e^{\hat{D}^N} + 1} \tag{8}$$

The classified results $A_1, A_2, \cdots, A_q$ are obtained from the output layer of CM. CO sends the classified data $A_1, A_2, \cdots, A_q$ to DA through CS. The classification accuracy (CA) of CM is calculated using Eq. (9), where $A_1, A_2, \cdots, A_{q''}$ indicates the number of correctly identified samples, $A_1, A_2, \cdots, A_q$ is the total number of samples in test data, and $q'' \leq q$.

$$\text{CA} = \frac{A_1, A_2, \ldots, A_{q''}}{A_1, A_2, \ldots, A_q} \tag{9}$$

The precision (P), and recall (R) are computed using Eq. (10) and (11), respectively.

$$P = \frac{(A_1, A_2, \ldots, A_{q''}) \cap (A_1, A_2, \ldots, A_q)}{A_1, A_2, \ldots, A_{q''}} \tag{10}$$

$$R = \frac{(A_1, A_2, \ldots, A_{q''}) \cap (A_1, A_2, \ldots, A_q)}{A_1, A_2, \ldots, A_q} \tag{11}$$

The F1-Score (FS) is measured using Eq. (12).

$$FS = 2 \times \frac{(P \times R)}{(P + R)} \tag{12}$$

6 Illustration

Consider DA has data (3.6216, 8.6661, – 2.8073, – 0.44699, 0), (4.5459, 8.1674, – 2.4586, – 1.4621, 0), ⋯, (3.9362, 10.1622, – 3.9435, – 3.5545, 0) of 20 DUs, respectively, and each DU demands classification services from CO. DA finds the location having minimum value of each DU's information using Eq. (1), which is – 2.8073, – 2.4586, ⋯, – 3.9435, respectively, for every DU. Afterwards, DA creates noise N using Eq. (5), which is based on Eqs. (2)–(4) and produces 20 noise values $N_1, N_2, \cdots, N_{20}$ equals 0.5886, 0.7830, ⋯, 0.1200, respectively. N is injected into DU's data to make it synthetic data D^N, as shown in Table 3. D^N is sent to CO through CS for preprocessing, which is performed using Eq. (6) and $\hat{D}^N$ is obtained as (0.8218, 0.8523, – 0.5940, 0.6034), (– 1.3304, – 1.5281, 2.2923, 0.2935), ⋯, (– 0.6391, 1.8055, – 0.1883, – 3.050). CM accepts 20 input data where ReLU activation function is applied at the hidden layers using Eq. (7) and sigmoid function is applied at the output layer using Eq. (8). CM produces the 20 labels as an output from the output layer. The model's CA is measured using Eq. (9). If the model correctly identifies 15 labels then the model's CA is 75% ((15/20)*100), where 20 is the total number of test items. The P of CM is measured by applying Eq. (10). If $A_1, A_2, \cdots, A_{q''}$ is 10, $A_1, A_2, \cdots, A_q$ is 20, and common labels among them is 8 then P of CM's is 80% ((8/10) * 100). Furthermore, R of CM is computed using Eq. (11). The CM's R is 40% ((8/20) * 100). The FS of CM is estimated by using Eq. (12), which is 54%.

7 Operational design and computational complexity

The operational description of the proposed model is presented in Algorithm 1, which initializes the list of data (D) and noise (N). In this algorithm, step 3 provides the location having minimum elements in D. N is injected into D and obtained the synthetic data (D^N) by applying step 4. In this step, N_1 is injected into D_1, N_2 is injected into $D_2, \cdots, N_n$ is injected into D_n. Step 7 depicts the classification parameters of DD-PPM. The steps for data classification have been described by Algorithm 2. In step 1, preprocessed data ($\hat{D}^N$), weight w, and bias b are initialized. In steps 4–5, the neural network classifies data. Step 8 provides the class labels.

Algorithm 1: DD-PPM operational summary

Input: Actual data D, Noise Vector N, function g
Output: CA, P, R, and FS
1 Initialize data $D := \{D_1, D_2, \ldots, D_n\}$, $N := \{N_1, N_2, \ldots, N_n\}$
2 **for** $i = 1, 2, \ldots, n$ **do**
3 $loc_i = g(D_i)$
4 $D_i^{N_i} = D_{loc_i} + N_i$
5 **Data_Classification** $(D_i^{N_i})$
6 **end for**
7 CA, P, R, and FS are calculated

Algorithm 2: Data Classification

Input: Input vector $\hat{D}^N$, weight w, bias b, activation function $f(x)$
Output: unknown class label A
1 Initialize parameter $\{w^1, w^2, w^3, b^1, b^2, b^3\}$
2 **for** $j = 1, 2, 3$ **do**
3 **for** $k = 1, \ldots, n_{j+1}$ **do**
4 $z_k^{(j+1)} = \sum_{t=1}^{n_j} \hat{D}_t^{N(j)} \cdot w_{kt}^{(j)} + b_k^{(j)}$
5 $A = f(z_k^{(j+1)})$
6 **end for**
7 **end for**
8 return A

The overall complexity of the operation flow of the proposed model is computed by steps involved in algorithms 1 and 2. In algorithm 1, steps 2–6 perform the classification over synthetic data, whose time complexity depends on the noise generation, noise addition, and use of a neural network. To find the location having minimum element takes the time $\mathcal{O}(\eta)$, and space $\mathcal{O}(\eta)$ in step 3, where η be the total number of input records. The synthetic data is produced in step 4 using the Laplace mechanism, which takes time $\mathcal{O}(\eta^2)$, and space $\mathcal{O}(\eta)$. The deep neural network model is performed using algorithm 2, which requires $\mathcal{O}(\eta^3)$ time, and $\mathcal{O}(\eta^2)$ space. Therefore, the total time, and space complexity of DD-PPM is $\mathcal{O}(\eta^3) = (\mathcal{O}(\eta) + \mathcal{O}(\eta^2) + \mathcal{O}(\eta^3))$, and $\mathcal{O}(\eta^2) = (\mathcal{O}(\eta) + \mathcal{O}(\eta) + \mathcal{O}(\eta^2))$, respectively. DD-PPM complexity analysis implies that endurable time and space aid protects the data, establishing its potency.

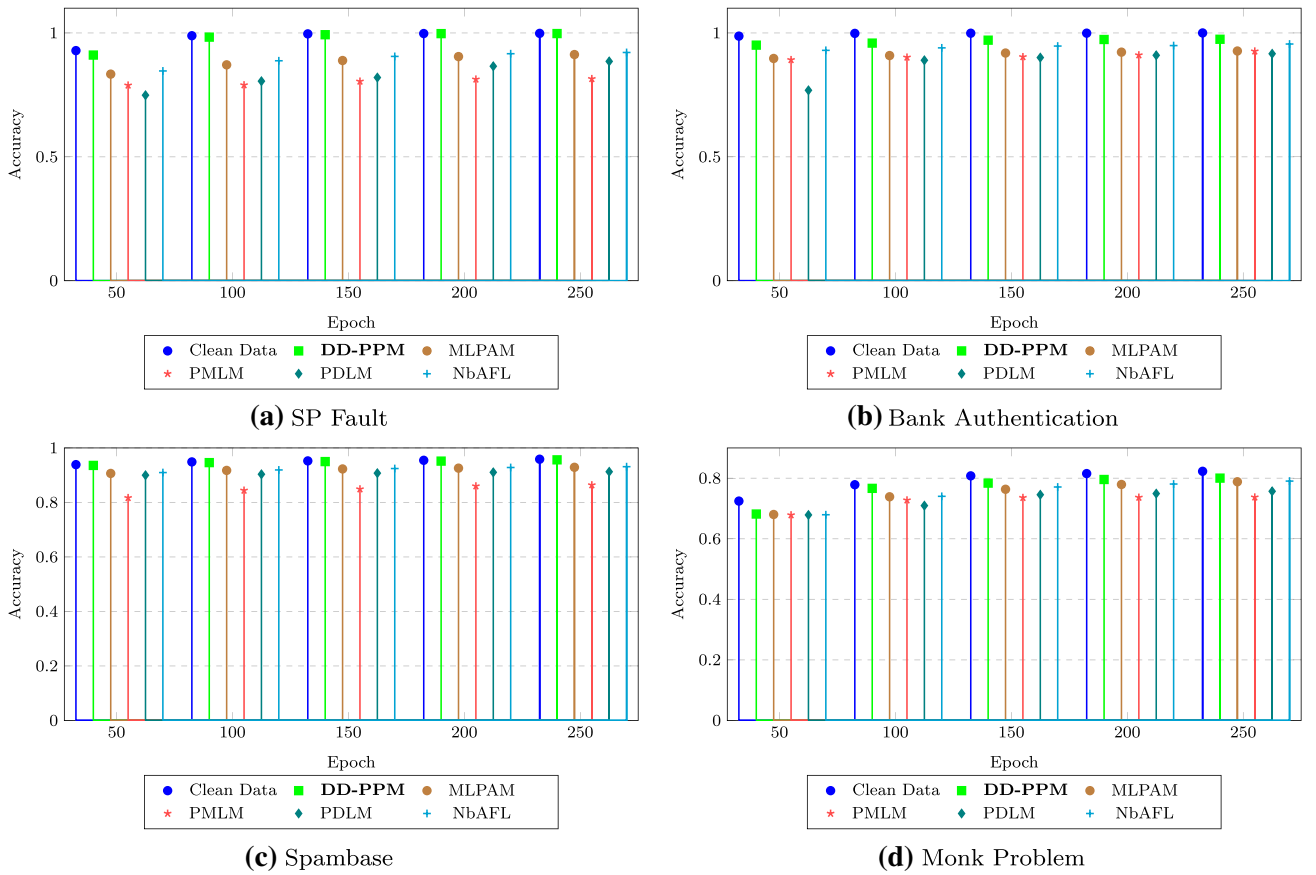
8 Performance evaluation

8.1 Experimental setup

The experiments are carried out on a server system equipped with two Intel[registered] Xeon[registered] Silver 4114 CPU with a 40 core processor and 2.20 GHz clock speed.

Table 4 Basic information of used datasets

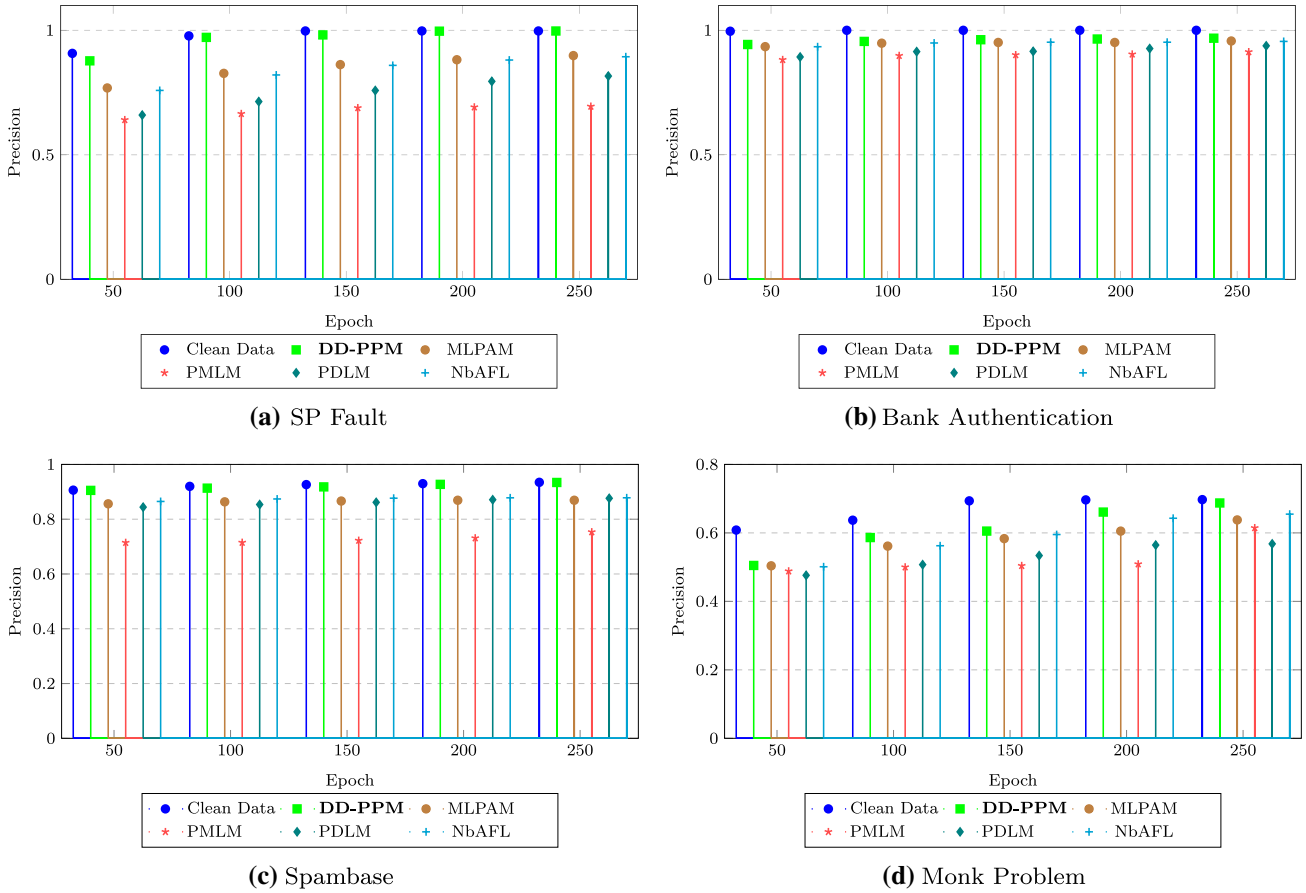
Dataset	#Instances	#Features	#Classes	Training set samples	Testing set samples
SP fault	1941	34	2	1552	389
Spambase	4601	58	2	3680	921
Banknote authentication	1372	5	2	1098	274
Monk problem	556	7	2	444	112



(a) SP Fault **(b)** Bank Authentication

(c) Spambase **(d)** Monk Problem

Fig. 3 Accuracy of classification model; DD-PPM vs state-of-the-art works



(a) SP Fault **(b)** Bank Authentication

(c) Spambase **(d)** Monk Problem

Fig. 4 Precision of classification model; DD-PPM vs state-of-the-art works

The computation machine is run on 64-bit Ubuntu with 8 GB of main memory. To perform the classification tasks, Python 3.7 has been used. In addition, the proposed work's performance is assessed using DNN with 50, 100, 150, 200, and 250 epochs over four separate datasets: Steel plates (SP) Fault, Spambase, Banknote Authentication, and Monk Problem, all of which have been taken from the UCI Machine Learning Repository (Frank 2010) for training CM. These datasets have 34, 58, 5, 7 attributes, and 1941, 4601, 1372, and 556 instances. The description of these datasets is shown in Table 4.

8.2 Classification parameters evaluation

The 80% of the total datasets are used for training, while the rest are for testing. For a particular case, there are 1941 instances in the SP Fault dataset. The 1552 instances (i.e., 8/10 of 1941 instances) are used as training samples, and the remaining 389 instances for testing samples. The training is carried out on clean as well as synthetic data. The Laplace mechanism is applied with a privacy level's value 0.1 to make synthetic data. The classification results including CA, *P*, *R*, and FS are achieved by CM on Clean data, DD-PPM, MLPAM (Gupta et al. 2020), PMLM (Li et al. 2018b), PDLM (Ma et al. 2018), and NbAFL (Wei et al. 2020). *CA* of the entire dataset is calculated, as shown in Fig. 3a–d. On the SP Fault dataset at 250 epochs, the highest value of CA is 99.75%, and the minimum value of CA is 68.17% over 50 epochs for the Monk Problem dataset. The average CA is 97.63%, 96.37%, 94.84%, and 76.56% over the SP Fault, Banknote Authentication, Spambase, and Monk Problem datasets, respectively. *P* for all datasets is demonstrated in Fig. 4a–d. The maximum value of *P* is 99.72% on the SP Fault dataset at 250 epochs, and the smallest value of *P* is 50.49% over 50 epochs for the Monk Problem dataset. The average *P* is 96.48%, 95.87%, 91.97%, and 60.89% over the SP Fault, Banknote Authentication, Spambase, and Monk Problem datasets, respectively. As shown in Fig. 5a–d, the value of *R* over the entire dataset is evaluated. On the SP Fault dataset at 250 epochs, the highest value of *R* is 99.72%, and the lowest value of *R* is 55.53% over 50 epochs for the Monk Problem dataset. The average *R* is 95.72%, 95.63%, 92.06%, and 62.58% over the SP Fault, Banknote Authentication, Spambase, and Monk

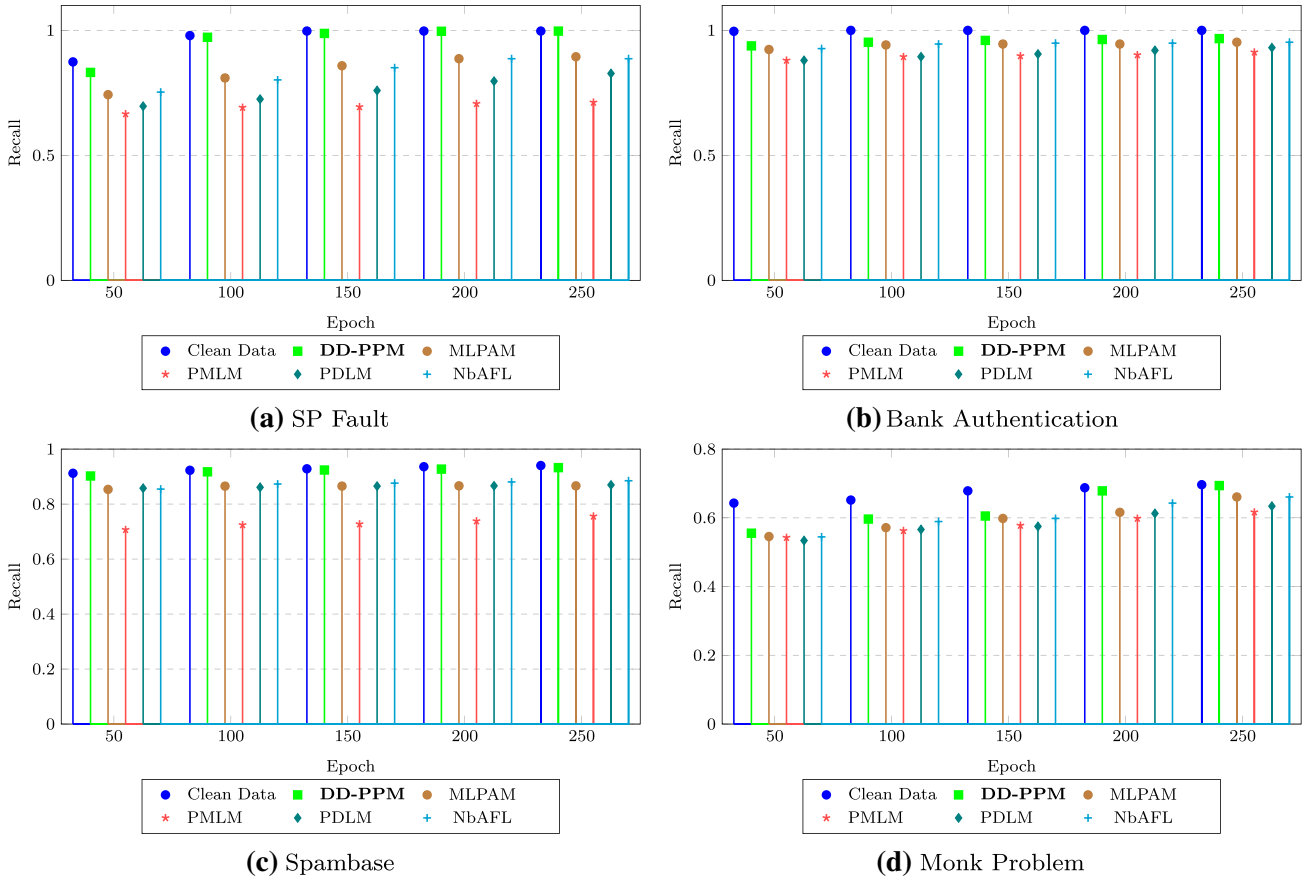


Fig. 5 Recall of classification model; DD-PPM vs state-of-the-art works

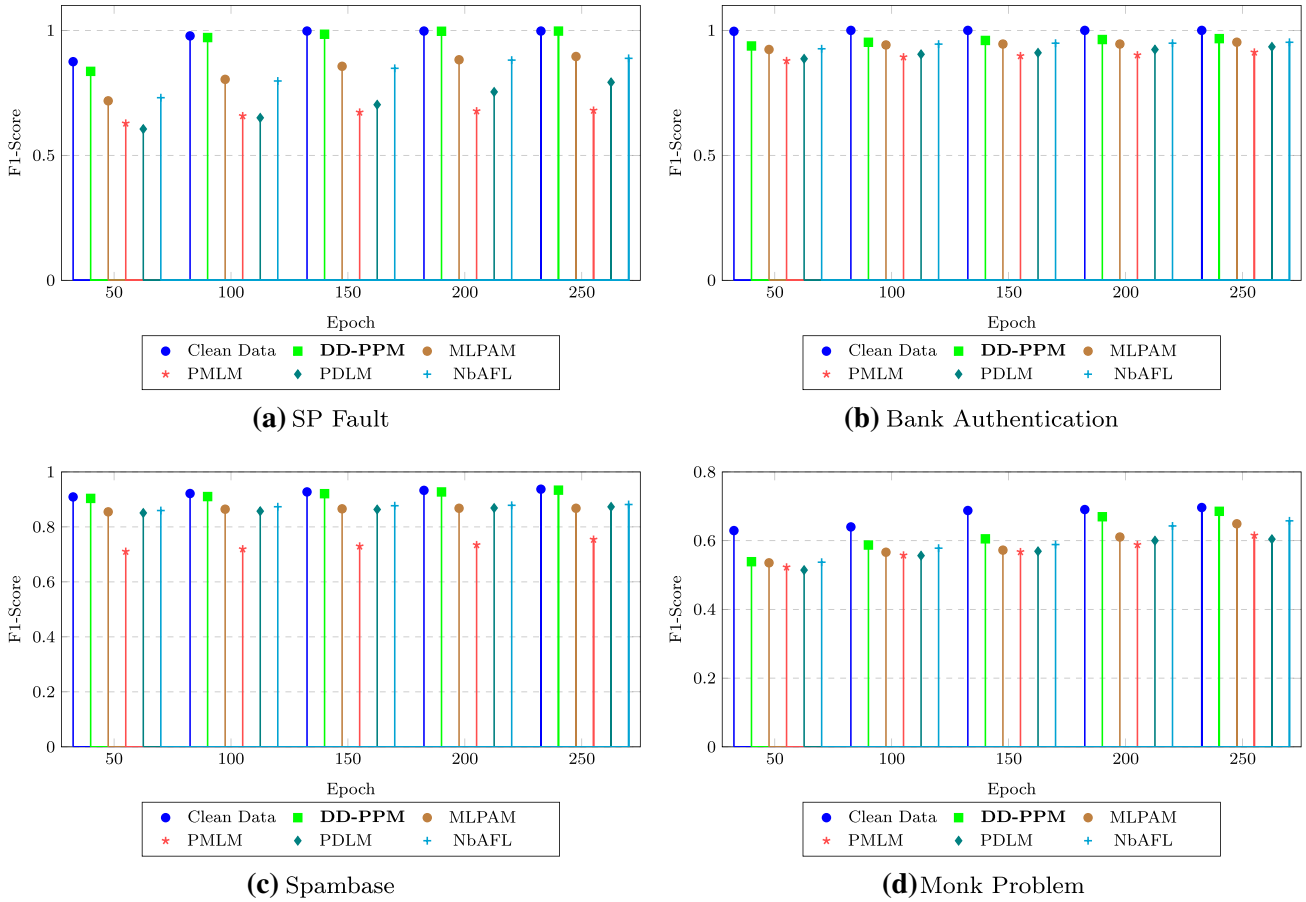
Problem datasets, respectively. FS of the complete dataset is assessed, as shown in Fig. 6a–d). The maximum value of FS is 99.72% obtained on the SP Fault dataset at 250 epochs, and the minimum value of FS is 53.88% over 50 epochs for the Monk Problem dataset. The average FS is 95.71%, 95.61%, 91.91%, and 61.72% over the SP Fault, Banknote Authentication, Spambase, and Monk Problem datasets, respectively. The datasets' performance descends in order: SP Fault, Banknote Authentication, Spambase, and Monk Problem. Overall, the value of classification parameters over the SP Fault dataset provides better results than other datasets because of the dataset's environments (technically, dimensions, and domain space).

9 Comparative analysis

9.1 Classification parameters

The experimental results are compared with MLPAM (Gupta et al. 2020), PMLM (Li et al. 2018b), PDLM (Ma et al. 2018), and NbAFL (Wei et al. 2020) via implementing these on the same platform. From Table 5, it is observed that the maximum gaps for CA, *P*, and FS of DD-PPM, MLPAM, PMLM, PDLM, and NbAFL are 19.34%, 30.67%, and 32.11%, respectively, on the SP Fault dataset at 100 epochs, and the minimum gaps are found 0.15%, 0.10%, and 0.15%, respectively, on the Monk Problem dataset at 50 epochs. Similarly, the highest difference for *R* is 29.39% on the SP Fault dataset at 150 epochs. In contrast, the minimum differences are found 0.71% on the Monk Problem dataset at 150 epochs. DD-PPM outperforms MLPAM, PMLM, PDLM, and NbAFL in all cases because of injecting the noise at a single and appropriate location rather than applying noise to the entire data set and reducing accuracy loss.

Moreover, the resultant values of CA, *P*, *R*, FS parameters of DD-PPM are lesser than clean data in all the cases because of the usage of synthetic data. It can be seen from Table 6 that the lowest gap for CA is 0.03% on the SP Fault dataset at 200 epochs, but the maximum gap is found 4.28% on the Monk Problem dataset at 50 epochs. Similarly, the lowest difference for *P* is 0.02% on the SP Fault dataset at 250 epochs, while the highest difference is found 10.34% on



(a) SP Fault **(b)** Bank Authentication

(c) Spambase **(d)** Monk Problem

Fig. 6 F1-Score of classification model; DD-PPM vs state-of-the-art works

the Monk Problem dataset at 50 epochs. The smallest decrement for R of DD-PPM is 0.02% from clean data on the SP Fault dataset at 250 epochs, whereas the maximum decrement is 8.75% on the Monk Problem dataset at 50 epochs. The lowest difference for FS is 0.02% on the SP Fault dataset at 250 epochs, but the highest difference is 9.05% on the Monk Problem dataset at 50 epochs. However, due to noise addition, CA, P, R, and FS results for DD-PPM are less but almost equal and also have more protection than clean data, MLPAM, PMLM, PDLM, and NbAFL.

9.2 Features

Table 7 presents a comprehensive feature analysis and comparison of DD-PPM with state-of-the-art works (Gong et al. 2020; Li et al. 2018b; Gupta et al. 2020; Li et al. 2018d; Ma et al. 2018; Wei et al. 2020). DD-PPM is the only model that enables multiple data users by identifying all involved entities as untrustworthy, preventing multiple public-key (PK) pairing, reducing classification accuracy loss, and providing high parameter values for classification.

9.3 Statistical

The statistical analysis is used to compare the performance in terms of CA, P, R, and FS of DD-PPM with the existing models MLPAM, PMLM, PDLM, and NbAFL. In Wilcoxon signed-rank test, the null hypothesis is considered, which states that the acquired outcomes from different algorithms are statistically identical. The value of classification parameters is measured with a significance level (p value) of 0.05 on datasets. Table 8 illustrates the results of the test statistics. It is observed that the null hypothesis is rejected for CA, P, R, and FS as their p value is less than 0.05, which validates DD-PPM for the SP fault, Banknote Authentication, Spambase, and Monk Problem datasets. The achieved statistics results demonstrate the superiority of the proposed model.

Table 5 Improvement of CA, *P*, *R*, and FS of DD-PPM over MLPAM (Gupta et al. 2020), PMLM (Li et al. 2018b), PDLM (Ma et al. 2018), NbAFL (Wei et al. 2020)

Dataset	Epoch	Evaluation parameters															
		Accuracy				Precision				Recall				F1-Score			
		Gupta et al. (2020)	Li et al. (2018b)	Ma et al. (2018)	Wei et al. (2020)	Gupta et al. (2020)	Li et al. (2018b)	Ma et al. (2018)	Wei et al. (2020)	Gupta et al. (2020)	Li et al. (2018b)	Ma et al. (2018)	Wei et al. (2020)	Gupta et al. (2020)	Li et al. (2018b)	Ma et al. (2018)	Wei et al. (2020)
SP fault	50	7.69	12.20	16.20	6.41	10.89	23.70	21.78	11.88	8.90	16.61	13.53	7.87	11.79	20.78	23.07	10.54
	100	11.18	19.34	17.78	9.54	14.45	30.67	25.78	15.09	16.28	28.10	24.76	17.05	16.73	31.35	32.11	17.38
	150	10.42	18.83	17.28	8.76	11.93	29.31	22.32	12.23	12.93	29.39	22.84	13.71	12.83	31.23	28.18	13.63
	200	9.27	18.44	13.47	8.11	11.47	30.51	20.16	11.59	10.97	28.96	19.96	10.97	11.40	31.91	24.26	11.54
	250	8.47	18.26	11.23	7.63	9.87	30.29	18.10	10.34	10.26	28.52	16.95	11.04	10.17	31.73	20.50	10.90
Banknote authentication	50	5.38	5.93	18.23	2.10	0.86	6.13	5.01	0.89	1.45	5.81	5.81	1.09	1.41	5.92	5.13	1.12
	100	5.02	5.75	6.93	1.92	0.71	5.73	4.10	0.63	1.09	5.82	5.82	0.73	1.07	5.91	4.81	0.74
	150	5.19	6.74	7.02	2.37	1.10	6.10	4.68	1.00	1.46	6.19	5.46	1.10	1.44	6.17	4.94	1.10
	200	5.11	6.29	6.38	2.46	1.38	6.12	3.83	1.28	1.82	6.18	4.36	1.46	1.81	6.22	4.02	1.47
	250	4.74	4.83	5.84	1.92	1.12	5.51	3.06	1.29	1.45	5.45	3.63	1.45	1.44	5.46	3.28	1.46
Spam-base	50	2.94	11.88	3.59	2.62	4.90	19.07	6.12	4.04	4.88	19.54	4.45	4.77	4.89	19.31	5.29	4.41
	100	2.86	10.20	4.27	2.69	5.00	19.86	5.98	3.95	5.21	19.32	5.64	4.45	4.60	19.09	5.32	3.70
	150	2.70	10.10	4.25	2.55	5.17	19.58	5.64	4.13	5.86	19.65	5.86	4.77	5.51	19.11	5.75	4.38
	200	2.58	9.17	4.11	2.34	5.80	19.61	5.61	4.91	6.08	18.89	6.08	4.67	5.94	19.25	5.84	4.86
	250	2.97	9.25	4.37	2.74	6.55	18.09	5.78	5.65	6.62	17.69	6.29	4.77	6.59	17.93	6.07	5.22
Monk problem	50	0.15	0.33	0.30	0.23	0.10	1.66	2.88	0.38	1.07	1.28	2.14	1.07	0.33	1.60	2.44	0.15
	100	2.78	3.90	5.70	2.63	2.50	8.64	7.91	2.39	2.50	3.39	3.04	0.72	2.09	2.96	3.08	0.91
	150	2.03	4.81	3.83	1.28	2.20	10.12	7.15	1.01	0.71	2.75	3.03	0.71	3.29	3.79	3.62	1.66
	200	1.65	5.93	4.66	1.50	5.56	15.19	9.63	1.80	6.25	8.03	6.57	3.57	5.90	8.15	6.96	2.68
	250	1.20	6.31	4.35	0.98	4.95	7.28	11.94	3.26	3.32	7.73	6.00	3.32	3.63	7.00	8.12	2.77

Table 6 Reduction of accuracy, precision, recall, and F1-Score of DD-PPM over clean data

Dataset	Epoch	Decrements in the value of parameters			
		Accuracy	Precision	Recall	F1-Score
SP fault	50	1.80	3.00	4.21	3.87
	100	0.57	0.60	0.69	0.66
	150	0.34	1.59	0.95	1.28
	200	0.03	0.09	0.09	0.09
	250	0.08	0.02	0.02	0.02
Banknote authentication	50	3.64	5.33	5.82	5.86
	100	3.92	4.46	4.73	4.75
	150	2.81	3.78	4.00	4.02
	200	2.55	3.50	3.64	3.65
	250	2.55	3.17	3.28	3.29
Spambase	50	0.29	0.09	0.98	0.53
	100	0.26	0.65	0.55	1.10
	150	0.25	0.83	0.44	0.64
	200	0.30	0.26	0.87	0.57
	250	0.23	0.06	0.76	0.35
Monk problem	50	4.28	10.34	8.75	9.05
	100	1.20	5.07	5.53	5.28
	150	2.40	8.83	7.32	8.24
	200	1.95	3.56	0.90	2.09
	250	2.25	0.98	0.25	1.10

9.4 Security

As the data needs to be shared among untrustworthy entities such as DU, CS, and CO; therefore, there are many possible ways for an attacker to gain access to the data D itself, resulting in a loss of information. In DD-PPM, differential privacy is an effective way to prevent data loss and ensure data protection. This mechanism injects statistical noise into D and keeps it private from untrusted entities. DA shares D^N to CS for storing and sharing purposes. Whenever CO requests for D^N, CS sends it. Since CM is performed over D^N. In this way, unauthorized users cannot obtain users' actual data from CS or other parties. Furthermore, the security primitives are evaluated with the ProVerif's security analysis tool that represents cryptographic protocols with Horn clauses. If the query is satisfied and returns true, the attacker cannot assault and protect the security primitive; otherwise, ProVerif reproduces the execution route to detect the attack phases. Figure 7 shows a phrase 'RESULT not attacker', which indicates that all variables, including DU, CS, CO, D^N, produced queries, and events, are protected from security breaches. Based on the results of using the ProVerif tool, it can be concluded that DD-PPM provides a robust security environment.

Table 7 Features comparison between state-of-the-arts works (Gong et al. 2020; Li et al. 2018b; Gupta et al. 2020; Li et al. 2018d; Ma et al. 2018; Wei et al. 2020), and DD-PPM

Schemes	UE	Data users	Avoids PK pair	CA loss	Security analysis	Methods		Parameters			
						Classifier	Privacy	CA	P	R	FS
PEDL Gong et al. (2020)	Multiple	Multiple	×	×	×	Deep learning	Differential privacy	90.34	×	×	×
PMLM Li et al. (2018b)	Multiple	Multiple	×	×	✓	Machine learning	Differential privacy	92.62	91.32	91.27	91.25
MLPAM Gupta et al. (2020)	Multiple	Multiple	×	×	✓	Machine learning	Differential privacy	92.89	95.71	95.27	95.27
POCC Li et al. (2018d)	Multiple	Multiple	×	×	✓	Naive bayes	Cryptography	×	×	×	×
PDLM Ma et al. (2018)	Single	Multiple	×	×	×	Deep learning	Cryptography	91.61	93.77	93.09	93.43
NbAFL Wei et al. (2020)	Single	Multiple	✓	×	×	Federated learning	Differential privacy	95.53	95.54	95.27	95.25
DD-PPM	Multiple	Multiple	✓	✓	✓	Deep neural	Differential privacy	99.75	99.72	99.72	99.72

Table 8 Wilcoxon test statistics (p value is 0.05)

Dataset	Classification parameters	Comparison of DD-PPM & Gupta et al. (2020)		Comparison of DD-PPM & Li et al. (2018b)		Comparison of DD-PPM & Ma et al. (2018)		Comparison of DD-PPM & Wei et al. (2020)	
		p value	Result	p value	Result	p value	Result	p value	Result
SP fault	Accuracy	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Precision	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Recall	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	F1-score	0.043	RE	0.043	RE	0.043	RE	0.043	RE
Banknote authentication	Accuracy	0.043	RE	0.043	RE	0.043	RE	0.042	RE
	Precision	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Recall	0.042	RE	0.043	RE	0.043	RE	0.043	RE
	F1-score	0.042	RE	0.043	RE	0.043	RE	0.043	RE
Spambase	Accuracy	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Precision	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Recall	0.043	RE	0.043	RE	0.043	RE	0.042	RE
	F1-score	0.043	RE	0.043	RE	0.043	RE	0.043	RE
Monk problem	Accuracy	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Precision	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	Recall	0.043	RE	0.043	RE	0.043	RE	0.043	RE
	F1-score	0.043	RE	0.043	RE	0.043	RE	0.043	RE

AC: Accept Null hypothesis, RE: Reject Null hypothesis

$--$ Query not attacker($DU[]$)	RESULT inj-event(endDU)
RESULT not attacker($DU[]$) is true.	$==>$ inj-event(startDU) is true.
$--$ Query not attacker($CS[]$)	$--$ Query inj-event(endCS)
RESULT not attacker($CS[]$) is true.	$==>$ inj-event(startCS)
$--$ Query not attacker($CO[]$)	RESULT inj-event(endCS)
RESULT not attacker($CO[]$) is true.	$==>$ inj-event(startCS) is true.
$--$ Query not attacker($D^N[]$)	$--$ Query inj-event(endCO)
RESULT not attacker($D^N[]$) is true.	$==>$ inj-event(startCO)
$--$ Query inj-event(endDU)	RESULT inj-event(endCO)
$==>$ inj-event(startDU)	$==>$ inj-event(startCO) is true.

Fig. 7 Security validation

10 Conclusion

This paper proposed a novel model named DD-PPM that optimizes the differential privacy and performs privacy-preserving learning over multiple data users' combined outsourced data in a cloud environment. It enabled various users to outsource their data to the cloud for storage, computation, and different statistical noise is injected into the data. The performance evaluation, feature, statistical, and security analysis illustrated that DD-PPM ensures high accuracy, precision, recall, F1-score and is secure, efficient, optimal than existing works. This model can be further extended by sharing the collected data in multiple environments and developing a more efficient data protection mechanism for various users.

Acknowledgements This work is supported by University Grant Commission, New Delhi, India, under the scheme of National Eligibility Test-Junior Research Fellowship (NET-JRF) with reference id-3515/(NET-NOV 2017).

Declarations

Conflict of interest The authors have no conflict of interest regarding the publication.

References

Ali M, Dhamotharan R, Khan E, Khan SU, Vasilakos AV, Li K, Zomaya AY (2015) Sedasc: secure data sharing in clouds. IEEE Syst J 11(2):395–404

Anand A, Singh A (2022) A hybrid optimization-based medical data hiding scheme for industrial internet of things security. IEEE Trans on Indus Inform

Cisco. Cisco Secure. https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-cybersecurity-series-2021-cps.pdf. [Online; accessed 2021]

Frank A (2010) Uci machine learning repository. http://archive.ics.uci.edu/ml

Fu Z, Xia L, Sun X, Liu AX, Xie G (2018) Semantic-aware searching over encrypted data for cloud computing. IEEE Trans Inform Foren Secur 13(9):2359–2371

Gao C-z, Cheng Q, He P, Susilo W, Li J (2018) Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack. Inf Sci 444:72–88

Gong M, Feng J, Xie Y (2020) Privacy-enhanced multi-party deep learning. Neural Netw 121:484–496

Gupta I, Gupta R, Singh AK, Buyya R (2020) Mlpam: a machine learning and probabilistic analysis based model for preserving security and privacy in cloud environment. IEEE Syst J 15:4248–4259

Hassan A, Hamza R, Yan H, Li P (2019) An efficient outsourced privacy preserving machine learning scheme with public verifiability. IEEE Access 7:146322–146330

Hauer B (2015) Data and information leakage prevention within the scope of information security. IEEE Access 3:2554–2565

Hesamifard E, Takabi H, Ghasemi M, Wright RN (2018) Privacy-preserving machine learning as a service. Proc Privacy Enhanc Tech 2018(3):123–142

Li P, Li J, Huang Z, Li T, Gao C-Z, Yiu S-M, Chen K (2017) Multi-key privacy-preserving deep learning in cloud computing. Future Gen Comp Syst 74:76–85

Li J, Chen X, Chow SSM, Huang Q, Wong DS, Liu Z (2018) Multi-authority fine-grained access control with accountability and its application in cloud. J Netw Comp Appl 112:89–96

Li P, Li T, Ye H, Li J, Chen X, Xiang Yang (2018) Privacy-preserving machine learning with multiple data providers. Future Gen Comp Syst 87:341–350

Li T, Huang Z, Li P, Liu Z, Jia C (2018) Outsourced privacy-preserving classification service over encrypted data. J Netw Comp Appl 106:100–110

Li P, Li J, Huang Z, Gao C-Z, Chen W-B, Chen K (2018) Privacy-preserving outsourced classification in cloud computing. Cluster Comput 21(1):277–286

Li T, Li J, Liu Z, Li P, Jia C (2018) Differentially private Naive Bayes learning over multiple data sources. Inf Sci 444:89–104

Li H, Cui J, Meng X, Ma J (2019) Ihp: improving the utility in differential private histogram publication. Distrib Parallel Databases 37(4):721–750

Ma X, Ma J, Li H, Jiang Q, Gao S (2018) Pdlm: Privacy-preserving deep learning model on cloud with multiple keys. IEEE Trans Serv Comput 14:1251–1263

Manimuthu A, Murugaboopathi G (2021) An enhanced approach on distributed accountability for shared data in cloud. J Ambient Intell Human Comput 12(5):5421–5425

Phuong TT et al (2019) Privacy-preserving deep learning via weight transmission. IEEE Trans Inf Foren Secur 14(11):3003–3015

Sandhia GK, Raja SVK (2021) Secure sharing of data in cloud using MA-CPABE with elliptic curve cryptography. J Ambient Intell Humaniz Comput 13:1–10

Saxena D, Singh AK (2020) Security embedded dynamic resource allocation model for cloud data centre. Electron Lett 56(20):1062–1065

Shen W, Qin J, Jia Y, Hao R, Jiankun Hu (2018) Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. IEEE Trans Inform Foren Secur 14(2):331–346

Singh AK, Kumar J (2019) Secure and energy aware load balancing framework for cloud data centre networks. Electron Lett 55(9):540–541

Singh KN, Singh AK (2022) Towards integrating image encryption with compression: a survey. ACM Trans Multimed Comput Commun Appl (TOMM) 18(3):1–21

Wang C, Wang A, Xu J, Wang Q, Zhou F (2020) Outsourced privacy-preserving decision tree classification service over encrypted data. J Inform Secur Appl 53:102517

Wei J, Liu W, Hu X (2016) Secure data sharing in cloud computing using revocable-storage identity-based encryption. IEEE Trans Cloud Comp 6(4):1136–1148

Wei K, Li J, Ding M, Ma C, Yang HH, Farokhi F, Jin S, Quek TQS, Poor HV (2020) Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans Inf Foren Secur 15:3454–3469

Yonetani R, Boddeti Naresh V, Kitani KM, Sato Y (2017) Privacy-preserving visual learning using doubly permuted homomorphic encryption. In: Proc. of the IEEE Int. Conf. on Comput. Vis., pp 2040–2050

Zaghloul E, Zhou K, Ren J (2019) P-mod: secure privilege-based multilevel organizational data-sharing in cloud computing. IEEE Trans Big Data 6(4):804–815

Zhang T, Zhu Q (2016) Dynamic differential privacy for ADMM-based distributed classification learning. IEEE Trans Inf Foren Secur 12(1):172–187