Computational Economics (2025) 66:2619–2643
https://doi.org/10.1007/s10614-024-10731-0

PPDNN-CRP: CKKS-FHE Enabled Privacy-Preserving Deep Neural Network Processing for Credit Risk Prediction

Vankamamidi S. Naresh[1] · D. Ayyappa[1]

Accepted: 20 September 2024 / Published online: 3 December 2024

Abstract

Credit risk prediction is essential in modern financial analytics, yet it poses significant privacy challenges due to the sensitive nature of the data involved. This study presents the PPDNN-CRP framework, which integrates Deep Neural Networks (DNNs) with the Cheon-Kim-Kim-Song (CKKS) Homomorphic Encryption (HE) to achieve privacy-preserving credit risk prediction. The framework ensures privacy throughout the entire prediction process, including both the training and inference phases, by protecting training data, input data, the model, and output data.We evaluate the performance of the PPDNN-CRP framework using real-world datasets from Kaggle, comparing it against DNN-CRP (on unencrypted data) and PPLR. The results indicate that PPDNN-CRP outperforms the other models across most performance metrics, making it the most suitable choice for applications where privacy is a critical concern. Furthermore, the security analysis confirms that the PPDNN-CRP framework effectively mitigates threats such as data poisoning, evasion attacks, membership inference, model inversion, and model extraction throughout the machine learning lifecycle.

Keywords Credit risk prediction · Cheon-Kim-Kim-Song (CKKS) schem · Deep neural network · Homomorphic encryption · Privacy preserving

1 Introduction

In the current era of data-centric operations, safeguarding privacy is crucial, particularly in the financial industry where confidential information is essential for processes like assessing credit risk. Banks play a critical role in economic growth, primarily through lending systems that depend on accurately evaluating borrowers' creditworthiness. This evaluation necessitates the use of sensitive personal and financial data, making the protection of information security and privacy essential.

✉ Vankamamidi S. Naresh
vsnaresh111@gmail.com

[1] Department of CSE, Sri Vasavi Engineering College, Tadepalligudem, Andhra Pradesh, India

The emergence of Deep Neural Networks (DNNs) has transformed credit risk assessment, providing exceptional accuracy through their capacity to analyze and model intricate patterns in extensive datasets. These sophisticated models can identify complex relationships within the data, making them highly effective at predicting potential loan defaults. However, the implementation of DNNs in finance raises significant privacy issues. Many financial institutions are reluctant to fully embrace these models due to the potential risks of exposing confidential information during data processing. This challenge highlights the necessity for advanced techniques that can maintain data privacy while leveraging the predictive capabilities of DNNs.. Homomorphic Encryption (HE) presents itself as a promising approach to address the challenge of data security, enabling computations on encrypted information without the need for decryption. This technology ensures data remains protected and confidential even during processing by sophisticated models.

The Cheon-Kim-Kim-Song (CKKS) scheme is particularly noteworthy among various HE methods for its efficient handling of encrypted real numbers, which are prevalent in financial datasets. In contrast to earlier encryption techniques like the Paillier Homomorphic Encryption (PHE) Cryptosystem, which supported only limited homomorphic operations, Fully Homomorphic Encryption (FHE), introduced by Gentry in 2009, and specifically the CKKS scheme, allows for both addition and multiplication operations on ciphertexts. The CKKS scheme is especially suitable for privacy-preserving credit risk assessment, as it allows financial institutions to apply complex machine learning models, such as Deep Neural Networks (DNNs), directly to encrypted data. This capability enables accurate creditworthiness evaluation without exposing sensitive information. By combining DNNs with CKKS, banks can uphold high standards of data privacy and security while leveraging the predictive power of advanced machine learning models. This integration not only bolsters the security and privacy of loan processing systems but also empowers financial institutions to make more informed and reliable lending decisions, ultimately contributing to the stability and growth of the financial sector.

This study investigates the application of the CKKS scheme in conjunction with DNNs for privacy-preserving credit risk assessment. Our objective is to show that this method can deliver strong security assurances without sacrificing the precision of credit risk assessments. This approach offers a promising solution for financial organizations looking to harness advanced data analysis techniques while protecting confidential information.

1.1 Contributions

The main contribution of this paper is to build a CKKS-FHE-enabled PPDNN-CRP system with the following features:

- Proposed an CKKS-FHE-enabled DNN Processing Framework that can provide privacy in training and inference phase with training data privacy, model privacy, input privacy, and output privacy.

- We made a security analysis that shows that the proposed system can defend against poison, evasion, member inference, model inversion, and model extraction in the respective stages of machine learning.
- We conduct experiments using the Tenseal package on real datasets from the Kaggle to evaluate the performance of both DNN-CRP (over unencrypted data) and the proposed PPDNN-CRP.

2 Related Work

This study builds on previous work in credit risk prediction and privacy-preserving techniques, tackling the issue of protecting data confidentiality while maintaining accurate predictions in financial analysis. Earlier investigations primarily utilized machine learning models, such as those created by Gupta et al. (2020), Rath et al. (2021), and Anand et al. (2022), which concentrated on predicting credit risk using factors like credit history and income. Although these methods were effective, they typically relied on unencrypted data, exposing sensitive information to potential security risks.

With the rise of data protection concerns, researchers began integrating privacy-preserving methods into machine-learning models for subsequent studies. Investigations by Shoumo et al. (2019) and Ahamed et al. (2021) and Bhargav & Sashirekha (2023), explored the use of machine learning in credit risk assessment, implementing techniques such as random forests, which demonstrated some effectiveness in addressing privacy issues while maintaining accuracy. Additional studies, including those conducted by Dansana et al. (2023), Blessie et al. (2019), Zhu et al. (2019), and Alsaleem et al. (2020), investigated different machine learning techniques but encountered challenges in safeguarding sensitive information during data processing.

In response to the aforementioned challenge, a compilation of studies investigates advancements in privacy protection across diverse machine learning fields. These studies, including Wang et al. (2019), Uddin et al. (2023), Lu et al. (2022), Ma et al. (2023), Cristiano et al. (2022), Toubeau et al. (2022), Zhang et al. (2023), Ma, C et al. (2023), Lu et al. (2022), Cristiano et al. (2022), Zhang et al. (2022), and Xie et al. (2024), propose the implementation of various techniques such as differential privacy, homomorphic encryption, and federated learning. These methods aim to safeguard data while maintaining model efficacy and accuracy.

Recent studies have shifted focus towards integrating sophisticated privacy-enhancing techniques, including Differential Privacy (DP), Homomorphic Encryption (HE), and Federated Learning (FL), to address these concerns. A comprehensive analysis of HE's integration with neural networks was conducted by Pulido-Gaytan et al. (2022), highlighting its potential for secure predictions in privacy-critical applications. Furthermore, several researchers, including Al Hadhrami et al. (2023), Bragagnini et al. (2022), Cappello et al. (2022), Ghassemi et al.(2021), Scott et al. (2023), Aryal et al. (2023), Stephanie et al. (2022), Khalid et al. (2022), and Geambasu et al. (2021), investigated the combined potential of Homomorphic Encryption (HE), Multiparty Computation (MPC), and Differential Privacy (DP). Their studies demonstrated that these techniques can effectively protect privacy while maintaining model accuracy.

In current research, new frameworks have been proposed to improve privacy protection in machine learning systems. One such example is the work of Nguyen et al. (2023), who introduced an integrated approach that combines Federated Learning (FL) with Homomorphic Encryption (HE) using AI models and the CKKS algorithm. This method achieved high average accuracy across various datasets. Additionally, Su et al. (Zhou & Zhang, 2022) developed a secure data fitting technique for medical Internet of Things (IoT) applications, also based on the CKKS algorithm. Their approach demonstrated both high recognition accuracy and effective privacy safeguards.

Despite the advantages of Homomorphic Encryption (HE) in Deep Neural Networks (DNNs), researchers have identified certain constraints and compromises. Onoufriou et al. (2022) recognized that while Fully Homomorphic Encryption (FHE) excels in securing private predictions for neural networks, it is not a comprehensive answer to all privacy concerns in machine learning, particularly during model training. Jain et al. (2022) emphasized the computational depth restrictions and resource-intensive nature of FHE operations. To tackle scalability and privacy issues in DNNs, Zhou et al. (2022) developed a system that integrates both algorithmic and cryptographic methods, including homomorphic encryption. Addressing the problem of gradient leakage in Federated Learning (FL) systems, Pan et al. (2024) proposed an innovative adaptive segmented encryption technique to boost the CKKS scheme's capacity for encrypting neural network models.

This study builds upon existing advancements by proposing a PPDNN-CRP framework that combines Deep Neural Networks (DNNs) with CKKS Homomorphic Encryption (HE) to deliver a holistic privacy-preserving approach for credit risk prediction. In contrast to earlier models that often required trade-offs between privacy and effectiveness, the PPDNN-CRP framework guarantees privacy at every stage of the prediction process, from training to inference, while maintaining accuracy. Our experimental findings demonstrate that when evaluated on actual datasets, the PPDNN-CRP framework outperformed conventional models like PPLR and DNN-CRP across most performance indicators. Furthermore, an in-depth security evaluation verified the framework's resilience against various threats, including poisoning, evasion, and model inversion attacks. This positions PPDNN-CRP as a notable improvement in privacy-preserving credit risk evaluation, offering enhanced security and precision compared to previous methods.

3 Background Knowledge

This part gives a brief description of pallier and DNN for the proposed system.

3.1 Notations

Table 1 presents notations used in this paper.

Table 1 Notations

Symbol	Description
HE	Homomorphic Encryption
DNN	Deep Neural Network
DNN-CRP	Deep Neural Network Processing for Credit Risk Prediction
PPDNN-CRP	Privacy-Preserving Deep Neural Network Processing for Credit Risk Prediction
LR	Logistic Regressions
CSP	Cloud Service Provider
DP	Differential Privacy
PII	Personally Identifiable Information
MPC	Multi Party Computation
GANs	Generative Adversarial Networks
BDP	Boundary Differential Privacy
TP	Training Phase
TSP	Testing Phase
PUK	Public Key
PRK	Private Key
PPLR	Privacy-Preserving Logistic Regressions

3.2 Workflow of a Deep Neural Network (DNN)

The workflow of DNN is depected in Fig. 1 and DNN processing steps were presented as follows.

DNN Processing Steps

Step 1: Initialize and Secure Training Data

During training, encrypt the CRP Financial Dataset with homomorphic encryption to safeguard private data and stop illegal access

Step2: Input Data Privacy Enforcement

Feed the encrypted feature set X=(×1,×2,×3,…,xn) into the input layer while keeping it encrypted to ensure secure processing and maintain input data privacy within the DNN

Step3: Hidden Layer Computations with Model Weight Privacy

For each hidden layer, compute the linear transformation $Z_1 = W_1.X + b_1$ using encrypted weights and biases, apply ReLU activation $A_1 = ReLu(z_1)$, normalize outputs with batch normalization, and use dropout to deactivate neurons randomly, ensuring the model's learning process remains private by keeping weights encrypted

Step4: Output Layer and Data Privacy:

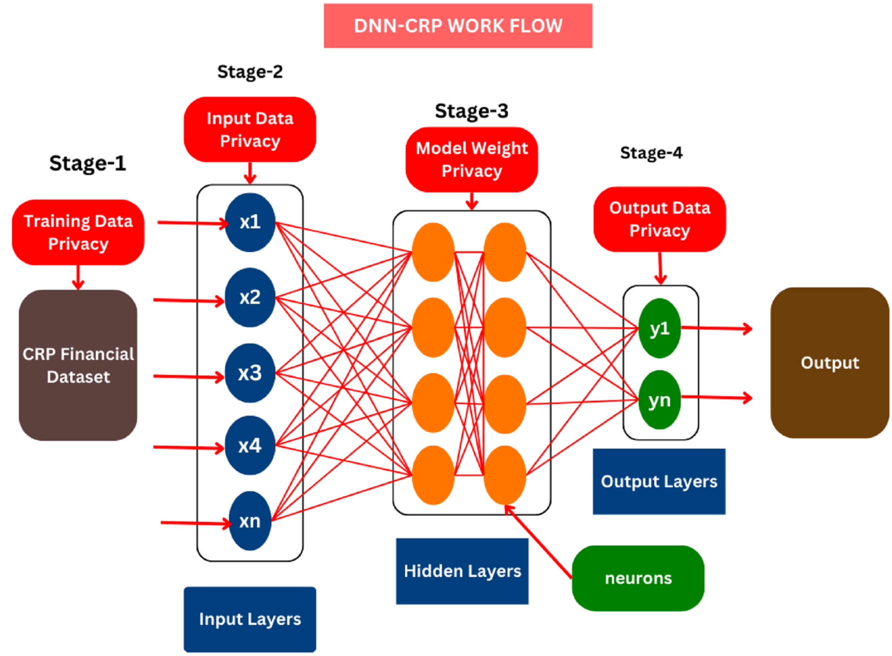


Fig. 1 Workflow of a Deep Neural Network (DNN)

DNN Processing Steps

The output layer calculates Z_output = W_(output).A_2 + b_output applies the sigmoid activation A_output = sigmoid(z_output) and encrypts the final output to secure model predictions and restrict access to authorized parties only

Where: • X is the input data

$w_1, b_1, w_2, b_2, w_{output}, b_{output}, aretheweightmatricesandbiasvectorsforeachlayer.$

$Z_1, A_1, Z_2, A_2, Z_{output}, A_{output}, arethepre - activation, post - activationvaluesforeachlayer.$

- ReLU is the rectified linear unit activation function
- Sigmoid is the sigmoid activation function

The proposed method aims to safeguard privacy across the entire deep neural network (DNN) process, encompassing the reception of encrypted data inputs and the production and management of encrypted results. These safeguards are essential in scenarios where data confidentiality is of utmost importance, such as in credit risk assessment applications. By implementing this approach, the DNN for credit risk prediction can maintain high levels of accuracy while complying with stringent data privacy requirement.

3.3 Cheon-Kim-Kim-Song (CKKS) Fully Homomorphic Encryption Cryptosystem

CKKS (Pan et al. 2024), a type of Fully Homomorphic Encryption (FHE), is an asymmetric homomorphic cryptographic system comprising key generation, encryption, and decryption components. Its homomorphic characteristic enables computations on encrypted information without the need for decryption, making it particularly valuable in privacy-preserving applications like the Proposed Credit Risk Prediction.

3.3.1 CKKS Algorithm

CKKS.Key Generation ***(λ)***

Step	Description
1. Choose Parameters	Select a security parameter λ to determine the size of the modulus and polynomial ring
2. Polynomial Modulus and Cyclotomic Polynomial	Choose a polynomial modulus degree N (a power of 2) and a ciphertext modulus q. Define the cyclotomic polynomial $\Phi_{N(x)}$.
3. Generate Secret Key	Randomly select a secret key polynomial s from a small, discrete distribution over $R_q = \frac{\mathbb{Z}_{q[x]}}{\Phi_{N(x)}}$.
4. Generate Public Key	Select a random polynomial a from $R_q.Computeb = -a * s + emodq$, where e is a small error polynomial. The public key is (a, b)
5. Evaluation Key Generation	Compute evaluation keys for homomorphic operations, enabling encrypted computations without needing the secret key

3.3.2 CKKS.Encryption*(m, pk)*

Step	Input	Output	Description
1. Encoding	Real or complex number m	Encoded polynomial m(x)	Encode m into polynomial m(x) in the polynomial ring
2. Encrypt	Encoded polynomial m(x), public key (a, b)	Ciphertext $\left(c_0, c_1\right)$	Select random u from R_q. Compute:$c_0 = b * u + e_1 + \Delta * mmodq$, $c_1 = a * u + e_2 modq.$
3. Ciphertext			Encrypted message represented as $\left(c_0, c_1\right)$.

3.3.3 CKKS.Decryption*(c, sk)*

Step	Input	Output	Description
1. Ciphertext	$\left(c_0, c_1\right)$.		Take the ciphertext $\left(c_0, c_1\right)$.
2. Decrypt	Secret key s, ciphertext $\left(c_0, c_1\right)$.	Polynomial v	$Computev = c_0 + c_1 * smod$ q
3. Decoding	Polynomial v	Approx. plaintext m	Decode v to retrieve the approximate original plaintext message m

3.3.4 Homomorphic Properties

Operation	Ciphertexts	Result	Description
Addition	$\left(c_0^1, c_1^1\right), (c_0^2, c_1^2))$	$\left(c_0^1 + c_0^2, c_1^1 + c_1^2\right) modq$	Homomorphic addition results in a ciphertext decrypting to $m_1 + m_2$
Multiplication	$\left(c_0^1, c_1^1\right), \left(c_0^2, c_1^2\right)$	New ciphertext (after relinearization)	Results in ciphertext decrypting to $m_1 * m_2$ after rescaling

3.3.5 CKKS Algorithm's Workflow for Privacy-Preserving DNN-CRP

Stage	Description	Key actions
Setup and Key Generation	Prepare the cryptographic keys for secure operations	Generate public, secret, and evaluation keys
Data Encryption	Secure the sensitive battery sensor data by converting it into ciphertext	Encode and encrypt raw data
Encrypted Data Transmission	Transmit the secured data to the server for further processing	Send ciphertext to server
Encrypted Computation	Perform required calculations on the encrypted data to predict SoH	Homomorphic addition, multiplication, scaling, ML model inference
Encrypted Result Transmission	Send the computed results back to the client while keeping them secure	Return encrypted result to client
Decryption and Interpretation	Convert the encrypted result back to a usable form and interpret the results	Decrypt and interpret SoH prediction

3.4 DNN-CRP Architecture and Rationale for Using a DNN for CRP

The DNN-CRP workflow is depicted as a flowchart in Fig. 2. The process starts with a financial dataset, which is then preprocessed and undergoes feature engineering to produce a refined dataset. This processed data is subsequently divided into training and testing subsets, which are utilized to construct and assess the DNN model's effectiveness.

3.4.1 Rationale for Using a DNN for CRP Prediction

DNNs are well-suited for SoH prediction due to their capacity to process complex, multidimensional sensor data, automatically identify relevant features, and model non-linear relationships. These capabilities are essential in SoH prediction, where accurate forecasts rely on detecting subtle patterns within large datasets. Additionally, DNNs offer scalability, allowing them to adapt to various levels of data complexity, and exhibit resilience to noisy data, ensuring consistent performance in less-than-ideal conditions. When compared to alternative models, DNNs frequently provide superior predictive accuracy, making them the optimal choice for this application.

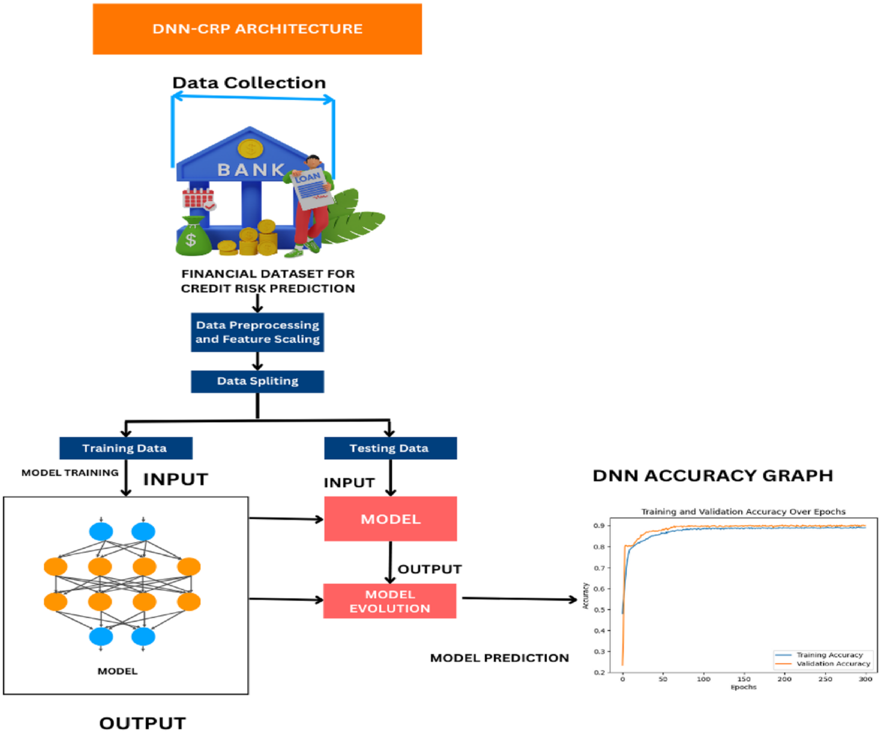


Fig. 2 Block Diagram For DNN-CRP

4 Proposed System

4.1 System Modelfor PPDNN-CRP

A system model for developing Deep Neural Network (DNN) and Logistic Regression models is depicted in Fig. 3. This model comprises three main components: a Bank, Customers submitting loan applications, and a Cloud Service Provider (CSP). The models are constructed using a dataset containing sensitive loan application information, with an emphasis on privacy-preserving techniques for secure data management. Key Entities: i. Customers (Loan Applicants):Individuals seeking loans who submit personal and financial information to the bank. This data includes income details, credit history, and other relevant factors necessary for loan approval decisions. ii. Bank:The financial institution that collects and manages customer-provided information during the loan application process. The bank possesses sensitive data, such as credit histories, assets, and income information. To ensure data confidentiality, the bank employs privacy-preserving methods, including encryption, to safeguard the dataset. iii. Cloud Service Provider (CSP):A third-party vendor offering cloud-based infrastructure for training and processing machine learning models. The CSP will utilize the encrypted dataset supplied by the bank to facilitate the training of both DNN and Logistic Regression models.

4.1.1 Data Flow

1. Information gathering: Customers provide personal and financial details to the bank during the loan application process, enabling the collection of essential financial information.

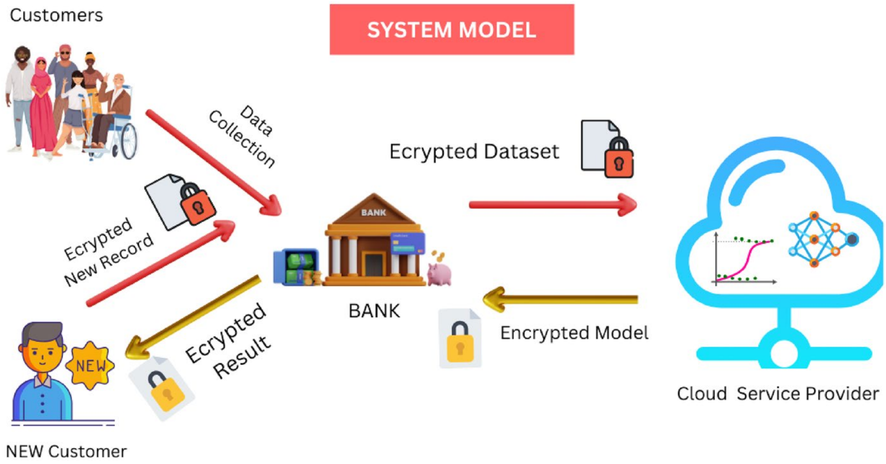


Fig. 3 System model for PPDNN-CRP

2. Information consolidation and de-identification: The bank mitigates privacy concerns by consolidating and de-identifying the gathered information, removing Personally Identifiable Information (PII).
3. Information security: The de-identified dataset undergoes encryption to safeguard its confidentiality and integrity, ensuring sensitive information remains protected during processing.
4. Secure transfer to CSP: The encrypted dataset is securely transmitted to the CSP, maintaining information integrity and confidentiality throughout the transfer.
5. Algorithm training (DNN and Logistic Regression): The CSP utilizes the encrypted dataset to train DNN and Logistic Regression algorithms. These algorithms are employed to maintain information privacy while generating predictive insights, such as assessing credit risk and enhancing loan approval decisions.
6. Algorithm implementation and application: Upon completion of training, the algorithms are returned to the bank's network for local application. This eliminates the need for continuous information transfer to the cloud, enhancing information security.

4.1.2 Privacy-Preserving Techniques

CKKS cryptosystems are employed to enable specific computations on encrypted data without decryption. This approach allows for the processing of confidential loan application information while maintaining privacy.

4.1.3 Security Measures

Data transmission is safeguarded through encryption to prevent unauthorized access or interception of sensitive loan application information. Both the Bank and CSP implement access controls and authentication protocols to ensure that only authorized individuals can view and modify the data. By incorporating these privacy-preserving methods and security protocols into the system design, the Bank can leverage insights from the DNN model while protecting loan applicants' privacy and adhering to ethical and legal requirements.

4.2 Privacy-Preserving Credit Risk Prediction (PPCRP) Framework

Privacy-Preserving Credit Risk Prediction (PPCRP) FrameworkThe methodology for the bank loan processing system using a DNN with CKKS homomorphic encryption comprises two primary stages, as illustrated in Fig. 4 and Fig. 5:

4.2.1 Training Phase

The training privacy phase, consisting of steps S1 through S6, initiates with the gathering and protected storage of confidential financial information. During S1 and S2, financial data is collected from clients by the bank and securely stored as a

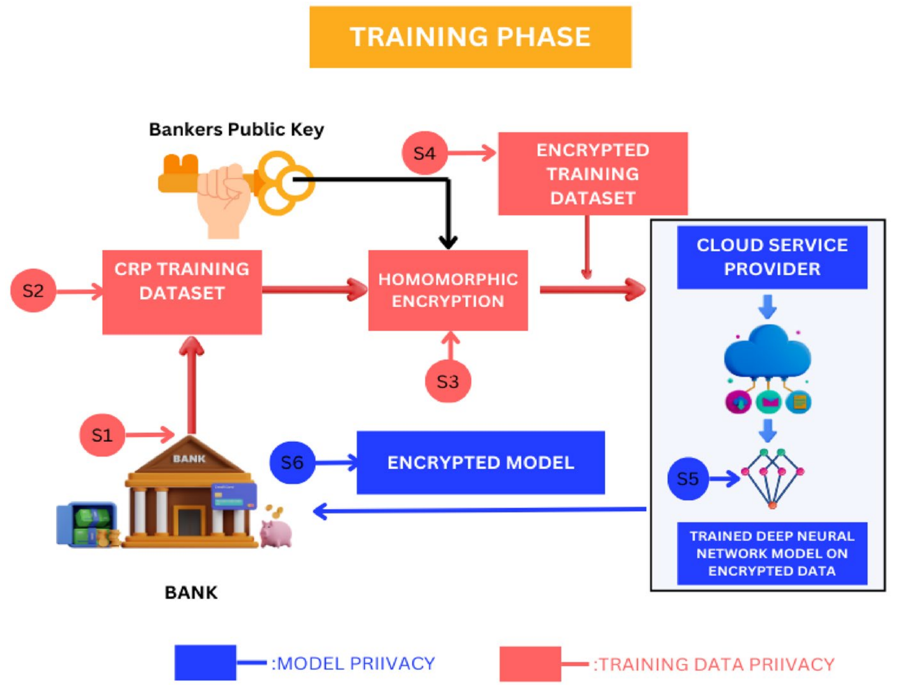


Fig. 4 Privacy-Preserving Credit Risk Prediction Framework (PPCRPF) For Training Phase

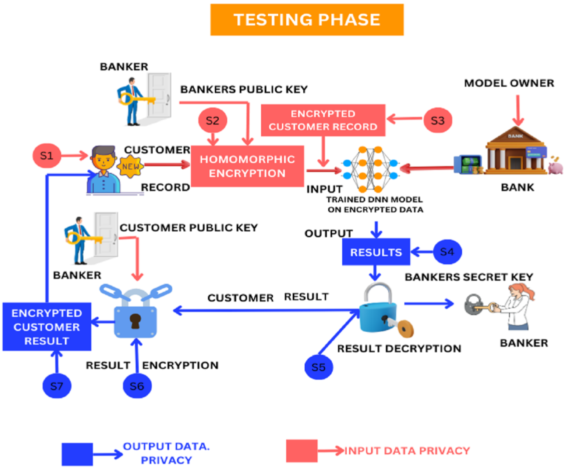


Fig. 5 Privacy-Preserving Credit Risk Prediction Framework (PPCRPF) For Testing Phase

Credit Risk Prediction (CRP) training dataset on its internal servers, preparing it for model training. In S3, the dataset undergoes encryption using the Banker's Public Key, utilizing the CKKS scheme of Homomorphic Encryption (HE) to safeguard data, resulting in an encrypted training dataset. S4 involves the secure transfer of this encrypted dataset to a Cloud Service Provider (CSP), establishing Training Data Privacy. During S5, the CSP utilizes the encrypted dataset to train a Deep Neural Network (DNN) model. Lastly, in S6, the encrypted model is securely sent back to the bank, ensuring Model Privacy.

4.2.2 Testing Phase

The evaluation stage, comprising steps S1 through S7, is carefully structured to protect data confidentiality during the credit risk assessment process. It commences with Step S1, in which a prospective client provides their financial information to the banking representative. Step S2 involves encrypting this data using the Banker's Public Key, employing the CKKS method of Homomorphic Encryption (HE), resulting in a securely encoded customer file. This encrypted information is subsequently utilized in Step S3, where it serves as input for the Deep Neural Network (DNN) model to perform secure predictions, thereby ensuring Input Data Privacy.

To safeguard Output Data Privacy, the DNN model produces encrypted prediction results in Step S4. The bank then uses its Secret Key to decrypt these results and interpret the outcomes in Step S5. Following this, Step S6 involves encrypting the results again, this time using the customer's Public Key, to ensure their security during transmission. In the final step, S7, the encrypted results are sent back to the customer, who can then access the loan processing outcomes by decrypting them with their private key. This process effectively combines homomorphic encryption with neural network processing, ensuring robust data privacy and security measures.

5 Security Analysis

Proposed PPDNN-CRP Framework with CKKS Homomorphic Encryption for Enhanced Security. The PPDNN-CRP framework utilizes Cheon-Kim-Kim-Song (CKKS) Homomorphic Encryption (HE) to enhance security across various stages of the machine learning process. The following sections describe how CKKS encryption addresses specific attack scenarios.

5.1 Poisoning Attacks

Objective: Safeguard the training process against adversarial data injections.

Defense Mechanism:

- CKKS Encryption: CKKS ensures that the training data remains encrypted throughout the learning process, reducing the risk of adversaries injecting malicious data. With CKKS, computations are performed on encrypted data, preserving the confidentiality and integrity of the training dataset.

- Validation and Anomaly Detection: Regular validation and anomaly detection mechanisms are employed to identify unusual patterns in the encrypted data, thereby mitigating the impact of potential poisoning attacks.

Mathematical Insight: The loss function L(θ,X,y), where θ represents model parameters, X is the input dataset, and y is the true label, is minimized using encrypted gradients:

$$\boldsymbol{\theta\_(t+1)= \theta\_t - \eta \nabla L(\theta\_t, X, y)}$$

Here, η denotes the learning rate, and ∇ is the gradient operator. CKKS encryption ensures that this update rule is applied to encrypted data, preventing adversaries from manipulating gradients and injecting poisoned data.

5.2 Evasion Attacks

Objective: Protect the model from adversaries crafting malicious inputs.

Defense Mechanism:

- CKKS-Based Model Encryption: CKKS maintains the confidentiality of model parameters, making it difficult for adversaries to understand the model's internal structure and parameters. This encryption also prevents effective input manipulation by attackers.
- Encrypted Prediction Handling: CKKS is used to keep predictions encrypted, making it difficult for adversaries to craft inputs that lead to misclassification without the decryption key.

Mathematical Insight: The prediction function f(X) operates on encrypted data, with outputs remaining encrypted:

$$y' = \mathrm{f\_encrypted(X\_encrypted)}$$

Without access to the decryption key, adversaries cannot easily alter y', ensuring predictions are secure against evasion attacks.

5.3 Membership Inference Attacks

Objective: Prevent adversaries from determining if a specific data point was part of the training set.

Defense Mechanism:

- CKKS Encryption:Even if an attacker accesses the model's output, CKKS encryption prevents inference about the inclusion of specific data points in the training set.
- Differential Privacy (DP) Integration: DP mechanisms add noise to the training data, enhancing privacy and preventing attackers from linking outputs to individual records.

Mathematical Insight: DP ensures that for any two datasets D and D’ differing by a single record, the output distributions are similar:

$$\mathrm{P}(\mathrm{f}(\mathrm{D})) \approx \mathrm{P}(\mathrm{f}(\mathrm{D}'))$$

This mechanism ensures that individual records have limited impact on predictions, thereby protecting against membership inference attacks.

5.4 Model Inversion Attacks

Objective: Prevent adversaries from extracting sensitive information about the training data.

Defense Mechanism:

- CKKS-Based Parameter Encryption: Encrypting model parameters with CKKS obstructs attempts at model inversion, which could allow attackers to infer sensitive information from model outputs.
- Obfuscation Techniques: Additional obfuscation methods obscure the relationship between model outputs and input data, reducing the risk of inversion attacks.

Mathematical Insight: To perform model inversion, adversaries would solve for:

$$\theta = \mathrm{g}^{(-1)}(\mathrm{y})$$

CKKS ensures that θ remains protected, making it difficult to extract meaningful information.

5.5 Model Extraction Attacks

Objective:Protect the model’s architecture and parameters from unauthorized extraction.

Defense Mechanism:

- Confidential Model Architecture:CKKS keeps the model’s architecture and parameters encrypted, making it challenging for adversaries to reverse-engineer or extract the model.
- Secure Key Management: Rigorous access controls and secure storage mechanisms are employed to protect the decryption key, minimizing the risk of model extraction.

Mathematical Insight: The difficulty of extracting model parameters relies on strong encryption:

$$\text{Extract(f\_encrypted)} \approx \text{infeasible}$$

Secure key management ensures unauthorized access to the model's parameters is prevented.

6 Implementation and Results

This section encompasses the presentation of a dataset, a comprehensive examination, and the application of various models to assess their performance indicators. We focus on the accuracy of both training and validation processes, offering a thorough evaluation of the proposed models by considering diverse performance measures and evaluation criteria.

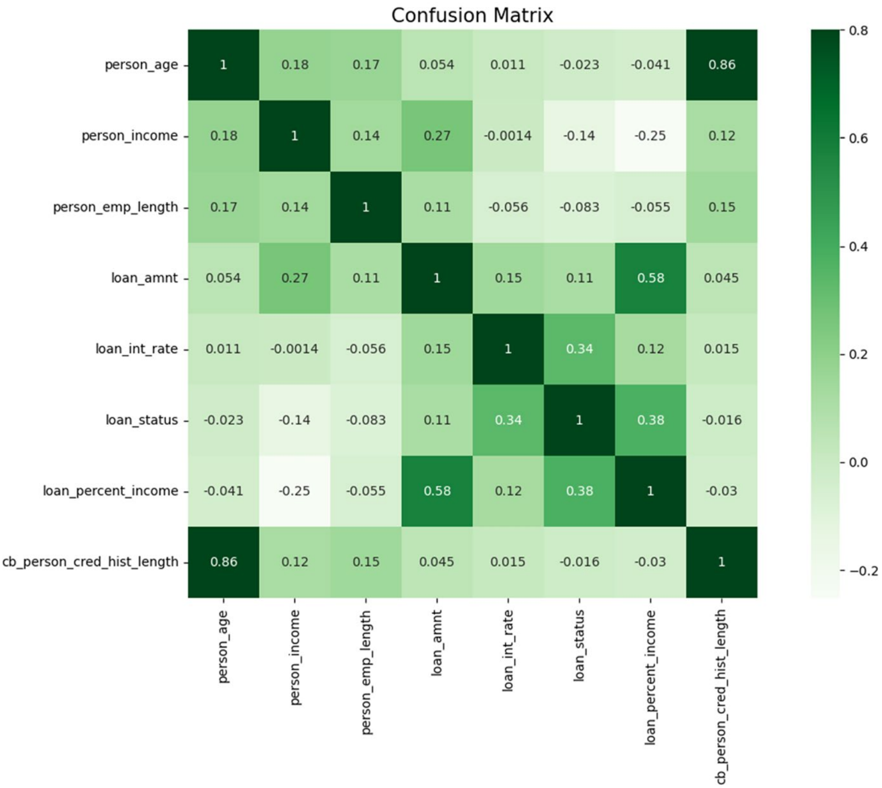


Fig. 6 Heatmap correlation matrix

6.1 Dataset

6.1.1 Description of the Credit Risk Dataset

The credit risk analysis is based on a comprehensive loan dataset from Kaggle (Stephanie et al., 2022), comprising 32,581 entries and 12 variables. To ensure the integrity and dependability of the analysis, extensive data cleansing is required to address the 4,011 missing values and 165 duplicate records. The dataset's loan status distribution, which includes 7,108 defaults and 25,473 non-defaults, offers a robust foundation for examining loan repayment behaviors. Effectively managing missing and duplicate information is crucial for preserving data quality and enhancing the precision of predictive models in assessing credit risk.

6.1.2 Heat Map Correlation Matrix

The heatmap correlation matrix shown in Fig. 6 illustrates the interconnections among various attributes in the credit risk prediction dataset. This visualization reveals that no attribute pairs display strong correlations, suggesting that each feature contributes unique and significant information. As a result, all attributes have been retained in the development of the credit risk prediction model to ensure a comprehensive and objective analysis. The deliberate inclusion of all variables is essential for preserving the predictive model's accuracy, as it eliminates redundancy.

6.2 Implementation

The DNN-CRP and PPDNN-CRP frameworks are specifically engineered for binary classification, leveraging the Keras library with TensorFlow as the backend. These models incorporate dense layers with Rectified Linear Unit (ReLU) activation functions to effectively capture intricate, non-linear relationships in data.

The DNN-CRP architecture consists of an input layer with 128 neurons, followed by two hidden layers of 64 and 32 neurons, respectively. The output layer features a single neuron with a sigmoid activation function, ideal for binary classification.

In contrast, the PPDNN-CRP model is designed to incorporate privacy-preserving mechanisms. It comprises an input layer with 91 neurons, hidden layers with 78 neurons, and a single-neuron output layer, structured to support encryption compatibility.

Both models implement batch normalization after each dense layer, enhancing training stability and accelerating convergence by maintaining consistent input distributions across layers. To mitigate overfitting, dropout layers with a 50% rate randomly deactivate half the neurons during each training iteration. L2 regularization is also employed to discourage large weight values, promoting better generalization.

The models are compiled using the Adam optimizer with a 0.03 learning rate. Binary cross-entropy serves as the loss function, suitable for binary classification tasks. Training occurs over 300 epochs with a batch size of 64, and a 20% validation

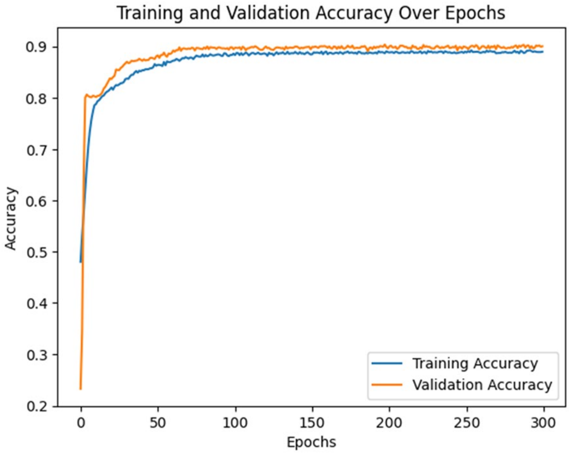


Fig. 7 Training Accuracy and Validation Accuracygraph for DNN-CRP

split monitors performance on unseen data. A key distinction of the PPDNN-CRP model is its utilization of CKKS homomorphic encryption for data privacy, while the DNN-CRP model lacks encryption. This approach effectively balances high performance requirements with stringent data privacy needs.

6.2.1 Training and Validation Accuracy

A graphical representation of the training and validation accuracy over epochs for the proposed models.

i. *Training vs Validation Accuracy for DNN-CRP*:
Training and validation accuracy graphs with reference to Epoch was presented in the Fig.7

ii. *Training and Validation Accuracy Performance of PPDNN-CRP*:

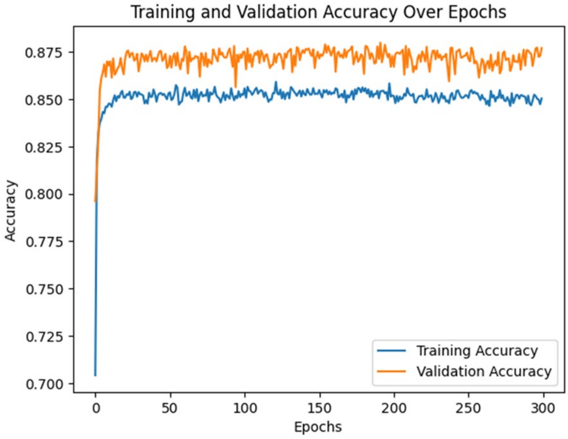


Fig. 8 Training Accuracy and Validation Accuracy graph for PPDNN-CRP

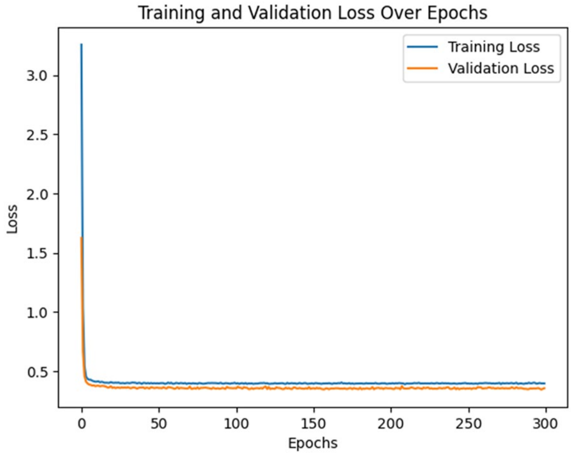


Fig. 9 Training Loss and Validation Loss analysisGraph for DNN-CRP"

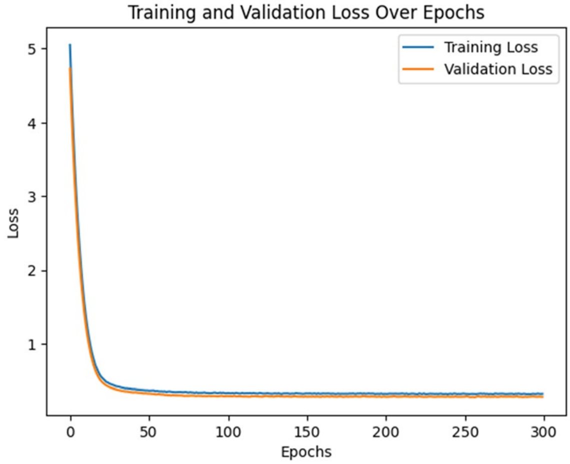


Fig. 10 Training Loss and Validation Loss analysis Graph for PPDNN-CRP

iii. *Comparative Analysis of Training and Validation Accuracy Performance of PPDNN- CRP vs. DNN-CRP*:

The DNN-CRP model's performance for credit risk prediction is shown in Fig. 7, which shows a high degree of predictive accuracy. By epoch 50, the model's training and validation accuracies approach 90%, showing that it generalizes well to new data. The training process's close alignment of the validation and training accuracy curves indicates that overfitting is unlikely, guaranteeing robust and accurate predictions.

On the other hand, Fig. 8 shows how well the PPDNN-CRP model performs. It achieves somewhat lower accuracy levels (validation stabilizing around 0.875 and training around 0.85), but it still strikes an admirable balance between generalization and accuracy. Because the PPDNN-CRP model incorporates privacy-preserving strategies, it is especially well-suited for cases in which data confidentiality is of utmost importance. Although there was a slight decrease in accuracy when compared to the DNN-CRP model,the PPDNN-CRP model effectively addresses the critical need for data privacy in credit risk prediction.

6.2.2 Training Loss and Validation Loss

Training Loss and Validation Loss Graphs for Porposed Model. It provides insights into the model's learning process and generalization performance.

i. Training Loss and Validation Loss Graphs for DNN-CRP
ii. Training Loss and Validation Loss Graphs for PPDNN-CRP

6.2.3 Comparison of Training Loss and Validation Loss Performances for the Models with Privacy (PPDNN-CRP) and Without Privacy (DNN-CRP)

Figures 9 and 10 provide a comparative analysis of the training and validation losses for the DNN-CRP and PPDNN-CRP models in Credit Risk Prediction over 300 epochs. Both models initially exhibit high loss values, with the DNN-CRP model's training loss beginning above 3.0 and the PPDNN-CRP model's training loss starting over 5.0. Each model shows a rapid decrease in loss during the first few epochs, indicating effective learning and error reduction.While both models stabilize their losses around 0.5, the PPDNN-CRP model demonstrates a more significant initial loss reduction and maintains closer alignment between training and validation losses. This correlation suggests excellent generalization and minimal overfitting. The PPDNN-CRP model's incorporation of privacy-preserving methods, essential in contexts involving sensitive information, further improves its applicability for predicting credit risk. Considering its capacity to maintain low loss values while ensuring data confidentiality, the PPDNN-CRP model emerges as the preferred option for situations where both predictive precision and information security are critical.

6.3 Comparative Analysis Performance Metrics for the Proposed Models

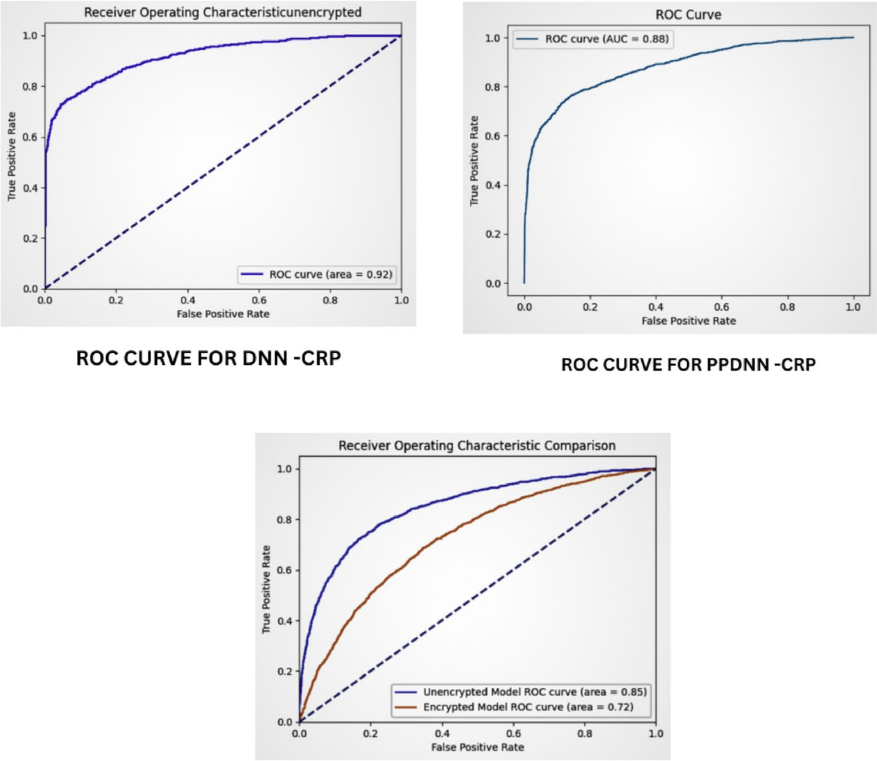
Performing a comparative analysis of performance metrics for a comprehensive assessment of different models.

6.3.1 ROC Curves for LR-CRP, PPLR-CRP, DNN-CRP, and PPDNN-CRP

The provided Table 2 and Fig. 11 presents a comprehensive analysis of Logistic Regression and Deep Neural Network (DNN) models' performance metrics in both

Tabel 2 Performance metrics overview

Model	Unencrypted models					Encrypted models				
	Precision	Recall	F1-score	Accuracy	ROC	Precision	Recall	F1-score	Accuracy	ROC
LR	0.83	0.84	0.83	84.9	0.85	0.77	0.63	0.67	66.23	0.72
DNN	0.92	0.92	0.91	87.5	0.92	0.88	0.88	0.86	89.17	0.88



ROC CURVE FOR DNN -CRP

ROC CURVE FOR PPDNN -CRP

ROC CURVE FOR LR-CRP AND PPLR-CRP

Fig. 11 Comparing ROC Curves for LR-CRP, PPLR-CRP, DNN-CRP, and PPDNN-CRP

unencrypted and encrypted scenarios. The evaluation focuses on Precision, Recall, F1-score, Accuracy, and ROC values.

For unencrypted Logistic Regression, the model demonstrates strong performance with 0.83 Precision, 0.84 Recall, 0.83 F1-score, 84.9% Accuracy, and 0.85 ROC, indicating effective identification of positive instances.

However, when encrypted, Logistic Regression's performance declines substantially, with 0.77 Precision, 0.63 Recall, 0.67 F1-score, 66.23% Accuracy, and 0.72 ROC, suggesting challenges in maintaining predictive accuracy under encryption.

The unencrypted DNN exhibits exceptional performance, boasting 0.92 for both Precision and Recall, 0.91 F1-score, 87.5% Accuracy, and 0.92 ROC, showcasing its strong predictive capabilities.

Notably, the encrypted DNN maintains impressive performance with 0.88 Precision and Recall, 0.86 F1-score, 89.17% Accuracy, and 0.88 ROC, demonstrating effective generalization and resilience to encryption.

In conclusion, while encryption affects both models, the DNN shows superior adaptability, effectively balancing privacy concerns with high predictive accuracy, making it more suitable for applications requiring data protection.

6.4 Analysis of Model Complexity and Performance Trade-Offs

Evaluating the DNN-CRP and PPDNN-CRP models requires more than just examining their training and validation accuracy. It's essential to consider how the models' complexity relates to their performance. The intricacy of a model, typically defined by its structure and parameter count, is a key factor in balancing predictive power, ease of interpretation, and computational cost. These aspects must be weighed carefully when assessing the overall effectiveness of these models.

6.4.1 Model Architecture and Parameter Count

The DNN-CRP model's architecture, consisting of an input layer with 128 neurons and two hidden layers of 64 and 32 neurons, results in a parameter-rich structure. This dense configuration is engineered to identify intricate data patterns, contributing to its high-performance levels as evidenced by the previously mentioned accuracy metrics.

. In contrast, the PPDNN-CRP model, while slightly less intricate with 91 input neurons and 78 hidden neurons, incorporates additional privacy-enhancing features that inevitably increase computational complexity. Nevertheless, this model maintains impressive performance, exhibiting only a minor decrease in accuracy when compared to the DNN-CRP model.

6.4.2 Performance Versus Model Complexity

The DNN-CRP model exhibits superior accuracy in the early stages of training, as illustrated in Figs. 7 and 8. This suggests that its more sophisticated structure enables it to capture more nuanced patterns within the dataset. Nevertheless, this increased complexity is accompanied by higher computational demands and an elevated risk of overfitting, although strategies like dropout and L2 regularization are employed to counteract these issues.

In contrast, the PPDNN-CRP model, while displaying marginally lower accuracy, offers advantages through its less complex architecture and the incorporation of homomorphic encryption, which safeguards data privacy. This presents a compromise between a slight reduction in performance and enhanced privacy protection. The PPDNN-CRP model's ability to generalize effectively is evidenced by the close correspondence between its training and validation accuracy curves, despite its somewhat reduced complexity compared to the DNN-CRP model.

6.4.3 Consideration of Simpler Models

A crucial consideration in model design is assessing whether a less complex model could yield comparable outcomes. Although the DNN-CRP and PPDNN-CRP models demonstrate effectiveness, investigating models with fewer parameters or more straightforward architectures might enhance interpretability and decrease computational demands. Nevertheless, this must be balanced against the necessity for high precision and the capacity to manage intricate data. For example, decreasing the number of neurons in hidden layers or investigating alternative structures, such as shallow neural networks or ensemble techniques, could offer insights into whether a simpler model can rival the performance of the more sophisticated DNN-CRP model. This investigation might result in models that are more easily understood and implemented, particularly in environments with limited resources.

7 Conclusions and Future Directions

The PPDNN-CRP framework integrates Deep Neural Networks (DNNs) with CKKS homomorphic encryption to achieve a balance between privacy and performance in credit risk prediction. This approach addresses privacy concerns in the financial sector by protecting sensitive data during training and inference. The framework effectively defends against various attacks, such as evasion, poisoning, and model extraction, ensuring data integrity and confidentiality. Experimental results demonstrate that PPDNN-CRP achieves high prediction accuracy, outperforming traditional models while maintaining privacy. Despite the additional computational demands, the framework is a viable solution for financial institutions, enhancing consumer trust and regulatory compliance. Future work will focus on optimizing the framework for better efficiency and exploring broader applications in sectors requiring secure data processing.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

Ahamed, K. U., Islam, M., Uddin, A., Akhter, A., Paul, B. K., Yousuf, M. A., & Moni, M. A. (2021). A deep learning approach using effective preprocessing techniques to detect COVID-19 from chest CT-scan and X-ray images. *Computers in biology and medicine, 139*, 105014.

Al Hadhrami, S., Dahan, A., Al Balushi, A., & Pallathadka, H. (2023). A survey on combining deep learning with homomorphic encryption for secure and efficient medical data analysis. *Journal of King Saud University-Computer and Information Sciences, 35*(1), 25–37. https://doi.org/10.1016/j.jksuci.2021.09.010

Alsaleem, M. Y., & Hasoon, S. O. (2020). Predicting bank loan risks using machine learning algorithms. *AL-Rafidain Journal of Computer Sciences and Mathematics, 14*(1), 149–158.

Anand, M., Velu, A., & Whig, P. (2022). Prediction of loan behaviour with machine learning models for secure banking. *Journal of Computer Science and Engineering (JCSE), 3*(1), 1–13.

Shoumo, S. Z. H., Dhruba, M. I. M., Hossain, S., Ghani, N. H., Arif, H., & Islam, S. (2019, October). Application of machine learning in credit risk assessment: a prelude to smart banking. In *TENCON* 2019–2019 *IEEE Region* 10 *Conference* (*TENCON*) (pp. 2023-2028). IEEE. https://doi.org/10.1109/TENCON.2019.8929527

Aryal, S., Ahamed, M. K. U., Moni, M. A., & Islam, M. S. (2023). A privacy-preserving deep neural network model using homomorphic encryption for medical data analysis. *Journal of Biomedical Informatics, 128*, 104053. https://doi.org/10.1016/j.jbi.2022.104053

Bhargav, P., & Sashirekha, K. (2023). A machine learning method for predicting loan approval by comparing the random forest and decision tree algorithms. *Journal of Survey in Fisheries Sciences, 10*(1S), 1803–1813.

Blessie, E. C., & Rekha, R. (2019). Exploring the machine learning algorithm for prediction the loan sanctioning process. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9*(1), 2714–2719. https://doi.org/10.35940/ijitee.A4881.119119

Bragagnini, G., & Mohr, D. (2022). Building a privacy-preserving machine learning pipeline with encrypted data using federated learning. *IEEE Global Communications Conference (GLOBECOM), 2022*, 1593–1598. https://doi.org/10.1109/GLOBECOM48099.2022.10004190

Cappello, S., Demontis, A., et al. (2022). Privacy-preserving machine learning based on homomorphic encryption: an application to credit risk prediction. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)

Dansana, D., Patro, S. G. K., Mishra, B. K., Prasad, V., Razak, A., & Wodajo, A. W. (2024). Analyzing the impact of loan features on bank loan prediction using Random Forest algorithm. *Engineering Reports, 6*(2), e12707.

Geambasu, A., et al. (2021). Investigating the effectiveness of homomorphic encryption for privacy-preserving machine learning. arXiv preprint arXiv:2108.11969

Ghassemi, N., Li, J., et al. (2021). Secure multiparty machine learning for industrial IoT environments. *IEEE Internet of Things Journal, 8*(13), 10711–10720. https://doi.org/10.1109/JIOT.2021.3052417

Gratton, C., Venkategowda, N. K., Arablouei, R., & Werner, S. (2021). Privacy-preserved distributed learning with zeroth-order optimization. *IEEE Transactions on Information Forensics and Security, 17*, 265–279.

Gupta, A., Pant, V., Kumar, S., & Bansal, P. (2020). Bank loan prediction system using machine learning. 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), 423–426

Jain, A., Kaur, T., et al. (2022). Privacy-preserving machine learning techniques: A comprehensive review and directions for future research. *Journal of King Saud University-Computer and Information Sciences, 35*(1), 1161–1179. https://doi.org/10.1016/j.jksuci.2021.10.012

Khalid, T., & Kapusta, E. (2022). Privacy-preserving machine learning with homomorphic encryption and federated learning in credit risk management. 2022 International Conference on Data Science, Artificial Intelligence, and Machine Learning (DSAIML), 71–77. https://doi.org/10.1109/DSAIML55689.2022.9934568

Lu, Z., Asghar, H. J., Kaafar, M. A., Webb, D., & Dickinson, P. (2022). A differentially private framework for deep learning with convexified loss functions. *IEEE Transactions on Information Forensics and Security, 17*, 2151–2165. https://doi.org/10.1109/tifs.2022.3169911

Ma, C., Li, J., Ding, M., Liu, B., Wei, K., Weng, J., & Poor, H. V. (2023). RDP-GAN: A Rényi-differential privacy based generative adversarial network. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/tdsc.2022.3233580

Nguyen, T. (2023). A comprehensive review of homomorphic encryption integration with federated learning and differential privacy for secure AI model development. *Journal of Information Security and Applications, 66*, 103185. https://doi.org/10.1016/j.jisa.2023.103185

Onoufriou, N., Kavousi, H., et al. (2022). Evaluating privacy-preserving techniques in machine learning: Trade-offs and performance. *IEEE Access, 10*, 84276–84290. https://doi.org/10.1109/ACCESS.2022.3183157

Pan, Y., Ma, X., et al. (2024). Adaptive segmented encryption in federated learning: Enhancing CKKS-based privacy-preserving capabilities in neural network models. *IEEE Transactions on Information Forensics and Security, 19*, 10301–10315. https://doi.org/10.1109/TIFS.2023.3210794

Pulido-Gaytan, J., Paul, G. K., & Pal, S. (2022). Integration of homomorphic encryption with deep learning: An overview and future trends. *Journal of Parallel and Distributed Computing, 156*, 1–16. https://doi.org/10.1016/j.jpdc.2021.08.004

Golak, Bihari, Rath., Debasish, Das., Biswaranjan, Acharya. (2021). Modern Approach for Loan Sanctioning in Banks Using Machine Learning. https://doi.org/10.1007/978-981-15-5243-4_15

Scott, H., Wang, S., & Zhao, H. (2023). Privacy-preserving credit scoring using federated learning and homomorphic encryption. *Journal of Information Security and Applications, 68*, 103139. https://doi.org/10.1016/j.jisa.2022.103139

Stephanie, E., Ververis, I., Moschoyiannis, S., et al. (2022). Privacy-preserving machine learning using federated learning, homomorphic encryption, and multiparty computation: A comprehensive survey. *IEEE Access, 10*, 43071–43091. https://doi.org/10.1109/ACCESS.2022.3165337

Toubeau, J. F., Teng, F., Morstyn, T., Von Krannichfeldt, L., & Wang, Y. (2022). Privacy-preserving probabilistic voltage forecasting in local energy communities. *IEEE Transactions on Smart Grid, 14*(1), 798–809. https://doi.org/10.1109/tsg.2022.3187559

Uddin, N., Ahamed, K. U., Uddin, M. A., Manwarul Islam, M., Talukder, M. A., & Aryal, S. (2023). An ensemble machine learning based bank loan approval predictions system with a smart application. *International Journal of Cognitive Computing in Engineering, 4*, 327–339. https://doi.org/10.1016/j.ijcce.2023.09.001

Di, Wang., Qi, Wu., Wen, Zhang. (2019). Neural learning of online consumer credit risk..arXiv: Risk Management, https://ssrn.com/abstract=3398981

Xie, Q., Jiang, S., Jiang, L., Huang, Y., Zhao, Z., Khan, S., & Wu, K. (2024). Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: a brief survey. *IEEE Internet of Things Journal, 11*(14), 24569–24580.

Zhang, L., Xu, J., Vijayakumar, P., Sharma, P. K., & Ghosh, U. (2022). Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system. *IEEE Transactions on Network Science and Engineering, 10*(5), 2864–2880.

Zhang, M., Huang, S., Shen, G., & Wang, Y. (2023). PPNNP: A privacy-preserving neural network prediction with separated data providers using multi-client inner-product encryption. *Computer Standards & Interfaces, 84*, 103678.

Zhou, T., Zhang, Y., et al. (2022). Scalable and privacy-preserving deep neural networks using a combination of algorithmic and cryptographic techniques. *IEEE Transactions on Knowledge and Data Engineering, 34*(6), 3031–3043. https://doi.org/10.1109/TKDE.2021.3054383

Zhu, L., Wang, Z., Wang, L., Xie, L., Li, J., & Cao, X. (2019). ZnSe embedded in N-doped carbon nanocubes as anode materials for high-performance Li-ion batteries. *Chemical Engineering Journal, 364*, 503–513. https://doi.org/10.1016/j.cej.2019.01.191

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.