Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

Research Article

Neural Network-Driven Privacy-Preserving Credit Risk Analysis: A Homomorphic Encryption Approach

V. V. L. Divakar Allavarpu[1], Vankamamidi S. Naresh[2*], A. Krishna Mohan[1]

[1] Computer Science Engineering, UCEK, JNTUK, Kakinada, 533003, East Godavari District, Andhra Pradesh, India
[2] Department of Computer Science and Engineering, Sri Vasavi Engineering College, Tadepalligudeam, 534101, Andhra Pradesh, India
E-mail: vsnaresh111@gmail.com

Received: 21 October 2024; **Revised:** 5 December 2024; **Accepted:** 11 December 2024

Abstract: With the increasing importance of credit risk analysis (CRA) with an emphasis on privacy, there's a notable need for a privacy-preserving machine learning (PPML) system. To address this demand, we propose a framework presenting a novel approach to privacy-preserving credit risk analysis (PPCRA) through integrating neural networks (NN) with homomorphic encryption (HE). The proposed framework offers robust privacy protection while maintaining the efficiency and accuracy of credit risk prediction systems. The implementation utilizes libraries such as TenSEAL and Torch to develop a HE-enabled NN model capable of processing encrypted data. Comprehensive security analysis establishes resilience against numerous privacy attacks of the system and empirical validation through experiments conducted on real-world financial datasets from multiple countries. The evaluation of the NN's performance, both with and without privacy preservation measures, provides insights into the efficacy of the proposed approach. This study offers significant advancements in privacy-preserving techniques for CRA, with implications for financial institutions and data security practitioners.

***Keywords*:** machine learning, artificial neural networks, privacy, homomorphic encryption, credit risk analysis, financial analytics

MSC: 65L05, 34K06, 34K28

Abbreviation

Acronym	Description
CRA	Credit Risk Analysis
ML	Machine Learning
NN	Neural Networks
HE	Homomorphic Encryption
FHE	Fully Homomorphic Encryption
PPML	Privacy-Preserving Machine Learning
PPCRA	Privacy-Preserving Credit Risk Analysis
PPNN	Privacy-Preserving Neural Networks

DOI: https://doi.org/10.37256/cm.6120255949

TenSEAL	Python SEAL software library for HE operations on Tensors.
CKKS	HE for Arithmetic of Approximate Numbers (HEAAN)

Symbol	Description
$(pk,\ sk)$	Public key and Private key
$\phi()$	Neural network classification model
N	Total number of samples in the dataset
X	Plain text vector input
Y	Plain text vector output
ω	Plain text vector weights
$[X]$	Encrypted vector input
$[Y]$	Encrypted vector output
$[\omega]$	Encrypted vector weights

1. Introduction

Credit risk analysis (CRA) [1] helps financial institutions evaluate the probability of a borrower default on a loan. This assessment enables them to make informed decisions regarding loan approval denial and set suitable interest rates and credit limits. It protects lenders from potential losses, supports responsible lending practices, and fosters trust in the financial system.

The finance industry faces a significant challenge in developing credit risk models to predict a borrower's likelihood of repaying a loan. Conventional statistical methods like Logistic Regression and Linear Discriminant Analysis are commonly employed in credit and customer risk assessment. However, they may face challenges when dealing with large-scale data analysis. Alternatively, machine learning (ML) [2] techniques, such as support vector machines (SVM), decision trees (DT), and NN, are increasingly utilized for predicting credit risk by leveraging behavioral and demographic data. ML can handle large datasets and provide optimized predictive models, contributing to improved accuracy in various financial tasks.

The financial institutions outsourced storage and computation in cloud-based services [3] allows for the training and prediction of ML models for CRA remotely without investing in specialized hardware. However, the most serious limitation for financial institutions is losing control over potentially sensitive data privacy. Financial institutions must maintain the privacy and security of borrowers' information to protect them from fraud initiatives. Attending to privacy issues in credit risk modelling isn't just a legal obligation. It is an essential element in upholding trust, ethical standards, and the sustained prosperity of financial institutions. Balancing the need for accurate credit risk assessment with privacy protection is crucial for responsible and sustainable financial analytics.

To safeguard the privacy and security of borrowers' personal information, PPML techniques [4] are employed to derive useful insights from the financial data while maintaining the confidentiality. One effective approach to tackle this is through the utilization of HE that enables computations on encrypted data. This is a substantial progress in the domine of cryptography and has important applications, particularly in privacy-preserving scenarios.

The data of a loan application undergoes encryption using HE, enabling the construction of a NN model using the encrypted data. Once the model training is complete, it can be utilized to predict outcomes on new loan applications without decryption. Subsequently, using HE, the output predictions ware decrypted to reveal the suitability of the credit application for approval. This strategy ensures the confidentiality and security of sensitive bank loan application data, empowering the bank to make informed decisions.

Preserving data privacy involves several steps, including input, output, training, and model privacy. Input privacy ensures data confidentiality during both training and inference, particularly when data is transmitted to an untrusted cloud server. Output data privacy involves protecting the confidentiality of information disclosed through a model's outputs or predictions during inference. Training privacy is crucial for safeguarding the confidentiality of training data and preventing

reverse engineering attempts. Model privacy aims to prevent the discovery of attributes and weight in the derived model, to deter theft by malicious entities.

Privacy-preserving credit risk analysis (CRA) models leverage advanced techniques like HE and secure multi-party computation to analyse encrypted borrower data without compromising confidentiality, ensuring compliance with data protection regulations like GDPR. HE enables computations on encrypted datasets, safeguarding sensitive information such as income and spending habits while facilitating secure collaborations between financial institutions. Combined with artificial intelligence (AI), these models improve the accuracy and fairness of credit evaluations by identifying patterns, automating risk assessments, and mitigating biases. AI further enhances this approach by enabling personalized, secure, and unbiased credit risk evaluations and fraud detection through federated learning and encrypted data processing. Despite challenges like computational complexity, advancements in HE schemes (e.g., CKKS, TFHE) make it increasingly efficient, allowing financial institutions to balance robust privacy with precision in risk assessment, fostering trust and regulatory compliance.

In this paper, we propose a CRA system that leverages an HE-aware neural network. This system is specifically designed to provide data privacy throughout all stages of the machine learning process, including input, output, training, and model.

1.1 *Contributions*

The main contribution of this study is the development of a PPCRA Framework, which includes:

• Construction of a HE-enabled NN model capable of operating on encrypted data.

• Conducting a comprehensive security analysis demonstrating the system's resilience against numerous privacy attacks, such as poisoning, member inference, evasion, model inversion, and model extraction.

• Performing experiments using authentic financial datasets from Germany, Japan, Australia, and Taiwan.

• Performance evaluation of the NN within the proposed system with and without privacy preservation measures.

The paper's structure is outlined as follows: Section 2 analyses of related work on privacy-preserving neural networks (PPNN). Section 3 introduces the background knowledge of NNs and HE. The PPCRA framework are proposed in the paper explained in Section 4. Section 5 conducts a security analysis, in Section 6 presents experiments demonstrating the efficacy and accuracy of this approach. Finally, Section 7 concludes by summarizing the key findings comprehensively.

2. Related work

There's been a growing interest in utilizing HE to safeguard privacy in data analysis, particularly in bank loan processing using artificial neural networks (ANN). This section presents recent advancements in PPNN for credit risk assessment systems.

A range of studies have explored the use of various prediction models. Ziru et al. [5], Vijaya et al. [2], Gide et al. [6], Ayad et al. [7], Mijwel et al. [8] emphasize the importance of ML algorithms including neural networks. With Ziru focusing on using historical transaction data and Vijaya "predicting credit risk in financial institutions using ensemble ML models". Li et al. [9] propose a "model for listed companies that combines a CNN-LSTM and an attention mechanism", while Balakrishnan et al. [10] developed "a credit risk model for Indian debt securities using ML techniques, including artificial NNs, support vector machines, and random forest". These studies collectively highlight the potential of ML and data mining in improving the accuracy of credit risk prediction models. However, these studies did not address privacy concerns regarding user data.

Various privacy-preserving techniques have been proposed for credit risk prediction. Zheng et al. [11] introduce PCAL, a framework based on adversarial learning that masks private information while maintaining utility. Maniar et al. [12] explore the application of differential privacy in credit risk modelling, evaluating its performance against a non-differentially private model. Andolfo et al. [13] evaluate the use of functional encryption for privacy-preserving credit scoring, highlighting its potential performance impact. Lin et al. [14] introduced "a privacy-preserving credit score system based on noninteractive zero-knowledge proof, which ensures that user information is not revealed during the

credit scoring process". These techniques offer promising solutions for maintaining data privacy in credit risk prediction. However, these models lack consideration for privacy protection in existing credit score computation, leaving user information vulnerable to potential leaks and increasing the risks of identity theft and credit card fraud.

Various investigations have explored the use of HE in credit risk prediction, with promising results. Allavarpu et al. [15] introduced a PPCRA framework using HE-aware logistic regression, demonstrating minimal accuracy differences compared to non-HE models. Nugent et al. [16] proposed a system for private fraud detection on encrypted transactions, achieving low latency and discussing use cases and deployment feasibility. Xiao et al. [17] developed "a privacy-preserved approximate classification algorithm based on HE, achieving feasible results for real-world problems". Cheon et al. [18] proposed "an ensemble method for privacy-preserving logistic regression, which reduces the number of iterations and improves performance". However, it suffers from higher execution costs due to the large number of iterations. Standard GD requires more iterations than the ensemble method. Bonte et al. [19] introduce logistic regression in machine learning, discuss the motivation for outsourcing computation to a cloud service, emphasize the need for privacy-preserving measures, and demonstrate the method's effectiveness in handling large datasets for real-life applications in medicine and finance. However, this model contains many approximations, which may lead to slightly worse performance compared to standard methods. The generalizability of the technique to other ML problems, such as NNs, is not straightforward and would require more complicated algorithms. These studies collectively suggest that HE holds promise for enhancing the privacy and security of credit risk prediction models.

Further, Amorim et al. [20] and Wingarz et al. [21] both highlight the potential of HE in preserving data privacy using NNs. Amorim et al. emphasize addressing a reliable and efficient privacy preservation approach. However, this model has limited support for advanced NN operations and scalability issues. Wingarz et al. discuss the significant overhead of running CNNs on homomorphically encrypted inputs. The ReActHE system by Song et al. [19] is a deep NN designed to facilitate privacy-preserving biomedical predictions through HE. This system aims to ensure the security of sensitive biomedical data while allowing for accurate predictions. It utilizes HE to enable computations on encrypted data without compromising privacy. Ivone et al. [22] "provide a comprehensive analysis of using HE for NN training and classification to enhance data privacy and security, highlighting challenges that need to be addressed for a reliable and efficient privacy preservation approach".

Given the constraints highlighted regarding the current CRA, there's a pressing need to develop a PPCRA. This arises from the necessity to balance evaluating individuals' creditworthiness effectively and safeguarding their sensitive personal data. The concept of a privacy-preserving CRA is founded on the imperative of maintaining an accurate risk assessment while upholding ethical and legal obligations to safeguard individuals' privacy, adhere to regulations, foster consumer trust, and bolster cybersecurity across the financial domain. Furthermore, Table 1 presents a comparison of various privacy-preserving methods discussed in the literature.

Table 1. Privacy-preserving methods and its applications

Sl. No	Paper	Insights	Applications	Limitations/Remarks
1	Jestine et al. [23]	The paper focuses on PPHE collective learning for in-hospital mortality prediction. Collective Learning protocol as mentioned in this paper presents a secure protocol to train a binary classifier model of time-series data using homomorphic encryption and logistic regression.	Medical applications.	The computational complexity of encrypted operations renders gradient descent training impractical. Previous works have focused solely on encrypted processes during the inference phase.
2	Mohammad et al. [24]	The study presents an integrated predictive accuracy algorithm for credit risk classification, utilizing ML classifiers like SVM, KNN, ANN, and DNN, and employing resampling techniques to improve prediction accuracy for default payments in imbalanced datasets.	Predicting default payments in credit risk assessments.	Limitations of current credit risk assessment methods discussed and no privacy for the user data.

Table 1. (cont.)

Sl. No	Paper	Insights	Applications	Limitations/Remarks
3	Emmanuel et al. [25]	The paper introduces an ML-based credit risk prediction model employing a classification that combines Gradient Boosting, Extreme Gradient Boosting and Random Forest. The model is evaluated across multiple datasets using metrics such as accuracy, AUC and F1-score.	Credit risk prediction for financial institutions.	No Privacy techniques used.
4	Yong et al. [26]	The paper presents an integrated graph representation learning approach for credit risk prediction, utilizing KNN for edge construction and GNN for node classification, enhancing predictive accuracy by combining unsupervised graph transformation with supervised classification, focusing solely on internal information.	Future applications of GNN in operational research tasks.	There is a need to explore methods for enhancing forecasting performance without relying on explicit relationships. Additionally, conducting a sensitivity analysis of the hyperparameter kk is crucial for optimal selection.
5	Yung et al. [27]	The paper discusses a multi-objective ensemble learning scheme for loan default prediction, which enhances credit risk analysis classification by integrating credit rating-specific features, improving predictive accuracy, and addressing the complexities of borrower behavior in default scenarios.	Loan default prediction using ensemble learning techniques.	Highly imbalanced class distribution in loan default prediction. Difficulty in achieving good classification accuracy.
6	Divakar et al. [15]	The paper presents a PPML framework for CRA using HE aware Logistic Regression (HELR), ensuring data privacy during training and inference phases while achieving satisfactory accuracy compared to non-HE models across various datasets.	PPCRA using HELR on various datasets.	Less Accuracy compared with proposed.
7	Lin et al. [14]	The paper proposes a PP credit score computation that utilizes Paillier encryption and zero-knowledge proofs to safe guard users information during credit risk analysis, addressing privacy concerns of existing credit score models.	Banking, financial institutions, insurance policy purchases and rental applications.	Privacy protection of existing CSC models is inadequate. Potential for leakage of user private information exists.
8	Ezgi et al. [28]	The paper focuses on privacy preserving classification algorithms, analyzing their performance on differentially private data. While it does not specifically address credit risk analysis, the techniques discussed can be applied to similar classification tasks requiring data privacy.	Privacy preserving classification in data mining applications.	Privacy levels decrease as classification accuracy improves.

Table 1. (cont.)

Sl. No	Paper	Insights	Applications	Limitations/Remarks
9	Qiao et al. [29]	The paper proposes a PP credit assessment system utilizing blockchain, featuring secure data sharing and multiparty computation. It employs linear transformation and homomorphic encryption to protect data privacy while enabling accurate credit risk analysis without exposing raw.	Privacy-preserving credit evaluation system. Secure sharing and multiparty computation.	Need for secure data sharing in multiparty computing.
10	Zheng et al. [11]	PCAL is a framework that utilizes adversarial learning to anonymize user data for credit risk modeling, balancing privacy protection and predictive utility. It aims to mitigate privacy leaks while maintaining effective risk analysis for financial institutions.	Credit risk modeling for loan decisions. Privacy-preserving machine learning for financial companies.	The effectiveness is contingent upon the robustness of the adversarial model, the complexity of balancing privacy and utility, and the specific datasets used for evaluation.
11	Maniar et al. [12]	The paper explores differential privacy in credit risk modeling, assessing its effectiveness in protecting customer data during model training. It compares the performance of differentially private models against non-differentially private models for banks.	Credit risk modeling for loan decisions.	Customer data leakage and mishandling risks. Need for privacy protections in model development.
12	Andolfo et al. [13]	The paper explores privacy-preserving credit risk scoring using functional encryption (FE), enabling users to learn only specific functions of encrypted data, thus enhancing security while addressing performance concerns associated with traditional methods like Homomorphic Encryption and Trusted Execution Environments.	Secure financial computations with Intel SGX and HE-based Zero-Knowledge Proofs.	High-performance overhead of homomorphic encryption (HE). Trusting Intel and availability of SGX hardware extension.
13	Nugent et al. [16]	System uses homomorphic encryption for private fraud detection on transactions. XGBoost model has better performance with low encrypted inference latency.	Private fraud detection on financial institutions.	Latency and storage requirements for encrypted inference. Complexity of securely deploying the neural network implementation.
14	Cheon et al. [18]	The paper discusses ensemble methods for PPLR based on HE. It focuses on an efficient algorithm using mini-batch enhanced Nesterov's accelerated gradient for training logistic regression on large encrypted datasets.	Evaluation on private financial data and public MNIST dataset.	Computational overhead increases training and inference times significantly. Inadequate support for advanced operations affects accuracy.

Table 1. (cont.)

Sl. No	Paper	Insights	Applications	Limitations/Remarks
15	Amorim et al. [19]	Homomorphic encryption (HE) enhances data privacy in neural networks (NNs) by enabling computations on encrypted data. However, challenges include computational overhead, limited support for advanced NN operations, and performance trade-offs, necessitating further research for optimization and scalability.	Milk yield forecasting in the agri-food sector.	Limited support for advanced NN operations affects accuracy and performance.
16	Song et al. [21]	ReActHE is a novel HE-friendly DNN designed for PP biomedical predictions. It utilizes a residue activation approach with a scaled power activation function, enabling secure computation over encrypted data while maintaining low approximation errors in various tasks.	Biomedical image datasets for privacy-preserving predictions and Secure machine learning evaluation.	Current homomorphic encryption supports limited arithmetic operations. Nonlinear activation functions hinder secure deep learning applications.

3. Background knowledge

This section offers a concise overview of NN and HE pertinent to the proposed system.

3.1 *Homomorphic encryption (HE)*

HE is a cryptographic method that enables computations to be conducted on encrypted data, preserving the confidentiality of sensitive information. It allows the third party to perform computations on encrypted cipher text without accessing the underlying plaintext. This is crucial for scenarios where privacy needs to be maintained while processing data. The core concept of HE ensures that operations on encrypted data yield results equivalent to those performed on unencrypted data upon decryption.

HE supports two basic operations on encrypted data:

1. **Addition (Homomorphic Addition):** If two numbers a and b are encrypted as $Enc(a)$ and $Enc(b)$, the addition operation $Enc(a)+Enc(b)$ produces $Enc(a+b)$ still in encrypted form.

2. **Multiplication (Homomorphic Multiplication):** Similarly, multiplying $Enc(a)$ and $Enc(b)$ produces $Enc(a\times b)$.

These operations form the foundation for performing more complex computations on encrypted data.

HE finds applications in various fields:

• **Secure Computation Outsourcing:** Third parties can compute encrypted data without accessing the raw information, ensuring privacy.

• **Privacy-Preserving Cloud Computing:** Users can store encrypted data on cloud servers while performing computations on it.

• **Machine Learning on Encrypted Data:** Enables training and inference on encrypted data, preserving data confidentiality.

• **Secure Multi-Party Computation:** Facilitates collaborative computations among multiple parties without disclosing private inputs.

Although HE offers robust privacy preservation, it involves computational overhead and complexity. The introduction of FHE by Gentry [22] in 2009 revolutionized the field, permitting computations on encrypted data. Our adoption of the CKKS [30] FHE scheme efficiently handles computations on encrypted data with real-number arithmetic operations. FHE empowers privacy-preserving computation across various applications without compromising data confidentiality.

3.2 *Artificial neural networks (ANN)*

ANNs are computational models inspired by biological NNs, like the human brain, characterized by interconnected nodes called neurons arranged into layers is shown in Figure 1.

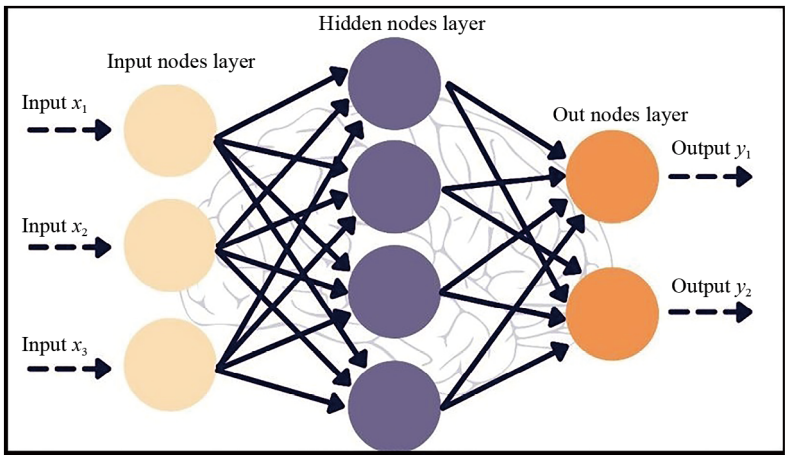

Figure 1. Artificial neural network structure

In CRA, ANNs are increasingly utilized to improve the accuracy of evaluating the credit worthiness of individuals or entities. This involves weights adjustment and biases of the NN to reduce the disparity among predicted outputs and actual target values. Through the training process, ANNs learn patterns and relationships inherent in the data. Here are the critical steps involved in training an ANN. The adoption of ANNs in CRA continues to grow, with financial institutions seeking to improve their credit scoring models' accuracy and predictive power. Regular validation, model monitoring, and adherence to regulatory standards are essential to deploying ANNs in credit risk.

3.3 *HE integrates with NN*

NN are widely used in tasks like image recognition, NLP, and medical diagnostics. However, training and using these networks often require access to sensitive data. Integrating HE with NN ensures:

1. **Data Privacy:** Sensitive data (e.g., medical records or financial details) never needs to be decrypted, safeguarding privacy.
2. **Secure Outsourcing:** Enables secure use of third-party computational resources without exposing raw data.
3. **Regulatory Compliance:** Addresses data protection laws like GDPR, HIPAA, etc.

The integration of HE with NN involves adapting neural network operations (e.g., matrix multiplications, activation functions) to work on encrypted data. Here's a simplified workflow:

1. **Encryption:** Input data (e.g., images, text) is encrypted using a public encryption key.
2. **Processing:** The encrypted data is fed into the neural network. Key computations, like forward passes, are performed directly on the encrypted data.
3. **Decryption:** The output remains encrypted until the user decrypts it with their private key.

This process ensures that at no point is the raw data exposed, even during computation (Figure 2).

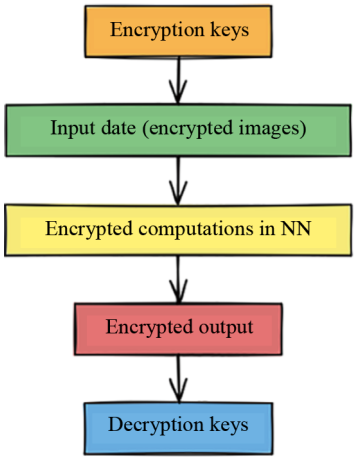

Figure 2. Workflow of HE integration with NN

4. Proposed system

4.1 *System model*

The proposed CRA system model involves three key entities: Customer (C), Bank (B), and Cloud Service Provider (CSP), as illustrated in Figure 3. The CSP is a semi-trusted third party and it provides extensive storage and computation resources through the Internet. The resources are used to compute on encrypted data while maintaining privacy details of the users.

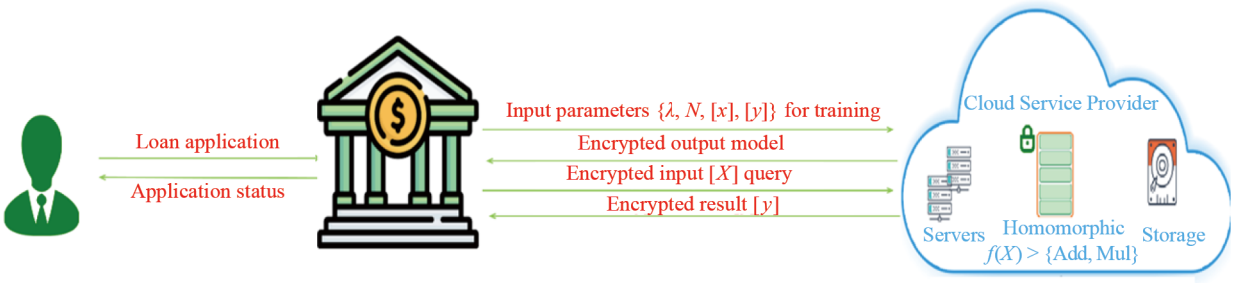

Figure 3. System model for PPCRA

Customer (C): Represents individuals or organizations seeking loans from the bank. To apply for a loan, customers provide sensitive personal information. Customers encrypt their data before outsourcing to safeguard privacy on the public cloud.

Bank (B): An institution offering loans and other financial services to consumers based on their credit history. With limited resources, the bank utilizes the CSP for data analysis services on encrypted data to train an NN model without compromising training data privacy. Encrypted training data is stored on the CSP. The bank utilizes the encrypted NN model for loan decision-making.

4.2 *Privacy-preserving credit risk analysis framework (PPCRAF)*

The framework comprises three phases:

(i) Privacy-preserving artificial neural network training (PPANNT).

(ii) Privacy-preserving prediction query (PPPQ).

(iii) Privacy-preserving result extraction (PPRE).

i. **Privacy-preserving artificial neural network training (PPANNT)**

This phase involves training an NN classifier over encrypted CRA data owned by the bank. The process follows the CKKS mechanism.

• Initially, plaintext data is encrypted with the bank's public key (pk_b) to generate ciphertext, outsourced to the CSP for building the encrypted NN model.

• Upon receiving the encrypted training dataset $\mathscr{D} = \lceil x_i \rceil_{pk_b}, \lceil y_i \rceil_{pk_b}$, where $1 \leq i \leq N$ and y_i represents the binary class label (0 or 1), the CSP initializes weights and biases randomly.

• Forward propagation computes the network's output based on input data, current weights, and biases, utilizing an activation function approximation for each neuron.

• Backpropagation calculates the gradient of the loss function for network parameters. This gradient updates weights and biases using optimization algorithms like gradient descent.

• The optimized approximation of $\lceil y_i \rceil_{pk_b}$ is obtained using the classifier model $\phi(\lceil x_i \rceil_{pk_b}, \lceil y_i \rceil_{pk_b})$, where $\phi(\mathscr{D}, \theta)$ denotes the hypothesis class, and θ represents a specific hypothesis parameter.

We depicted CRAF processing in the above phases in Figure 4.

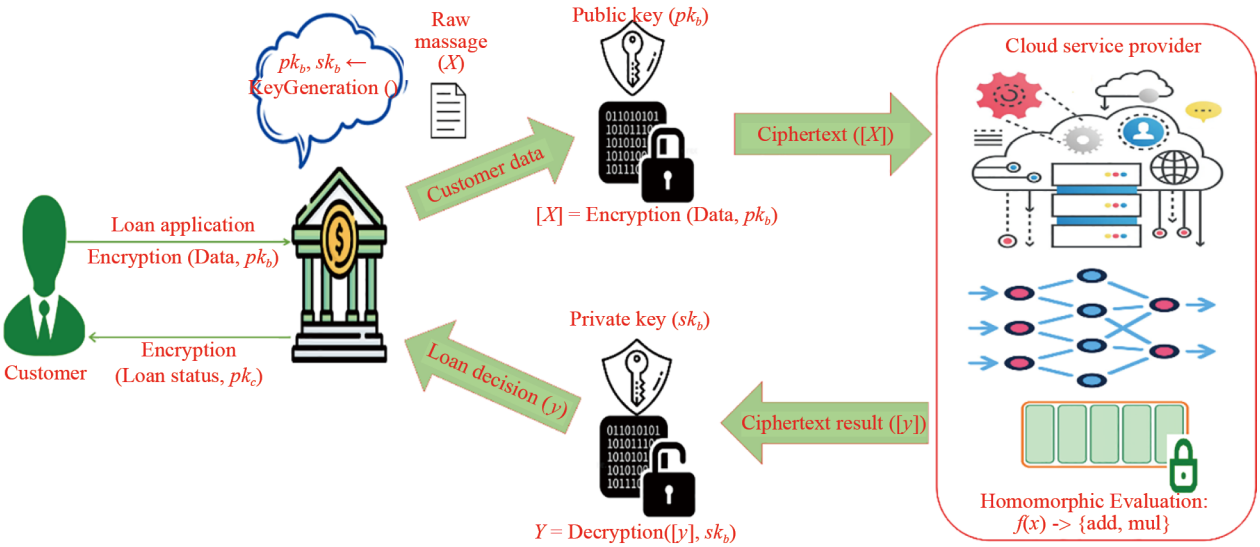

Figure 4. PPCRA framework

The CRA based on the artificial neural network (ANN) computational model is shown in Figure 5.

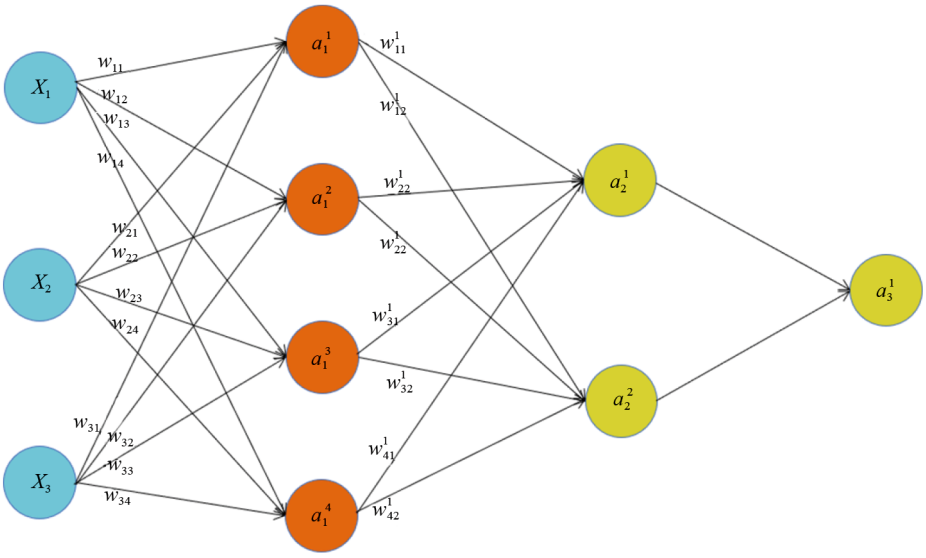

Figure 5. Artificial neural network computational model

The proposed NN mathematical formulation steps for the classification of plain text inputs are summarized as follows:

1. Let's consider the input vector: $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ with other network parameters, such as weight matrices W_1 and bias vector b_i are initialized as follows.

$$W_1 = \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \\ W_{14} & W_{24} & W_{34} \end{bmatrix}, \quad b_1 = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \end{bmatrix}$$

2. The input data is fed into the input layer of the network. After that, each neuron in the first hidden layer receives inputs from the input data layer. The inputs are multiplied by corresponding weights, and these weighted inputs are summed up as:

$$Z_1 = W_1 X + b_1 = \begin{bmatrix} W_{11}x_1 + W_{21}x_2 + W_{31}x_3 + b_{11} \\ W_{12}x_1 + W_{22}x_2 + W_{32}x_3 + b_{12} \\ W_{13}x_1 + W_{23}x_2 + W_{33}x_3 + b_{13} \\ W_{14}x_1 + W_{24}x_2 + W_{34}x_3 + b_{14} \end{bmatrix} = \begin{bmatrix} Z'_{11} \\ Z'_{21} \\ Z'_{31} \\ Z'_{41} \end{bmatrix} \tag{1}$$

3. Next, each neuron in a second hidden layer receives inputs from the first hidden layer. The inputs are multiplied by corresponding weights, and these weighted inputs are summed up as follows:

$$W_2 = \begin{bmatrix} W_{11}^{'} & W_{21}^{'} & W_{31}^{'} & W_{41}^{'} \\ W_{12}^{'} & W_{22}^{'} & W_{32}^{'} & W_{42}^{'} \\ W_{13}^{'} & W_{23}^{'} & W_{33}^{'} & W_{43}^{'} \\ W_{14}^{'} & W_{24}^{'} & W_{34}^{'} & W_{44}^{'} \end{bmatrix}$$

$$b_2 = \begin{bmatrix} b_{11}^{'} \\ b_{21}^{'} \end{bmatrix}$$

Compute

$$Z_2 = W_2 Z_1 + b_2 = \begin{bmatrix} W_{11}^{'} Z_{11}^{'} + W_{21}^{'} Z_{21}^{'} + W_{31}^{'} Z_{31}^{'} + W_{41}^{'} Z_{41}^{'} + b_{11}^{'} \\ W_{12}^{'} Z_{11}^{'} + W_{22}^{'} Z_{21}^{'} + W_{32}^{'} Z_{31}^{'} + W_{42}^{'} Z_{41}^{'} + b_{21}^{'} \end{bmatrix} \tag{2}$$

4. A Sigmoid activation function is applied to the weighted sum Z_2 to introduce non-linearity into the network. The output of the activation function becomes the input to the next layer of neurons. This process is repeated for each layer until the output layer arrives.

$$Z_3 = \text{Sigmod}\,(Z_2) = \frac{1}{1+e^{-Z_2}} = Z_3^1,\ Z_3^2 \tag{3}$$

5. Finally, the output of the NN is compared to the expected output, and the arg max function is often used during inference to determine the predicted class C_i, $i = 1, 2$ based on the output probabilities from the network's final layer for classification.

$$C_i = \text{argmax}\,(Z_3) = \frac{1}{1+e^{-Z_2}} = \text{argmax}\left(Z_3^{'},\ Z_3^{''}\right) = C_1 \ \text{ or } \ C_2 \tag{4}$$

Encrypted model

The proposed privacy-preserving artificial NN-based CRA with the above methodology is applied on encrypted text inputs are summarized as follows:

1. **Input layer processing:**

Let's consider the encrypted input polynomial vector: $[X] = \begin{bmatrix} [x_1] \\ [x_2] \\ [x_3] \end{bmatrix}$, initialized weight matrices $[W]$, bias vector $[b_i]$.

$$[W_1] = \begin{bmatrix} [W_{11}] & [W_{21}] & [W_{31}] \\ [W_{12}] & [W_{22}] & [W_{32}] \\ [W_{13}] & [W_{23}] & [W_{33}] \\ [W_{14}] & [W_{24}] & [W_{34}] \end{bmatrix}$$

$$[b_1] = \begin{bmatrix} [b_{11}] \\ [b_{21}] \\ [b_{31}] \\ [b_{41}] \end{bmatrix}$$

$$[Z_1] = [W_1 X + b_1] = \begin{bmatrix} [W_{11}x_1 + W_{21}x_2 + W_{31}x_3 + b_{11}] \\ [W_{12}x_1 + W_{22}x_2 + W_{32}x_3 + b_{12}] \\ [W_{13}x_1 + W_{23}x_2 + W_{33}x_3 + b_{13}] \\ [W_{14}x_1 + W_{24}x_2 + W_{34}x_3 + b_{14}] \end{bmatrix} = \begin{bmatrix} [Z'_{11}] \\ [Z'_{21}] \\ [Z'_{31}] \\ [Z'_{41}] \end{bmatrix} \tag{5}$$

2. **First, hidden layer processing**

$$[W_2] = \begin{bmatrix} [W'_{11}] & [W'_{21}] & [W'_{31}] & [W'_{41}] \\ [W'_{12}] & [W'_{22}] & [W'_{32}] & [W'_{42}] \\ [W'_{13}] & [W'_{23}] & [W'_{33}] & [W'_{43}] \\ [W'_{14}] & [W'_{24}] & [W'_{34}] & [W'_{44}] \end{bmatrix}$$

$$[b_2] = \begin{bmatrix} [b'_{11}] \\ [b'_{21}] \end{bmatrix}$$

Compute

$$[Z_2] = [W_2 Z_1 + b_2] = \begin{bmatrix} [W'_{11}Z'_{11} + W'_{21}Z'_{21} + W'_{31}Z'_{31} + W'_{41}Z'_{41} + b'_{11}] \\ [W'_{12}Z'_{11} + W'_{22}Z'_{21} + W'_{32}Z'_{31} + W'_{42}Z'_{41} + b'_{21}] \end{bmatrix} \tag{6}$$

3. **Second hidden layer processing**
Compute

$$[Z_3] = \text{Sigmod}([Z_2]) = \frac{1}{1 + e^{-[Z_2]}} = ([Z_3^1],\ [Z_3^2]) \tag{7}$$

Here, $\text{Sigmod}([Z_2])$ is polynomial approximation of encrypted data with the homomorphic properties are: $[X] + [Y] = [X + Y]$ and $[X] * [Y] = [X * Y]$.

4. **Output layer processing**

$$[C_i] = \text{argmax}([Z_3]) = \frac{1}{1+e^{-[Z_2]}} = \text{argmax}\left([Z_3'],\ [Z_3'']\right) = [C_1] \text{ or } [C_2] \tag{8}$$

ii. **Privacy-preserving prediction query (PPPQ)**

The Classification model $\phi(.)$ generated over an encrypted data by using the parameters θ^* by NN-training phase. The bank quries with m attributes $\mathbf{z} = (z_1, z_2, \ldots, z_m)$, and encrypts it as $[\mathbf{z}]_{pk_b} \leftarrow \text{Encrypt}(\mathbf{z}, pk_{pk_b})$, in the CSP. The CSP executes the classifier $\phi(.)$ on the new record $[\mathbf{z}]_{pk_b}$ using the encrypted parameter $[\theta^*]_{pk_b}$ to generate the prediction $[\mathbf{y}]_{pk_b} = \phi([\mathbf{z}]_{pk_b}, [\theta^*]_{pk_b})$.

iii. **Privacy-preserving result extraction (PPRE)**

After getting the prediction results $[\mathbf{y}]_{pk_b}$, the CSP sends it to the bank. Then the bank decrypt using $y \leftarrow \text{Decrypt}(sk_{pk_b}, [\mathbf{y}]_{pk_b})$ with bank's private key sk_b to retrieve the original prediction. Following the classification outcome, the bank informs the customer about the approval or rejection of the credit in an encrypted format.

5. Security analysis

In this section, we demonstrated a range of privacy attacks on HE-based NN solutions to defend against these attacks. Initially, we have established various possible privacy attacks on NN over financial data are visualized in Figure 6.

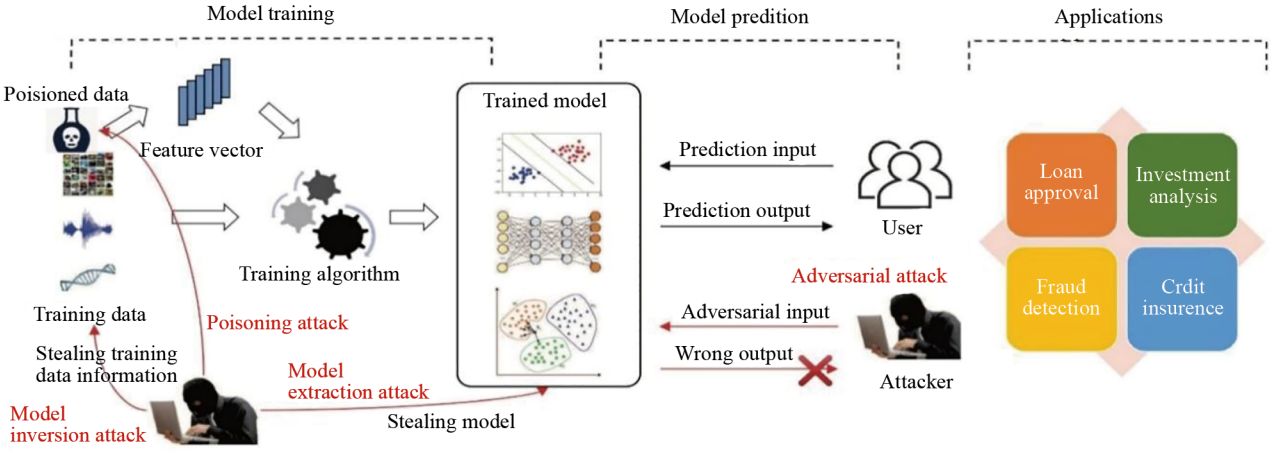

Figure 6. Privacy attacks on neural networks

The NN model can be defined as follows:

Let X be an $n \times m$ matrix representing the input data, where n is the number of data points, and m is the number of features. Each row of X corresponds to a data point, and each column represents a feature. Let Y be $1 \times n$ column vector for labeled data, where $Y = 0$ or 1, denoting the binary classification. Let L be the number of layers in NN, including the input and output layers. The activation of the neurons in the output and hidden layers is using activation functions. Let $\sigma(z)$ denote the activation function applied element-wise to the vector z.

The output of the i-th neuron in the l-th layer, denoted as $a_i^{(l)}$, is computed as follows:

$$a_i^{(l)} = \sigma\left(\sum_{j=1}^{n^{(l-1)}} w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)}\right) \tag{9}$$

where: $w_{ij}^{(l)}$ is the weight of the the j-th neuron in layer $l-1$ and the i-th neuron in layer l, $b_i^{(l)}$ is the bias, and $-a_j^{(l-1)}$ is the activation.

The weights and biases of the NN are attaind in training process with optimization algorithms such as gradient descent. Training aims to minimize a predefined loss function, typically the cross-entropy loss for binary classification tasks.

The prediction of the NN for a given input x is obtained by forward propagating x through the network to compute the output of the output layer. If the output is more significant than a threshold (typically 0.5 for binary classification), the prediction is 1; otherwise, it is 0.

5.1 *Poisoning attack's mathematical model*

A poisoning attack on an NN involves modifying the training data and labels to compromise the model's performance. Let's denote the original training data as X and the true labels as Y. The attacker's goal is to craft perturbations to X and Y to create a poisoned dataset, denoted as X_{poisoned} and Y_{poisoned}, respectively.

Mathematically, the attacker aims to maximize the NN's loss function by perturbing X and Y, while ensuring that the poisoned dataset still yields similar predictions as the original. This is typically formulated as an optimization problem:

$$\text{maximize } L(\theta) \text{ subject to } Y_{\text{poisoned}} = f(X_{\text{poisoned}}, \theta) \tag{10}$$

Here, $L(\theta)$ represents the loss function of the NN model, and $f(\cdot)$ represents the forward propagation process of the NN with parameters θ.

The success of the attack depends on factors like the NN's architecture, the training algorithm used, and the attacker's understanding of the model's vulnerabilities.

Poisoning attack's defense through HE:

A HE-based solution to defend against poisoning attacks on an NN involves encrypting the training data and model parameters, allowing computations on encrypted data.

Here's how it works:

• Encrypt both the training data X and the model parameters ω using FHE before training the model:

$$[X], [\omega] = \text{HE} \cdot \text{Encryption}(X, \omega)$$

• Perform gradient descent updates on the encrypted data and parameters to compute the new model parameters without decryption:

$$[\omega'] = [\omega] - \alpha \cdot [X] \cdot ([Y] - [Y'])$$

• The attacker cannot manipulate the data or model parameters directly on encrypted data.

Since computations occur on encrypted data and weights entire training phase. The attacker cannot modify the weights and training data without key. Consequently, HE can serve as a defense mechanism against poisoning attacks on NNs.

5.2 *Evasion attack's mathematical model*

In this evasion attack, the attacker alters the input data to appear authentic while deceiving the model's forecasts.

Let $f: \mathbb{R}^n \to \{1, 2, \ldots, C\}$ be a NN classifier that maps an input vector $\mathbf{x} \in \mathbb{R}^n$ to one of C class labels. The goal of an evasion attack is to find a perturbed input $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$ such that $f(\mathbf{x}) \neq f(\mathbf{x}')$ while keeping $\boldsymbol{\delta}$ small.

The primary objective is to maximize the misclassification error:

$$\max_{\delta} L(f(\mathbf{x}+\boldsymbol{\delta}), y) \tag{11}$$

where L is the loss function (e.g., cross-entropy loss), y is the true label of x, and δ is the perturbation vector.

The attacker modifies the input vector $x_{i'}$ by adding a small perturbation Δx_i, such that: $x_{i'} = x_i + \Delta x_i$

$$\text{Minimize } \| \Delta \mathbf{x}_i \| \tag{12}$$

$$\text{subject to} f(x_i^{'}) \neq y_i$$

where $f(x_i^{'})$ is the predicted label.

Evasion attack's defense through HE:

We aim to demonstrate that HE operations to perform on encrypted data that do not compromise the NN model performance. This assurance is pivotal, as it implies that even if an attacker manipulates the encrypted input data, the resulting decrypted output remains a precise prediction of the original NN model.

Let's consider the scenario where an attacker attempts to modify the encrypted input vector $[X]$ by introducing a perturbation vector $\text{Enc}(P)$. Consequently, the new encrypted input becomes:

$$[X_{\text{new}}] = [X] + [P] \tag{13}$$

The attacker's objective is to induce a discrepancy between the decrypted output, Y_{new} and the original output Y, achieved through careful selection of the perturbation vector $[P]$.

We assume that a maximum norm constrains the attacker's perturbation vector, denoted as $\| [P] \| \leq \varepsilon$, where ε represents a small value. Such an assumption is typical in evasion attacks, where the attacker's ability to modify input data is restricted.

To establish the capability of HE in preserving the NN model's accuracy amidst such attacks, we seek to prove that the disparity between the decrypted outputs y and Y_{new} remains bounded by a small value, even subsequent to the addition of the perturbation vector to the encrypted input as given:

• Encrypt the original input data x using the public key: $c = \text{Enc}_{pk}(x)$.

• Generate a small perturbation Δc in the encrypted domain.

• Add the perturbation to the encrypted input: $c' = c + \Delta c$.

• Perform forward propagation on the encrypted perturbed input using the encrypted model parameters to obtain an encrypted prediction: $c'' = f(c')$.

• Decrypt the prediction using the private key: $\widehat{y}' = Dec_{sk}(c'')$.

• Compare the decrypted prediction $\widehat{y}'$ with the true label y to assess the effectiveness of the evasion attack.

By integrating HE into the defence mechanisms against evasion attacks on NNs, we can ensure the confidentiality and integrity of both the model and the input data, thereby mitigating the effectiveness of adversarial perturbations. Additionally, leveraging HE for secure computation enables robust defences against evasion attacks while preserving the privacy of sensitive information.

5.3 *Member inference attack's mathematical model*

Member inference attacks on NNs involve adversaries attempting to determine whether specific data points were part of the training dataset used to train the model.

Let D represent the training dataset containing N data points, each with n features: $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ is the input data and y_i is the corresponding label. Consider a trained NN classifier f: $\mathbb{R}^n \to \{0, 1\}$ that predicts whether an input data point belongs to the training dataset ($f(x) = 1$) or not ($f(x) = 0$).

The attacker's goal is to infer whether a given input data point x' was part of the training dataset by exploiting the predictions of the NN model f. For a given input x', the attacker observes the model's prediction $f(x')$. If $f(x') = 1$, the attacker infers that x' was likely a member of the training dataset; otherwise, x' is considered non-member.

Member inference attack's defense through HE:

To safeguard against membership inference attacks employing HE, a strategy involves encrypting both the training data and the weights using FHE. This approach enables computations to occur on encrypted data without disclosing the original plaintext.

Let's consider the Public Key: pk and Private Key: sk. The encrypted training dataset is $D_{\text{enc}} = \{\text{Enc}_{pk}(x_i, y_i)\}_{i=1}^{N}$ and encrypted model parameters: Θ_{enc}.

• Encrypt the training dataset and the NN model parameters using the public key.

- NN Model: $f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$,
- Encrypted Input Data: $x_{\text{enc}} = \text{Enc}_{pk}(x)$,
- Encrypted Prediction: $\widehat{y}_{\text{enc}} = f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$.

• Train the NN model on the encrypted training dataset using encrypted computations.

- Encrypted training dataset: $D_{\text{enc}} = \{\text{Enc}_{pk}(x_i, y_i)\}_{i=1}^{N}$,
- Encrypted model parameters: Θ_{enc}.

• Homomorphically forward and backward propagation are performed to maintain the privacy of the training process, which will generate:

- NN Model: $f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$.

• In the inferencing process using encrypted input data: $x_{\text{enc}} = \text{Enc}_{pk}(x)$ forecast encrypted prediction: $\widehat{y}_{\text{enc}} = f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$.

• The decryption function is used to get the plain text prediction as follows:

- $Y_{\text{pred}} = \text{Dec}_{sk}([\widehat{y}_{\text{enc}}])$.

Using HE, it is possible to defend against member inference attacks on NNs while preserving the privacy of individual data points. By encrypting both the training data and the model parameters and ensuring that all computations are performed homomorphically, sensitive information is protected from adversaries attempting to infer membership.

5.4 *Inversion attack's mathematical model*

Model inversion attacks involve reconstructing training data from the model's outputs. We defend against these attacks by encrypting the weights and adding noise to the predicted probability vector.

Consider a trained NN model f: $\mathbb{R}^n \to \mathbb{R}^m$, where n is the input dimension, and m is the output dimension. The model is trained on a dataset D consisting of input-output pairs $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ represents the input data and $y_i \in \mathbb{R}^m$ represents the corresponding sensitive information about individuals.

The attacker aims to infer sensitive information y_i associated with a specific input x_i by exploiting the trained NN model f. By applying the reverse mapping g to the model's output $f(x_i)$, the attacker attempts to reconstruct the original input data x_i, thereby inferring the sensitive information associated with it.

Inversion attack's defense through HE:

Defending against model inversion attacks in NNs using HE involves protecting the privacy of sensitive information while still allowing the NN to perform its intended tasks. Here's a HE-based solution for defending against model inversion attacks:

Let's consider the Public Key: pk and Private Key: sk. The encrypted training dataset is $D_{\text{enc}} = \{\text{Enc}_{pk}(x_i, y_i)\}_{i=1}^{N}$ and encrypted model parameters: Θ_{enc}.

• Encrypt the training dataset and the NN model parameters using the public key.

- NN Model: $f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$,
- Encrypted Input Data: $x_{\text{enc}} = \text{Enc}_{pk}(x)$,
- Encrypted Prediction: $\widehat{y}_{\text{enc}} = f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$.

• Train the NN model on the encrypted training dataset using encrypted computations.

- Encrypted training dataset: $D_{\text{enc}} = \{\text{Enc}_{pk}(x_i, y_i)\}_{i=1}^{N}$,
- Encrypted model parameters: Θ_{enc}.

• Homomorphically forward and backward propagation are performed to maintain the privacy of the training process, which will generate:

- NN Model: $f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$.

• In the inferencing process using encrypted input data: $x_{\text{enc}} = \text{Enc}_{pk}(x)$ forecast encrypted prediction: $\widehat{y}_{\text{enc}} = f_{\text{enc}}(x_{\text{enc}}, \Theta_{\text{enc}})$.

• Decrypt the output using decryption function:

- $Y_{\text{pred}} = \text{Dec}_{sk}([\widehat{y}_{\text{enc}}])$.

• Add noise to the output of the NN to mask sensitive information and prevent attackers from accurately inferring it.

• We add some noise to the predicted output to prevent the attacker from accurately reconstructing the input data.

- $\widehat{y}_{\text{pred}} = Y_{\text{pred}} + \text{Noise}$.

This representation outlines the steps in the HE-based defence against model inversion attacks in NNs. By encrypting both the input data and the model parameters and ensuring that all computations are performed homomorphically, the privacy of sensitive information is preserved. Additionally, incorporating privacy-preserving techniques such as randomized response enhances the defence mechanism's robustness against model inversion attacks.

5.5 *Model extraction attack's mathematical model*

Model extraction attacks aim to extract a copy of a target NN model by querying it and using the responses to train a surrogate model.

Consider a target NN model. $F_{\text{target}}: \mathbb{R}^n \to \mathbb{R}^m$ with parameters Θ_{target}. The attacker's objective is to extract a copy of the target model by querying it and using the responses to train a surrogate model.

The attacker aims to approximate the behaviour of the target model F_{target} by training a surrogate model $F_{\text{surrogate}}: \mathbb{R}^n \to \mathbb{R}^m$ with parameters $\Theta_{\text{surrogate}}$.

The attacker queries the target model with input data points x_i to obtain the corresponding outputs $F_{\text{target}}(x_i)$. The attacker uses the input-output pairs $\{(x_i, F_{\text{target}}(x_i))\}_{i=1}^{N}$ to train the surrogate model $F_{\text{surrogate}}$.

Model extraction attack's defense through HE:

Defending against model extraction attacks in NNs using HE involves protecting the confidentiality of the model parameters and the privacy of the model's responses during querying.

• Encrypt the input data points x_i using the public key. Query the encrypted model with the encrypted input data to obtain the encrypted responses $F_{\text{target}}(x_i)$.

• Keep the model parameters encrypted throughout the querying process to prevent adversaries from accessing them directly.

• Incorporate noise into the encrypted responses before returning them to the querying party.

By leveraging HE, it is possible to defend against model extraction attacks on NNs while preserving the confidentiality of the model parameters and the privacy of the model's responses. By encrypting both the model parameters and the input data and ensuring that all computations are performed homomorphically, sensitive information is protected from adversaries attempting to extract a copy of the model.

6. Experiments and result analysis

This section presents experimental results and parameter sets used in our implementation, leveraging using TenSEAL library [21]. This library enables the construction of CKKS scheme for PPML using HE. All experiments were conducted on an Intel-i5-10500 CPU running at a 3.10 GHz processor with four cores and 16 GB of RAM.

Datasets

We conducted experiments on financial datasets obtained from the UCI ML Repository, including banks from Germany, Taiwan, Japan, and Australia [12–15]. These datasets contain a single binary outcome variable, making them suitable for training binary classifiers. Table 2 provides an overview of the datasets, including the number of observations (rows) and features (columns). During the study, we divided the datasets into training and testing subsets.

Description of datasets

Table 2. Description of datasets

	# Sample	# Features
German	1,000	20
Taiwan	30,000	25
Japan	691	16
Australia	690	14

Parameters and timings for the HE scheme

Training a PPNN using HE encounters challenges in directly calculating the sigmoid function within the NN model. Therefore, we adopted a sigmoid polynomial approximation approach. For security settings, we utilized a polynomial degree of 8,192 with coefficient module bit sizes [40, 21, 21, 21, 21, 21, 21, 40].

Table 3 evaluates the model performance based on the average inference time for the above datasets. The inference time varies depending on the number of features and data type. Additionally, it provides the transaction size for both encrypted and unencrypted inference across all datasets. The transaction size increases based on the number of features, data type, and security settings.

Table 3. Dataset encryption time and encrypted training time

Dataset	ANN without privacy		ANN with privacy	
	Inference time	Transaction size	Inference time	Transaction size
German	100.600 µs	0.307 KB	111.889 ms	324.831 KB
Taiwan	110.841 µs	0.331 KB	138.915 ms	323.187 KB
Japan	105.523 µs	0.267 KB	116.523 ms	322.845 KB
Australia	103.779 µs	0.259 KB	114.827 ms	323.335 KB

Tables 4 and 5 compare the models generated using our PPANN approach with HE and the standard ANN without privacy for the German, Taiwan, Japan, and Australia datasets. To assess the effectiveness of the models, we computed accuracy (%) and area under the curve (AUC) values while varying the learning rate with a fixed number of epochs set to five hundred.

Table 4. Accuracy comparison of ANN with privacy and without privacy by varying learning rate

Data set	Learning rate											
	ANN with privacy						ANN without privacy					
	0.01	0.02	0.03	0.1	0.2	0.3	0.01	0.02	0.03	0.1	0.2	0.3
German	77%	77%	79%	79%	79%	77%	77%	77%	79%	79%	79%	77%
Taiwan	63%	61%	61%	62%	61%	60%	63%	61%	61%	62%	61%	59%
Japan	83%	85%	85%	83%	83%	83%	83%	85%	85%	83%	83%	83%
Australia	87%	87%	88%	86%	86%	87%	87%	87%	87%	86%	86%	87%

Table 5. Comparison of ANN with privacy and without privacy of AUC with varying learning rate

Data set	Learning rate											
	ANN with privacy						ANN without privacy					
	0.01	0.02	0.03	0.1	0.2	0.3	0.01	0.02	0.03	0.1	0.2	0.3
German	0.76	0.76	0.78	0.78	0.78	0.77	0.82	0.81	0.82	0.82	0.83	0.83
Taiwan	0.63	0.61	0.62	0.62	0.61	0.60	0.70	0.68	0.67	0.64	0.64	0.63
Japan	0.83	0.85	0.85	0.83	0.82	0.83	0.91	0.91	0.90	0.88	0.88	0.88
Australia	0.88	0.87	0.87	0.86	0.86	0.87	0.94	0.94	0.94	0.90	0.90	0.89

Further, Figures 7 and 8 depict comparison of accuracy and AUC for proposed model with and without privacy respectively. To evaluate the effectiveness of the models, we computed the accuracy (%) metric by varying the number of epochs.

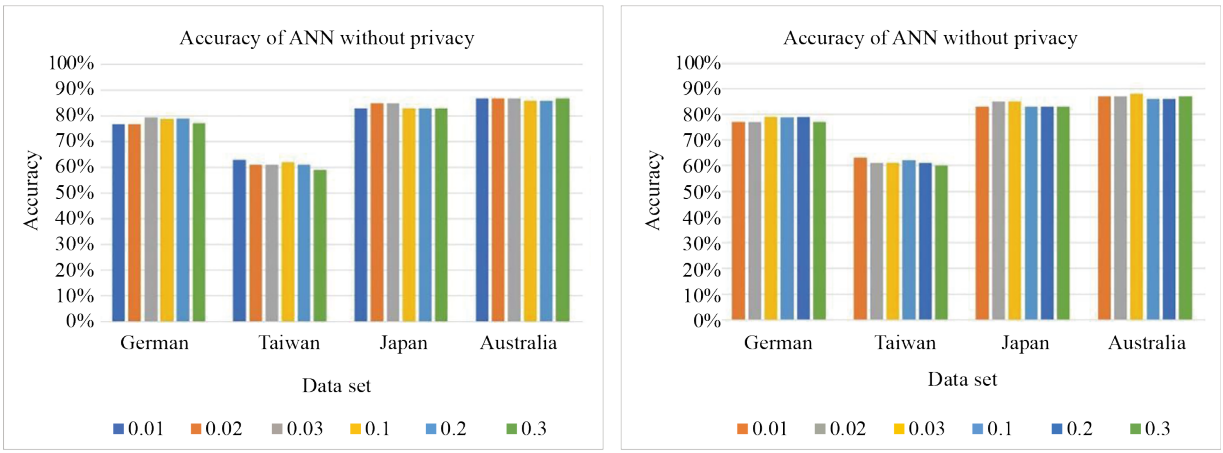

Figure 7. Accuracy comparison of ANN by varying learning rate

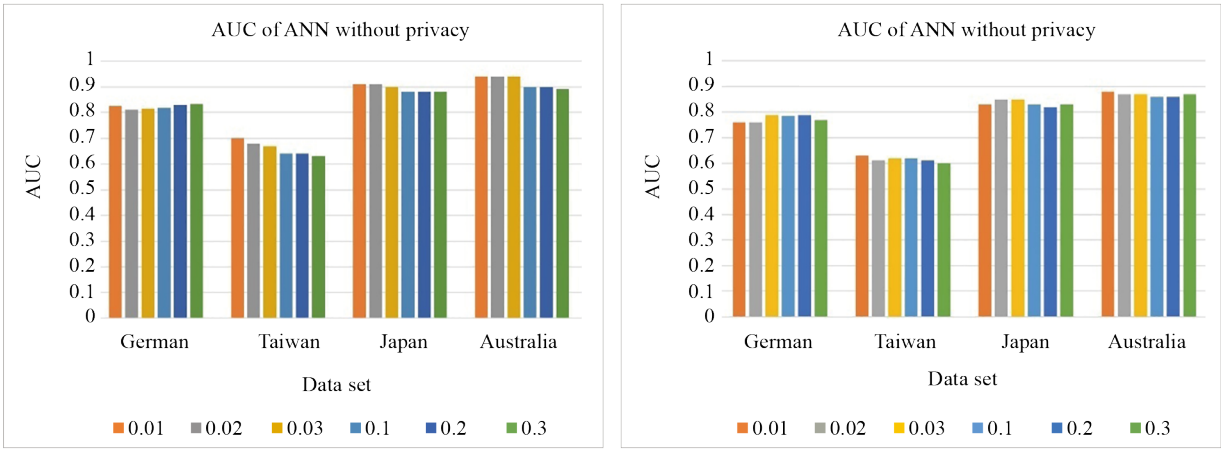

Figure 8. Comparison of AUC with privacy and without privacy using ANN

The comparative analysis of the accuracy and AUC with and without privacy using HE are shown in Figure 9. The experiment was conducted by varying 500 epochs.

Table 6. State-of-the-art comparison: Privacy-preserving techniques and key insights

Study	Approach	Privacy technique	Applications	Key findings/limitations
Zheng et al. [11]	PCAL framework.	Adversarial learning.	Credit risk modeling.	Balances privacy and predictive utility but requires further dataset validation.
Maniar et al. [12]	Differential privacy in CRA.	Differential privacy.	Loan default prediction.	Accuracy decreases as privacy improves; focus on customer data protection.
Andolfo et al. [13]	Functional encryption for credit scoring.	Functional encryption.	Financial data processing.	High computational overhead; hardware reliance on Intel SGX.
Lin et al. [14]	Privacy-preserving credit scoring.	Non-interactive Zero-Knowledge proofs.	Banking applications.	Challenges in scalability and potential for privacy leaks in deployment.
Divakar et al. [15]	Privacy-preserving credit scoring.	Homomorphic encryption.	Privacy-preserving credit scoring.	Comparable accuracy to non-HE models; slightly lower computational efficiency.
Nugent et al. [16]	Fraud detection on encrypted transactions.	Homomorphic encryption.	Fraud detection.	Low latency but storage and complexity issues during deployment.
Song et al. [21]	ReActHE neural network.	Homomorphic encryption.	Biomedical predictions.	Effective privacy but requires advanced NN optimizations for scalability.
Proposed work	HE-integrated NN for CRA.	Homomorphic encryption.	Multi-national financial datasets.	Maintains privacy with minimal accuracy loss; computational overhead remains significant.

The results show that ANN with privacy has a deviation from ANN without privacy of around 1%, 1%, 0.5%, and 1%; in some instances, ANN with privacy outperforms ANN without privacy in accuracy and AUC. Additionally, the

inference time is one thousand times faster in ANN without privacy over ANN with privacy. Further, the transaction size of the ANN without privacy is one hundred times lesser size when compared to ANN with privacy as it includes polynomial encryption that increases the size of the transaction.

The proposed PPCRA framework demonstrates notable advancements in balancing data privacy and predictive accuracy for credit risk analysis. When compared with state-of-the-art approaches, it addresses gaps in privacy protection while achieving competitive performance. Future work can explore hybrid approaches, such as integrating Federated Learning with HE, to improve scalability and computational efficiency. Our implementation revealed that using ANN with privacy attains accuracy and AUC like ANN without privacy (Table 6).

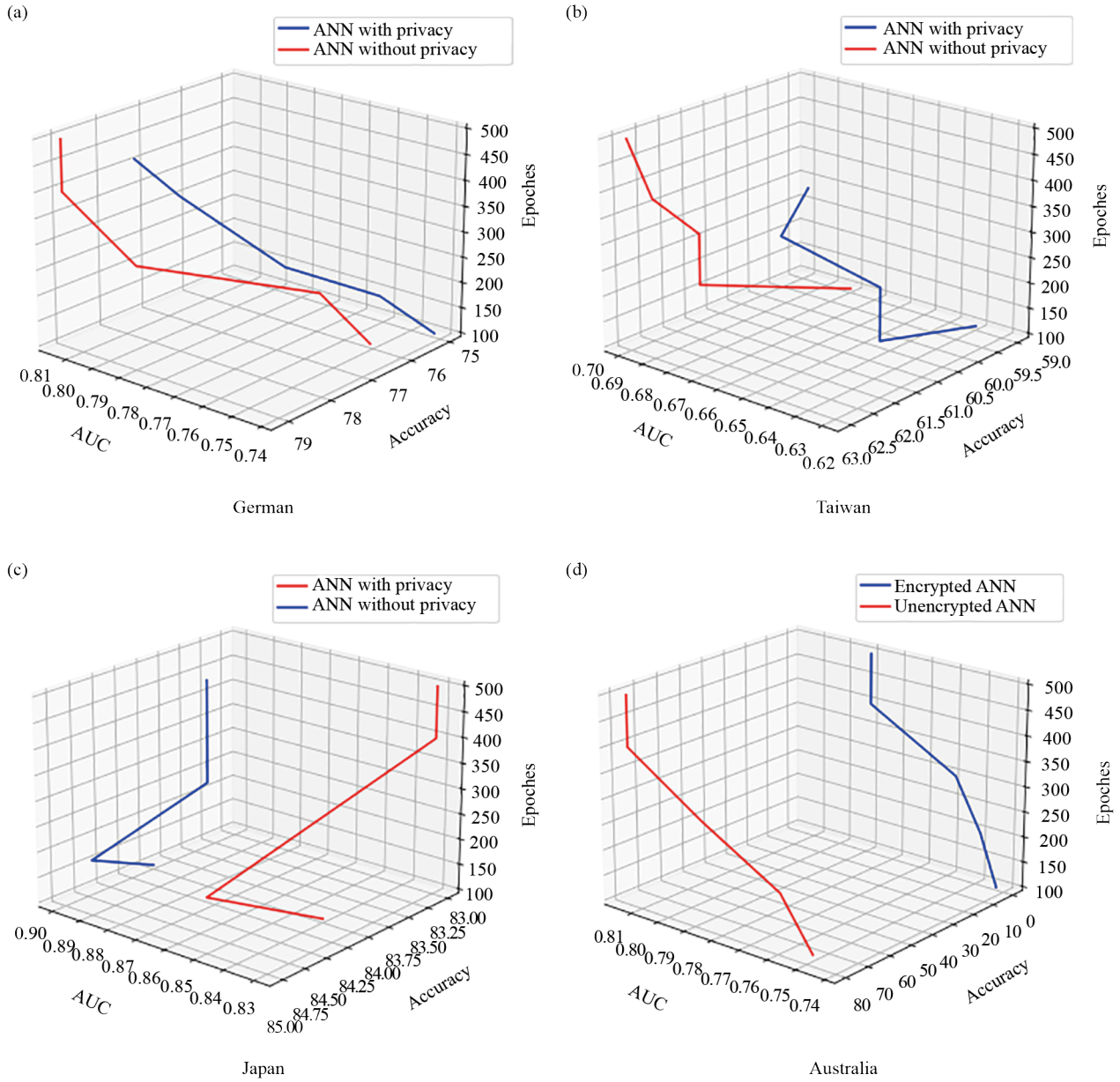

Figure 9. Average accuracy and AUC of unencrypted neural network and encrypted neural network

7. Conclusions and future research

In this paper, we present a novel framework for privacy-preserving credit risk analysis (PPCRA) using neural networks (NN) and homomorphic encryption. Our method tackles the urgent requirement for privacy protection in credit risk assessment while maintaining accuracy as well as efficiency in decision-making processes. The use of HE enables sensitive financial data to be encrypted and securely processed, ensuring that confidentiality is maintained throughout the analysis. A HE-enabled NN model capable of operating on encrypted data was developed. Further, an exhaustive security analysis showing resilience against privacy attacks and empirical validation using real-world financial datasets from multiple countries is part of our achievements. We conducted evaluations under both preservation of privacy measures and no such measures that gave insight into its effectiveness.

Experimental results showcased that our PPANN approach with HE achieved comparable accuracy and area under the curve (AUC) values to standard NN models without privacy preservation. The performance was slightly affected by privacy protecting model relative to its unsecured version only, which he subjected to several standard privacy attacks. Nevertheless, we discovered experimentally that this privacy preserving model could be implemented in a fully operational setting if time cost and HE transaction sizes are kept within tolerable bounds. This paper shows progress made in confidentiality conservation methods for CRA and it is greatly relevant to financial institutions and data protection experts. We have other future works on exploring additional private preservations approaches apart from scalability and computational efficiency for our proposed framework.

In the future, we aim to explore the integration of advanced techniques to address privacy challenges in credit risk analysis. Hybrid models hold significant promise, that combine the strengths of Homomorphic Encryption (HE), Federated Learning, and Differential Privacy. Multi-key homomorphic encryption (MKHE) will enable secure, collaborative data analysis, while blockchain technology offers a decentralized and transparent framework for safeguarding sensitive data in privacy-preserving credit risk analysis (PPCRA). Additionally, with the rise of quantum computing, post-quantum cryptographic solutions will become essential to ensure long-term security and resilience against quantum threats. These innovations aim to create robust, scalable, and privacy-preserving frameworks for credit risk analysis.

Authors contribution

V V L Divakar Allavarapu: Data Curation-Equal, Formal Analysis-Equal, Investigation-Equal, Methodology-Equal, Writing-original draft-Lead, Writing-review and editing-Equal, Vankamamidi S Naresh: Conceptualization-Lead, Formal Analysis-Lead, supervision-Lead, Investigation-Equal, Methodology-Equal, Writing-review and editing-Equal. Krishna Mohan Ankala: Supervision-Lead. All authors read and approved the final manuscript.

Availability of data

Data will be provided based on a request.

Research involving human and/or animals

This study did not involve any procedures or experiments with human participants or animals. All research was conducted without the inclusion of living subjects in accordance with ethical guidelines.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Bhattacharya A, Biswas SK, Mandal A. Credit risk evaluation: a comprehensive study. *Multimedia Tools and Applications*. 2023; 82(12): 18217-18267. Available from: https://doi.org/10.1007/s11042-022-13952-3.

[2] Vijaya K. Credit Credit risk prediction using ensemble machine learning algorithms. In: *2023 International Conference on Inventive Computation Technologies (ICICT)*. Lalitpur, Nepal: IEEE; 2023. p.41-47. Available from: https://doi.org/10.1109/ICICT57646.2023.10134486.

[3] Abdulsalam YS, Hedabou M. Security and privacy in cloud computing: technical review. *Future Internet*. 2022; 14(1): 11. Available from: https://doi.org/10.3390/fi14010011.

[4] Li J, Kuang X, Lin S, Ma X, Tang Y. Privacy preservation for machine learning training and classification based on homomorphic encryption schemes. *Information Sciences*. 2020; 526: 166-179. Available from: https://doi.org/10.1016/j.ins.2020.03.041.

[5] Ziru W. Credit risk prediction model of financial companies based on machine learning algorithm. In: 2023 *IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)*. Raichur, India: IEEE; 2023. p.1-6. Available from: https://doi.org/10.1109/ICICACS57338.2023.10099908.

[6] Gide AI, Mu'azu AA. A Real-time intrusion detection system for DoS/DDoS attack classification in IoT networks using knn-neural network hybrid technique. *Babylonian Journal of Internet of Things*. 2024; 2024: 60-69. Available from: https://doi.org/10.58496/BJIoT/2024/008.

[7] Ayad J. Survey on neural networks in networking: applications and advancements. *Babylonian Journal of Networking*. 2024; 2024: 135-147. Available from: https://doi.org/10.58496/BJN/2024/014.

[8] Mijwel MM, Esen A, Shamil A. Overview of neural networks. *Babylonian Journal of Machine Learning*. 2023; 2023: 42-45. Available from: https://doi.org/10.58496/BJML/2023/008.

[9] Li J, Xu C, Feng B, Zhao H. Credit risk prediction model for listed companies based on CNN-LSTM and attention mechanism. *Electronics*. 2023; 12(7): 1643. Available from: https://doi.org/10.3390/electronics12071643.

[10] Balakrishnan C, Thiagarajan M. Credit risk modelling for indian debt securities using machine learning. *Buletin Ekonomi Moneter Dan Perbankan*. 2021; 24: 107-128. Available from: https://doi.org/10.21098/bemp.v24i0.1401.

[11] Zheng Y, Wu Z, Yuan Y, Chen T, Wang Z. PCAL: A Privacy-preserving intelligent credit risk modeling framework based on adversarial learning. *arXiv:2010.02529*. 2020. Available from: https://doi.org/10.48550/arXiv.2010.02529.

[12] Maniar T, Akkinepally A, Sharma A. Differential privacy for credit risk model. *arXiv:2106.15343*. 2021. Available from: https://doi.org/10.48550/arXiv.2106.15343.

[13] Andolfo L, Coppolino L, D'Antonio S, Mazzeo G, Romano L, Ficke M, et al. Privacy-preserving credit scoring via functional encryption. In: *Computational Science and Its Applications-ICCSA 2021: 21st International Conference*. Cagliari, Italy; 2021. p.31-43.

[14] Lin C, Luo M, Huang X, Choo K-KR, He D. An efficient privacy-preserving credit score system based on noninteractive zero-knowledge proof. *IEEE Systems Journal*. 2022; 16(1): 1592-1601. Available from: https://doi.org/10.1109/JSYST.2020.3045076.

[15] Divakar Allavarpu VV, Naresh VS, Mohan AK. Privacy-preserving credit risk analysis based on homomorphic encryption aware logistic regression in the cloud. *Security and Privacy*. 2024; 7(3): e372. Available from: https://doi.org/10.1002/spy2.372.

[16] Nugent D. Privacy-preserving credit card fraud detection using homomorphic encryption. *arXiv:2211.06675*. 2022. Available from: https://doi.org/10.48550/arXiv.2211.06675.

[17] Xiao X, Wu T, Chen Y, Fan X. Privacy-preserved approximate classification based on homomorphic encryption. *Mathematical and Computational Applications*. 2019; 24(4): 92.

[18] Cheon JH, Kim D, Kim Y, Song Y. Ensemble method for privacy-preserving logistic regression based on homomorphic encryption. *IEEE Access*. 2018; 6: 46938-46948.

[19] Amorim I, Maia E, Barbosa P, Praça I. Data privacy with homomorphic encryption in neural networks training and inference. In: *International Symposium on Distributed Computing and Artificial Intelligence*. Cham, Switzerland: Springer Nature Switzerland; 2023. p.365-374.

[20] Wingarz T, Gomez-Barrero M, Busch C, Fischer M. Privacy-preserving convolutional neural networks using homomorphic encryption. In: *2022 International Workshop on Biometrics and Forensics (IWBF)*. Salzburg, Austria: IEEE; 2022. p.1-6. Available from: https://doi.org/10.1109/IWBF55382.2022.9794535.

[21] Song C, Shi X. ReActHE: A homomorphic encryption friendly deep neural network for privacy-preserving biomedical prediction. *Smart Health*. 2024; 32: 100469.

[22] Gentry C. Fully homomorphic encryption using ideal lattices. In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. Bethesda, MD, USA; 2009. p.169-178.

[23] Jestine P, Meenatchi S, Muthu S, Annamalai W, Ming W, Ahmad A, et al. Privacy-preserving collective learning with homomorphic encryption. *IEEE Access*. 2021; 9: 132084-132096.

[24] Mohammad M, Kimiagari S, Marriappan V. Credit risk classification: an integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research*. 2021; 330(1): 609-637. Available from: https://doi.org/10.1007/s10479-021-04114-z.

[25] Emmanuel I, Yanxia S, Zenghui W. A machine learning-based credit risk prediction engine system using a stacked classifier and a filter-based feature selection method. *Journal of Big Data*. 2024; 11(1): 23. Available from: https://doi.org/10.1186/s40537-024-00882-0.

[26] Yong S, Yida Q, Zhensong C, Yunlong M, Yunong W. Improved credit risk prediction based on an integrated graph representation learning approach with graph transformation. *European Journal of Operational Research*. 2023; 315(2): 786-801. Available from: https://doi.org/10.1016/j.ejor.2023.12.028.

[27] Yung-Joo S, Yuyan W, Xin Y, Russell Z, Chuanren L. Loan default prediction using a credit rating-specific and multi-objective ensemble learning scheme. *Information Sciences*. 2023; 629: 599-617. Available from: https://doi.org/10.1016/j.ins.2023.02.014.

[28] Ezgi Z, Selma AÖ. Privacy preserving classification over differentially private data. *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery*. 2021; 11(3): e1399. Available from: https://doi.org/10.1002/widm.1399.

[29] Qiao Y, Lan Q, Zhou Z, Ma C. Privacy-preserving credit evaluation system based on blockchain. *Expert Systems With Applications*. 2022; 188: 115989. Available from: https://doi.org/10.1016/j.eswa.2021.115989.

[30] Cheon JH, Kim A, Kim M, Song Y. Homomorphic encryption for arithmetic of approximate numbers. In: *Cryptology-ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security*. Hong Kong, China: Springer International Publishing; 2017. p.409-437.