Privacy-Preserving Credit Card Approval Using Homomorphic SVM: Toward Secure Inference in FinTech Applications

Faneela[1], Baraq Ghaleb[1], Jawad Ahmad[2], William J. Buchanan[1], Sana Ullah Jan[1]

[1] Blockpass ID Lab, Edinburgh Napier University, Edinburgh.
[2] College of Business Administration, Prince Mohammad Bin Fahd University, Al Khobar, Saudi Arabia.

Abstract. The growing use of machine learning in cloud environments raises critical concerns about data security and privacy, especially in finance. Fully Homomorphic Encryption (FHE) offers a solution by enabling computations on encrypted data, but its high computational cost limits practicality. In this paper, we propose PP-FinTech, a privacy-preserving scheme for financial applications that employs a CKKS-based encrypted soft-margin SVM, enhanced with a hybrid kernel for modeling non-linear patterns and an adaptive thresholding mechanism for robust encrypted classification. Experiments on the Credit Card Approval dataset demonstrate comparable performance to the plaintext models, highlighting PP-FinTech's ability to balance privacy, and efficiency in secure financial ML systems.

Keywords: Homomorphic Encryption, CKKS, Support Vector Machine, Encrypted Inference, Privacy-preserving

1 Introduction

The rise of cloud computing has accelerated machine learning adoption in sensitive sectors like finance and healthcare. While offering scalability and efficiency, outsourcing ML tasks to the cloud raises critical concerns about data privacy and result integrity [1]. Indeed, numerous ML-based financial systems have emerged, with Support Vector Machines (SVMs) gaining popularity due to their effectiveness in classification tasks [2], but their reliance on external computation risks exposing sensitive data and receiving unverified outputs. To address these challenges, we propose PP-FinTech, a privacy-preserving SVM scheme leveraging the CKKS fully homomorphic encryption scheme, which supports real-valued encrypted computation [3]. Our approach addresses both data confidentiality and computational correctness, two key challenges in secure financial ML systems [4]. The key contributions of this paper are:

– A CKKS-encrypted SVM with a hybrid Polynomial–RBF kernel for modeling non-linear data.

– An adaptive thresholding method to improve encrypted classification robustness under noise.
– An optimized inference pipeline using SIMD (Single Instruction, Multiple Data) techniques, allowing multiple encrypted samples to be processed in parallel and reducing per-sample latency.
– A careful selection of encryption parameters to stay within the noise budget for reliable decryption.
– Experimental evaluation on the Credit Card Approval dataset showing promising performance while ensuring strong privacy guarantees.

2 Preliminaries

2.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm widely used for classification tasks, particularly effective for linearly separable problems. Its primary objective is to identify the optimal hyperplane that maximizes the margin between different classes in the training dataset. For cases involving nonlinear data distributions, SVM leverages kernel functions—such as polynomial and Gaussian (RBF) kernels—to implicitly map the input data into a higher-dimensional feature space, where linear separation becomes feasible [5]. The classification decision function of an SVM can be expressed as:

$$\hat{y} = \text{sign}\left(\sum_{i \in S} \alpha_i y_i K(x_i, t) + b\right)$$

Here, S denotes the set of indices corresponding to the support vectors, while x_i and y_i represent the support vectors and their associated labels, determined during the training phase. The coefficient α_i is the Lagrange multiplier for each support vector, t is the input test sample, and b is the bias term. The function $K(x_i, t)$ is the kernel function that measures the similarity between the training sample x_i and the test sample t. The output $\hat{y}$ represents the predicted class label.

2.2 CKKS Scheme

The CKKS scheme [6] is a leveled fully homomorphic encryption framework based on the Ring Learning With Errors (RLWE) problem, designed to support approximate arithmetic on encrypted real-valued data [7]. It operates with key parameters such as the ring dimension n, multiplicative depth L, modulus chain $q_L \geq \ldots \geq q_1$, and scaling factor Δ, which together control the precision and noise growth during encrypted computations. CKKS also supports SIMD encoding, allowing multiple values to be packed into a single ciphertext. This property enables efficient batch processing during inference, which we leverage in our encrypted SVM pipeline. Further details about parameter settings and noise budget control are described in the secure inference section.

In our implementation, we employed the OpenFHE library to perform all homomorphic operations using the CKKS scheme. OpenFHE is an open-source fully homomorphic encryption library that provides implementations of state-of-the-art FHE schemes [8].

3 Related Work

Encrypted inference using SVMs has been the subject of increasing attention in privacy-preserving ML. For instance, Park et al. [9] proposed an SVM framework built on Homomorphic Encryption (HE) that incorporates fairness constraints during training. Deng et al. [10] introduced a soft-margin SVM model tailored for secure disease diagnosis. Other studies, including [11] and [12], have developed privacy-preserving diagnosis systems and verifiable SVM classification protocols across multiple clouds using HE. While these approaches demonstrate secure inference capabilities, they either operate under partially encrypted settings or lack support for practical functionalities such as batch processing, noise tolerance, and encrypted post-processing Beyond SVMs, HE has also been employed in other machine learning models, such as logistic regression [13] and deep neural networks [14, 15], particularly in healthcare and credit scoring contexts. However, these works primarily focus on linear or deep learning-based models, without exploring encrypted non-linear classification using SVMs. In contrast to prior work, our approach introduces a fully encrypted inference pipeline using a hybrid-kernel SVM implemented under the CKKS scheme. We incorporate adaptive thresholding to mitigate the effects of encryption-induced noise and leverage SIMD batching to enhance runtime performance. This work bridges a key gap between cryptographic privacy guarantees and practical, real-time encrypted inference, particularly in financial application domains.

4 Proposed Scheme

We propose PP-FinTech, a privacy-preserving SVM classification framework based on the CKKS FHE scheme. The approach consists of two phases: plaintext model training and encrypted inference. During training, an SVM is trained on an 80:20 split of the Credit Card Approval dataset, producing support vectors, dual coefficients, and bias. These parameters are encrypted using CKKS to enable secure inference over encrypted input data. Predictions remain encrypted during computation and are decrypted only by the user, ensuring complete data privacy. The proposed model is detailed in the following subsections:

4.1 Model Training and Feature Selection

In the training phase, a standard SVM model is trained on plaintext data using the Credit Card Approval dataset, which is which underwent multiple preprocessing and feature selection stages. To ensure the robustness and effectiveness of

the proposed SVM model, we conducted a thorough preprocessing pipeline, encompassing normalization and feature selection techniques tailored for financial datasets. The dataset employed for training and evaluation is the Credit Card Approval dataset, obtained from the UCI ML Repository. This dataset comprises 690 instances and 15 attributes, a mix of both numerical and categorical features such as age, income, employment status, and credit history indicators. Due to its real-world origin and widespread use in benchmarking financial decision systems, it serves as an ideal candidate for evaluating the performance of privacy-preserving classification models. Before training the model, all numerical attributes are normalized using the `StandardScaler` method, a widely accepted technique for standardizing features by removing the mean and scaling to unit variance. This step is essential for mitigating issues arising from feature dominance and ensuring numerical stability, particularly when using distance-based classifiers such as SVMs [16]. The transformation is mathematically defined as:

$$X' = \frac{X - \mu}{\sigma}$$

where X denotes the original feature value, μ is the mean, and σ is the standard deviation. By centering and scaling the input features, we ensure that each contributes equally to the model's decision boundary, preventing any bias caused by differing value ranges.

Following normalization, a filter-based feature selection approach is employed to reduce the dimensionality of the dataset and improve computational efficiency during encrypted inference. Filter methods assess the relevance of each feature independently based on statistical measures such as correlation with the target class. This process aids in eliminating redundant or irrelevant features, which can otherwise introduce noise and degrade model performance especially in HE settings, where computational overhead is sensitive to input size. The selection process identified a subset of features that retained strong predictive power while significantly reducing the dimensionality of the input space.

Hybrid Kernel Approach To improve classification accuracy and generalization, we adopt a hybrid kernel combining Polynomial and Radial Basis Function (RBF) kernels. The polynomial kernel captures high-order feature interactions, while the RBF kernel models local, non-linear patterns by projecting inputs into an infinite-dimensional space, allowing the model to form complex and flexible decision boundaries even with limited training data [17]. This combination leverages the strengths of both kernels, enabling the model to handle diverse data structures without substantial computational overhead. The hybrid kernel function is mathematically defined as:

$$K(X', SV_j) = \lambda_1 K_p(X', SV_j) + \lambda_2 K_R(X', SV_j)$$

where λ_1 and λ_2 are weight factors controlling the contribution of each kernel. In our implementation, the weight parameters were tuned using a grid search approach on a validation set. The best empirical results were obtained with λ_1

= 0.7 and $\lambda_2 = 0.3$. This configuration achieved an optimal balance between classification accuracy, model interpretability, and generalization performance. Additionally, we adopt a soft-margin SVM, which introduces slack variables to balance margin maximization with allowable misclassifications. This flexibility is essential for noisy or overlapping classes, as often seen in real-world financial data where perfect separability is rarely possible.

Conservative Scaling for Error Recovery To ensure stable training, we apply a conservative scaling strategy [18] that adjusts model updates based on gradient stability. The scaling factor is dynamically set as:

$$\beta = \begin{cases} 1, & \text{if update is stable} \\ 0.1, & \text{if instability occurs} \\ 0.01, & \text{for further correction} \end{cases}$$

This factor moderates updates to the dual coefficients and bias:

$$\alpha_i \leftarrow \alpha_i + \beta \cdot \Delta\alpha_i, \quad b \leftarrow b + \beta \cdot \Delta b$$

Here, $\Delta\alpha_i$ and Δb denote the calculated updates for the dual coefficients and the bias term, respectively. This mechanism is used only during the training phase and helps stabilize convergence when large margin shifts or noisy samples are detected.

Input: Normalized test sample X', support vectors $\{SV_j\}_{j=1}^{k}$, polynomial kernel K_p, RBF kernel K_r, kernel weights λ_1, λ_2, dual coefficients α_j, scaling factor β
Output: Encrypted decision score D
Initialize decision score: $D \leftarrow 0$;
for *each support vector* SV_j **do**
  Evaluate polynomial kernel: $K_p^j \leftarrow K_p(X', SV_j)$;
  Evaluate RBF kernel: $K_r^j \leftarrow K_r(X', SV_j)$;
  Compute hybrid kernel output: $K^j \leftarrow \lambda_1 \cdot K_p^j + \lambda_2 \cdot K_r^j$;
  Apply conservative scaling: $\tilde{\alpha}_j \leftarrow \alpha_j + \beta \cdot \Delta\alpha_j$;
  Update decision score: $D \leftarrow D + K^j \cdot \tilde{\alpha}_j$;
end
return D;
Algorithm 1: Hybrid Kernel Evaluation with Conservative Scaling

4.2 Encrypted Inference

After training, the PP-FinTech model performs encrypted inference, classifying test samples without ever decrypting them. We use the CKKS scheme with a 128-bit security level (HEStd_128_classic), a ring dimension of 32,768, a modulus degree of 16,384, and the scaling factor Δ of 2^{20}. A three-level modulus chain supports the multiplicative depth needed for kernel evaluation while balancing precision, performance, and noise growth. To manage noise from ciphertext multiplications, rescaling is applied after each operation. Parameter settings were selected to ensure computations remain within the noise budget, preventing decryption errors. The inference phase consists of four steps: data encryption, homomorphic kernel evaluation, adaptive thresholding, and result decryption.

Data Encryption and HE Kernel Evaluation In the data encryption phase, data samples and model parameters are encrypted. For a given data sample X', each feature is individually encrypted using

$$C(X'_i) = \text{Enc}(X'_i)$$

where $C(X'_i)$ is the encrypted representation of the feature.The encrypted test sample is then represented as a vector of ciphertexts:

$$C(X') = \{C(X'_1), C(X'_2), ..., C(X'_n)\}$$

Encrypted samples are processed using homomorphic operations to evaluate the SVM decision function without decryption. The encrypted decision score is computed as:

$$C(S') = \sum_j C(\alpha_j) \cdot C(K(X', SV_j)) + C(b)$$

The hybrid kernel output for each support vector is computed as:

$$C(K(X', SV_j)) = \text{Enc}(K(X', SV_j))$$

SIMD-Based Encrypted Matrix Multiplication To boost efficiency, we leverage SIMD techniques for parallel computation within a single homomorphic operation [19]. Each test sample is packed into one ciphertext, enabling homomorphic dot products:

$$C(D_j) = \sum_{i=1}^{n} C(X'_i) \cdot C(SV_{j,i})$$

Where $C(D_j)$ represents the encrypted dot product between the test sample (X'_i) and the corresponding support vector. For multiple test samples, this extends naturally to matrix form for batch processing.:

$$C(D) = C(X') \cdot C(SV)^T$$

which can be expanded as:

$$C(D) = \begin{bmatrix} \sum_{i=1}^{n} C(X'_{1,i}) \cdot C(SV_{1,i}) & \cdots & \sum_{i=1}^{n} C(X'_{1,i}) \cdot C(SV_{k,i}) \\ \sum_{i=1}^{n} C(X'_{2,i}) \cdot C(SV_{1,i}) & \cdots & \sum_{i=1}^{n} C(X'_{2,i}) \cdot C(SV_{k,i}) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} C(X'_{m,i}) \cdot C(SV_{1,i}) & \cdots & \sum_{i=1}^{n} C(X'_{m,i}) \cdot C(SV_{k,i}) \end{bmatrix}$$

To improve performance, SIMD batching was applied during encrypted matrix multiplications, enabling parallel inference over multiple samples within a single ciphertext. This optimization reduced per-sample latency and memory usage. While we did not benchmark SIMD separately, qualitative observations confirmed reduced total inference time, particularly when processing batches of test samples.

Adaptive Threshold for Secure Classification Our model leverages adaptive thresholding which dynamically adjusts the decision boundary to improve classification results, and is computed as:

$$\theta = \lambda_1 \cdot \mu + \frac{\lambda_2}{\sigma}$$

where μ represents the mean value of encrypted decision scores across multiple test samples, and σ outlines the variations in encrypted classification outputs. The weights $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$ were determined empirically and fixed to maintain consistent classification behavior across encrypted batches. This approach is, to our knowledge, among the first to embed adaptive thresholding within a CKKS-encrypted SVM pipeline, offering improved classification stability under encrypted inference conditions.

Input: Encrypted test sample $\text{Enc}(X')$, encrypted support vectors $\text{Enc}(SV_j)$, dual coefficients α_j, hybrid kernel weights λ_1, λ_2, model bias b
Output: Predicted class label (in encrypted form or after decryption)
Initialize encrypted decision score: $\text{Enc}(D) \leftarrow 0$;
for *each support vector* SV_j **do**
  Compute encrypted polynomial kernel:
    $\text{Enc}(K_p^j) \leftarrow K_p(\text{Enc}(X'), \text{Enc}(SV_j))$;
  Compute encrypted RBF kernel:
    $\text{Enc}(K_r^j) \leftarrow K_r(\text{Enc}(X'), \text{Enc}(SV_j))$;
  Compute encrypted hybrid kernel:
    $\text{Enc}(K^j) \leftarrow \lambda_1 \cdot \text{Enc}(K_p^j) + \lambda_2 \cdot \text{Enc}(K_r^j)$;
  Multiply with plaintext coefficient:
    $\text{Enc}(S_j) \leftarrow \alpha_j \cdot \text{Enc}(K^j)$;
  Accumulate score:
    $\text{Enc}(D) \leftarrow \text{Enc}(D) + \text{Enc}(S_j)$;
end
Add encrypted bias:
  $\text{Enc}(D) \leftarrow \text{Enc}(D) + \text{Enc}(b)$;
Decrypt decision score:
  $D \leftarrow \text{Dec}(\text{Enc}(D))$;
Compute adaptive threshold:
  $\theta = \lambda_1 \cdot \mu + \lambda_2/\sigma$;
return $\text{sign}(D - \theta)$;

Algorithm 2: Encrypted Inference with Adaptive Thresholding

Secure Decryption and Classification After encrypted inference, the classification results are sent to the client for decryption using their private key:

$$S' = \text{Dec}(C(S'))$$

The classification label is then determined with adaptive thresholds:

$$y' = \begin{cases} 1, & \text{if } S' > \theta \\ 0, & \text{otherwise} \end{cases}$$

5 Results and Discussion

In this section, we evaluate the performance of four models:

- **PT-Linear**: A baseline plaintext model that uses a standard linear SVM kernel. This model serves as a reference point for performance without encryption or advanced kernel techniques..
- **PP-Linear**: An encrypted version of the linear SVM model, implemented using the CKKS homomorphic encryption scheme which performs inference directly on encrypted data.

– **PT-FinTech**: A plaintext hybrid-kernel model that combines Polynomial and Radial Basis Function (RBF) kernels to capture more complex, nonlinear relationships in the data.
– **PP-FinTech**: An encrypted hybrid-kernel SVM model combining Polynomial and RBF kernels using CKKS.

Fig. 1 compares only two models — PT-Fintech and PP-Fintech — to highlight the effect of encryption on the performance of the hybrid-kernel SVM. We assessed the models using four key metrics: accuracy, precision, recall, and F1-score, defined by Equations (1)–(4):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{4}$$

Here, TP and TN denote correctly predicted positive and negative instances, while FP and FN represent misclassified negatives and positives, respectively [20]. Fig. 1 illustrates a comparative analysis of accuracy, precision, recall, and F1-score between our proposed PP-FinTech and the baseline model (PT-FinTech). This evaluation aims to assess the impact of integrating CKKS-based FHE into our model. The results show that PP-FinTech achieved a precision of 98.33%, slightly higher than PT-FinTech's 96.82%. Conversely, PT-FinTech attained a marginally higher accuracy of 97.51%, compared to 97.06% for PP-FinTech. Recall values were 96.58% for PT-FinTech and 95.16% for PP-FinTech. The F1-score was nearly identical across both models, with 96.74% for PP-FinTech and 96.70% for PT-FinTech. While minor differences are observed, they are not statistically significant and fall within the expected margin of variation. These findings confirm that the encrypted PP-FinTech model delivers comparable predictive performance to its non-encrypted counterpart, despite the computational noise introduced by FHE. The slightly higher precision of PP-FinTech may reflect improved robustness in positive classifications, though such differences should be interpreted cautiously. Overall, the results justify the use of FHE in privacy-preserving inference without compromising model effectiveness. A detailed comparison across all evaluation metrics is presented in Table 1.

5.1 ROC Curve

In addition to standard evaluation metrics, we examined the Receiver Operating Characteristic (ROC) curve to further evaluate model classification performance. The ROC curve plots the True Positive Rate (TPR) against the False Positive

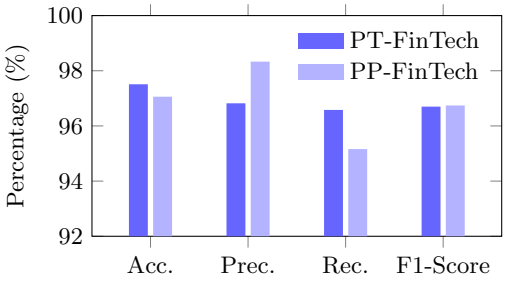


Fig. 1: Performance Comparison Between PT-Fintech and PP-FinTech

Rate (FPR) across varying classification thresholds, providing insight into the model's ability to distinguish between classes [21] [22]. A key summary metric is the Area Under the Curve (AUC), where higher values indicate better discriminatory power [23]. Fig. 2 shows both models exhibit strong performance indicating high TPR and low FPR across thresholds. Notably, the encrypted PP-FinTech model maintains a performance nearly identical to its plaintext counterpart, suggesting that encryption introduces minimal degradation in classification effectiveness.

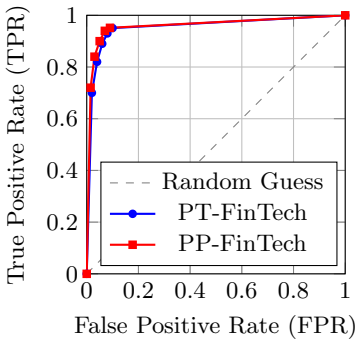


Fig. 2: ROC Curve Comparison

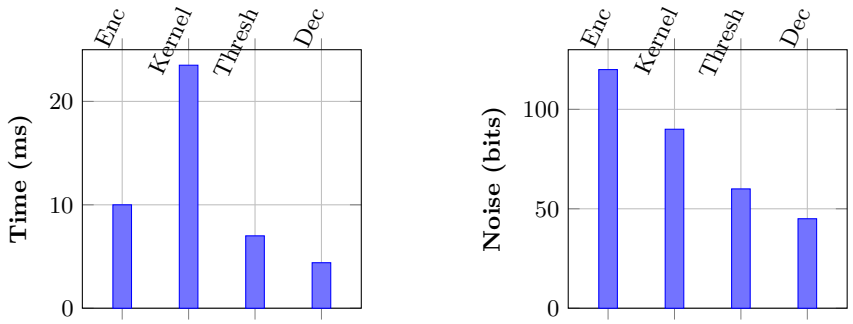
5.2 Computational Overhead

We also evaluated computation time across all four models to measure efficiency. As shown in Table 1, encryption adds noticeable latency, with PP-FinTech showing higher inference time than PT-FinTech due to the overhead of FHE. However, this is mitigated by SIMD batching and optimized CKKS parameters. Despite the added cost, PP-FinTech maintains practical latency, making it viable for privacy-sensitive financial applications with moderate throughput needs.

Table 1: Baseline Comparison of Linear and Hybrid SVM Models

Model	Encrypted	Acc. (%)	Pre. (%)	Rec. (%)	F1-Score (%)	Time (ms)
PT-Linear	No	90.53	88.26	89.03	88.58	0.34
PP-Linear	Yes	89.04	87.05	88.01	87.48	1.75
PT-FinTech	No	97.51	96.82	96.58	96.70	2.91
PP-FinTech	Yes	97.06	98.33	95.16	96.74	44.9

Runtime Breakdown To analyze the performance of the encrypted inference pipeline, we profiled each stage to identify its contribution to the total latency. As shown in Fig. 3a, hybrid kernel evaluation was the most time-consuming step, averaging 23.5 ms per sample. Encryption and adaptive thresholding took 10 ms and 7 ms respectively, while decryption was relatively fast at 4.4 ms. This results in a total average inference time of 44.9 ms per test sample, demonstrating the efficiency of our pipeline under FHE constraints.



(a) Runtime Breakdown (b) Noise Budget Analysis

Fig. 3: Empirical analysis of runtime and noise behavior during encrypted inference.

5.3 Noise Budget Analysis

During encrypted inference, each homomorphic operation in CKKS contributes to cumulative noise growth. To visualize this effect, Fig. 3b presents the remaining noise budget across the main stages of our inference pipeline: initial encryption, hybrid kernel evaluation, adaptive thresholding, and the final output before decryption. The noise budget begins at approximately 120 bits and decreases with each multiplication and rescaling step, reaching around 45 bits before decryption. This confirms that our chosen CKKS parameters provide sufficient multiplicative depth while keeping the noise within safe limits, ensuring accurate decryption at the end of inference.

5.4 Scalability Evaluation

To assess the scalability of our encrypted inference system, we simulated batch inference using the observed per-sample latency of approximately 44.9 ms. As shown in Fig. 4, the total inference time increases linearly with the number of test samples, confirming predictable scalability under larger workloads. This stable per-sample latency is attributed to the efficiency of SIMD batching in our encrypted pipeline, which enables parallel processing of multiple encrypted operations.

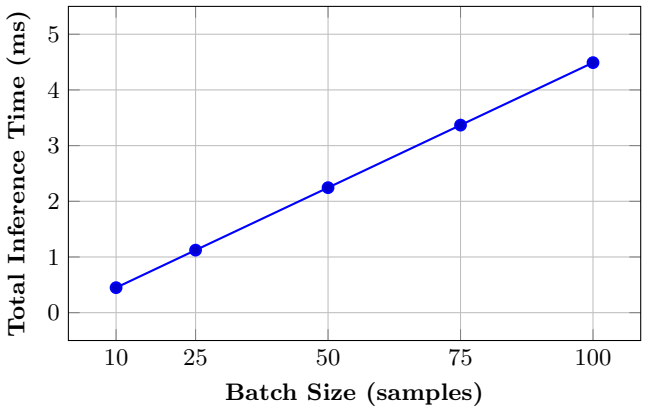


Fig. 4: Simulated scalability: Total inference time across different batch sizes for the PP-FinTech model.

6 Conclusion

This paper presented a privacy-preserving classification model that integrates SVM with the CKKS homomorphic encryption scheme for secure credit scoring. By employing a hybrid kernel and adaptive thresholding, the model effectively handles non-linear patterns while enhancing robustness. Despite the added encryption overhead, it delivers performance comparable to its plaintext counterpart with realistic latency, making it practical for privacy-sensitive financial applications. Future work will explore extensions to deep learning and real-time datasets to further improve scalability and efficiency.

References

1. S. Sahu, R. Ganeshan, and V. Muneeswaran, "Homomorphic encryption enabled svm for preserving privacy of p2p communication," in *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE, 2024, pp. 1–6.

2. C. Jana, S. Banerjee, S. Maur, and S. Dalai, "Mathematical morphology based sensing of power system disturbances using pca aided support vector machine," *IEEE Sensors Journal*, 2024.
3. J. Lee, P. N. Duong, and H. Lee, "Configurable encryption and decryption architectures for ckks-based homomorphic encryption," *Sensors*, vol. 23, no. 17, p. 7389, 2023.
4. C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. Fitzek, and N. Aaraj, "Survey on fully homomorphic encryption, theory, and applications," *Proceedings of the IEEE*, vol. 110, no. 10, pp. 1572–1609, 2022.
5. Y. Wu, X. Sun, Y. Zhang, X. Zhong, and L. Cheng, "A power transformer fault diagnosis method-based hybrid improved seagull optimization algorithm and support vector machine," *Ieee Access*, vol. 10, pp. 17 268–17 286, 2021.
6. J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in cryptology–ASIACRYPT 2017: 23rd international conference on the theory and applications of cryptology and information security, Hong kong, China, December 3-7, 2017, proceedings, part i 23*. Springer, 2017, pp. 409–437.
7. A. Kim, A. Papadimitriou, and Y. Polyakov, "Approximate homomorphic encryption with reduced approximation error," in *Cryptographers' Track at the RSA Conference*. Springer, 2022, pp. 120–144.
8. A. A. Badawi, A. Alexandru, J. Bates1, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio1, C. Pascoe, Y. Polyakov, S. R. Ian Quah, K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca, "OpenFHE: Open-source fully homomorphic encryption library," Cryptology ePrint Archive, Paper 2022/915, 2022, https://eprint.iacr.org/2022/915. [Online]. Available: https://eprint.iacr.org/2022/915
9. S. Park, J. Byun, and J. Lee, "Privacy-preserving fair learning of support vector machine with homomorphic encryption," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3572–3583.
10. G. Deng, M. Tang, Y. Xi, and M. Zhang, "Privacy-preserving online medical prediagnosis training model based on soft-margin svm," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 2072–2084, 2022.
11. B. Xie, T. Xiang, X. Liao, and J. Wu, "Achieving privacy-preserving online diagnosis with outsourced svm in internet of medical things environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 4113–4126, 2021.
12. C. Hu, C. Zhang, D. Lei, T. Wu, X. Liu, and L. Zhu, "Achieving privacy-preserving and verifiable support vector machine training in the cloud," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3476–3491, 2023.
13. V. S. Naresh and S. Reddi, "Exploring the future of privacy-preserving heart disease prediction: a fully homomorphic encryption-driven logistic regression approach," *Journal of Big Data*, vol. 12, no. 1, p. 52, 2025.
14. V. S. Naresh, "Ppdnn-crp: privacy-preserving deep neural network processing for credit risk prediction in cloud: a homomorphic encryption-based approach," *Journal of Cloud Computing*, vol. 13, no. 1, p. 149, 2024.
15. U. Sirisha and B. S. Chandana, "Privacy preserving image encryption with optimal deep transfer learning based accident severity classification model," *Sensors*, vol. 23, no. 1, p. 519, 2023.

16. M. Razavi, S. Ziyadidegan, R. Jahromi, S. Kazeminasab, V. Janfaza, A. Mahmoudzadeh, E. Baharlouei, and F. Sasangohar, "Machine learning, deep learning and data preprocessing techniques for detection, prediction, and monitoring of stress and stress-related mental disorders: a scoping review," *arXiv preprint arXiv:2308.04616*, 2023.
17. C. B. Pande, N. Kushwaha, I. R. Orimoloye, R. Kumar, H. G. Abdo, A. D. Tolche, and A. Elbeltagi, "Comparative assessment of improved svm method under different kernel functions for predicting multi-scale drought index," *Water Resources Management*, vol. 37, no. 3, pp. 1367–1399, 2023.
18. P. Krishnamurthy and F. Khorrami, "Efficient non-conservative realization of dynamic scaling-based controllers via matrix pencils for uncertain nonlinear strict-feedback systems," *Systems & Control Letters*, vol. 169, p. 105393, 2022.
19. D. Mustafa, R. Alkhasawneh, F. Obeidat, and A. S. Shatnawi, "Mimd programs execution support on simd machines: a holistic survey," *IEEE Access*, 2024.
20. B. H. Reddy and P. Karthikeyan, "Classification of fire and smoke images using decision tree algorithm in comparison with logistic regression to measure accuracy, precision, recall, f-score," in *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. IEEE, 2022, pp. 1–5.
21. A. M. Carrington, D. G. Manuel, P. W. Fieguth, T. Ramsay, V. Osmani, B. Wernly, C. Bennett, S. Hawken, O. Magwood, Y. Sheikh *et al.*, "Deep roc analysis and auc as balanced average accuracy, for improved classifier selection, audit and explanation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 329–341, 2022.
22. J. Miao and W. Zhu, "Precision–recall curve (prc) classification trees," *Evolutionary intelligence*, vol. 15, no. 3, pp. 1545–1569, 2022.
23. C.-Y. Lee and W.-C. Lin, "Induction motor fault classification based on roc curve and t-sne," *Ieee Access*, vol. 9, pp. 56 330–56 343, 2021.