Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
Afyon Kocatepe University – Journal of Science and Engineering
https://dergipark.org.tr/tr/pub/akufemubid

Araştırma Makalesi / Research Article
e-ISSN: 2149-3367
DOI: https://doi.org/10.35414/akufemubid.1686923
AKÜ FEMÜBİD **26** (2026) 015101 (64-77)
AKU J. Sci. Eng. **26** (2026) 015101 (64-77)

Homomorphic Encryption in Finance: Training and Inference on Encrypted Data with Concrete ML

Makale Bilgisi / Article Info
Alındı/Received: 29.04.2025
Kabul/Accepted: 08.09.2026
Yayımlandı/Published: 03.02.2026

Finansta Homomorfik Şifreleme: Concrete ML ile Şifrelenmiş Veriler Üzerinde Eğitim ve Çıkarım

Ensar YILMAZ[1], Burak AKTÜRK[1], Didem CİVELEK[1], Tolga BÜYÜKTANIR[1,*], Kazım YILDIZ[2]

[1] *Agra Research Lab, Agra Fintech Software Solutions, İstanbul, Türkiye*
[2] *Computer Engineering Department, Faculty of Technology, Marmara University, İstanbul, Türkiye*

© 2026 The Authors | Creative Commons Attribution-Noncommercial 4.0 (CC BY-NC) International License

Abstract

Although machine learning is a significant technology that provides valuable insights into financial analysis procedures, privacy concerns restrict the real application of these technologies. This study presents a privacy-preserving machine learning framework based on Fully Homomorphic Encryption (FHE), enabling encrypted inference on sensitive financial data using models trained on open datasets. The system was implemented using the Concrete-ML library and evaluated with three well-known financial distress prediction models, which are Altman Z-Score, L-Score, and Zmijewski Score. The results show that encrypted models can achieve high predictive performance, with R^2 values ranging from 0.68 to 0.98 and Adjusted R^2 values up to 0.96. The Concrete ML XGBoost model yielded the highest performance with $R^2 = 0.98$ on the Zmijewski Score. While the plaintext models achieved MAE values as low as 2.64, the encrypted versions showed increased error due to quantization, with MAE values reaching up to 396.12 in some cases. A five-fold cross-validation was performed to ensure model generalizability, and the results remained similar across folds. The inference times for encrypted models were significantly longer (up to 845 seconds) compared to plaintext inference (under 2 seconds), highlighting the privacy-performance trade-off. Nevertheless, the findings suggest that the proposed approach offers a secure and effective solution for financial applications where compliance with data protection laws such as the General Data Protection Regulation (GDPR) and Law on the Protection of Personal Data (KVKK) is critical.

Keywords: *Fully Homomorphic Encryption; Encrypted Model Training; Financial Privacy; Concrete ML Framework; Financial Risk and Distress Prediction*

Öz

Makine öğrenimi, finansal analiz süreçlerine faydalı bilgiler sunan önemli bir teknoloji olmasına rağmen, bu teknolojilerin gerçek kullanımı veri gizliliği endişeleriyle sınırlıdır. Bu çalışma, açık veri kümeleri üzerinde eğitilmiş modeller aracılığıyla hassas finansal veriler üzerinde şifreli çıkarım yapılmasına olanak tanıyan, Tam Homomorfik Şifreleme tabanlı gizlilik koruyucu bir makine öğrenmesi çerçevesi sunmaktadır. Sistem, Concrete-ML kütüphanesi kullanılarak uygulanmış ve Altman Z-Score, L-Score ve Zmijewski Score olmak üzere üç yaygın finansal sıkıntı tahmin modeliyle değerlendirilmiştir. Elde edilen sonuçlar, şifreli modellerin yüksek tahmin performansı sergileyebildiğini göstermekte; R^2 değerleri 0.68 ile 0.98 arasında, Düzeltilmiş R^2 (Adjusted R^2) değerleri ise 0.96'ya kadar ulaşmaktadır. Concrete-ML XGBoost modeli, Zmijewski skoru üzerinde $R^2 = 0.98$ ile en yüksek performansı göstermiştir. Açık modellerde 2.64 gibi düşük MAE değerleri elde edilirken, şifreli modellerde kuantizasyon etkisiyle 396.12'ye kadar çıkan hata değerleri gözlemlenmiştir. Modelin genellenebilirliğini değerlendirmek amacıyla 5 katlı çapraz doğrulama uygulanmış ve sonuçlar katlar arasında tutarlı bulunmuştur. Şifreli çıkarım süreleri (845 saniyeye kadar) açık çıkarıma (2 saniyenin altında) kıyasla belirgin şekilde uzun sürmüş; bu durum, gizlilik-performans dengesi sorununu ortaya koymuştur. Bununla birlikte, elde edilen bulgular, önerilen yaklaşımın Genel Veri Koruma Yönetmeliği (GDPR) ve Kişisel Verileri Koruma Kanunu (KVKK) gibi veri koruma yasalarıyla uyumluluğun kritik olduğu finansal uygulamalarda güvenli ve etkili bir çözüm sunduğunu göstermektedir.

Anahtar Kelimeler: *Tam Homomorfik Şifreleme; Şifreli Model Eğitimi; Finansal Mahremiyet; Concrete ML Çerçevesi; Finansal Risk ve Zorluk Tahmini*

1. Introduction

Machine learning (ML) is intensively applied in many sectors such as healthcare, finance, education, public services, logistics, retail, and e-commerce, as well as cybersecurity (Int. Ref. 1). In the field of finance, ML is widely used in many areas such as risk analysis, fraud detection, algorithmic trading, portfolio optimization, customer behavior analysis, and personalized financial services (Pattnaik et al. 2025). With the increasing adoption of machine learning, data privacy, and data protection have become significant concerns. ML models typically require large amounts of data for training. Occasionally, this data can also come from various sources of data. While data providers want to preserve their data, model providers are also sensitive about protecting their models (Int. Ref. 2). In order to protect data privacy when predicting credit risk, Naresh and Ayyappa created a deep learning model that uses homomorphic encryption. The model achieved high accuracy (80.48%) while making inferences with

*Sorumlu Yazar/Corresponding Author: Tolga BÜYÜKTANIR e-posta/e-mail: tolgabuyuktanir@gmail.com

encrypted data and demonstrated resistance to privacy attacks. This model provides a feasible option for privacy-preserving credit scoring applications in a cloud environment (Naresh and Ayyappa 2024). Ahamed and colleagues propose a wavelet neural network-integrated with FHE to protect data privacy in predicting bank failures. The proposed model has been tested on four different financial datasets and showed similar or better performance compared to unencrypted models. This method ensures privacy in financial forecasts while maintaining the model's effectiveness (Ahamed and Vadlamani 2024). Lee and colleagues designed a privacy-preserving machine learning model using FHE on deep neural networks. In tests conducted on the CIFAR-10 dataset, the suggested model achieved an accuracy rate of 90.67%, achieving results very close to the unencrypted model. This technique demonstrates that deep learning models can be effectively applied to encrypted data (Lee et al. 2022). Kadhe and colleagues developed a privacy-preserving anomaly detection model using federated learning (FL) on vertically and horizontally divided data among financial institutions. The suggested PV4FAD system guarantees data privacy during both the training and inference phases by combining FHE, secure multi-party computation (SMPC), and differential privacy (DP) techniques. The model allows the processing of confidential financial data between banks and payment systems without sharing, and it handles anomaly detection with high accuracy rates (Int. Ref. 3). Basu et al. proposed a model that integrates differential privacy (DP) and federated learning (FL) techniques for financial text classification. This BERT and RoBERTa-based model perform natural language processing (NLP) tasks while preserving the privacy of private financial data. In tests conducted on the Financial Phrase Bank dataset, it has been demonstrated that the model exhibits a balanced performance between privacy and accuracy (Int. Ref. 4).

In this study, the problem definition and background are presented in the introduction section, followed by the original contributions and novelty of the work. Section 2 describes the materials and methods, which include the dataset, the homomorphic encryption-based applications, the developed methodology, and the prototype implementation. Section 3 presents the experimental setup, results, and findings. Finally, the conclusion of the study is provided in the last section.

1.1 Motivation

Accurately evaluating financial ratios, credit scores, and ranking individuals or institutions in financial matters requires analyzing historical financial data. This data is sensitive to personal information. They are also protected by the laws KVKK in Turkey and GDPR in the European Union (Evren 2023), CCPA in the United States (Bradley 2024). Financial institutions trust this sensitive data to make better decisions and improve their performance. However, due to privacy concerns and legal restrictions, sharing this data with external third parties is difficult, especially when internal expertise is limited. Additionally, individuals and institutions often want to understand their financial status before formally contacting financial organizations, but hesitate to share detailed financial data due to privacy concerns. This hesitation creates obstacles to transparent and efficient financial decisions. Machine learning and decision support systems require large amounts of relevant data to be effective. Current limitations on data sharing and privacy concerns create a significant challenge. Traditional data analysis methods that depend on openly sharing data are no longer suitable. Thus, there is a need to develop and use advanced techniques, such as privacy-preserving data analysis and federated learning. These techniques enable effective analysis and modeling without revealing or transferring sensitive personal data. Solving this issue will improve financial analysis, meet legal requirements, and maintain trust by protecting the privacy of individuals and institutions. Therefore, the requirement briefly involves extracting decision support information from limited and anonymized data.

1.2 Contributions of the paper

The primary advancements in the field of privacy-preserving machine learning in finance include:

- Using the Concrete ML framework, the study presents a completely homomorphic encryption-based method that allows for safe training and inference on encrypted financial data without giving up privacy.
- The paper shows similar accuracy (R2: 0.68–0.98) for encrypted and plaintext models when applied to three well-known financial risk models (Altman Z-Score, L-Score, and Zmijewski Score).
- A client-server architecture for encrypted inference is demonstrated in a prototype system, guaranteeing that private information is kept private during computation and transmission.
- The study examines the computational cost of FHE and provides information on how to balance inference time, model accuracy, and quantization precision (n_bits).
- The approach offers financial organizations that handle sensitive data and achievable solutions by addressing data privacy laws (such as GDPR and KVKK).

Table 1. Comparison of selected studies using homomorphic encryption in financial machine learning

Study	HE Type	MI Model(s)	Application Domain	Dataset Type	Accuracy	Contribution/ Limitation
Lee et al(2022)	FHE	Deep Neural Network	Credit Scoring	Proprietary	80.48%	Cloud-based privacy scoring
Kadhe et al (2023)	FHE+WNN	Wavelet NN	Bank Failure	4 Financial Sets	Similar to unencrypted	WNN integration, custom dataset
Naresh & Ayyappa (2024)	FHE	CNN(DNN)	General	CIFAR -10	90.67%	Deep learning on FHE
Ahamed & Vadlamani (2024)	FHE+FL+DP	Anomaly Detection	Financial systems	Horizontally split	High precision	Federated anomaly detection model
Faneela et al. (2025)	FHE	SVM (CKKS-based)	Credit Card Approval	Public (UCI dataset)	86.1%	Introduced PP-FinTech framework with hybrid kernel SVM
Our proposed approach	FHE(Concrete)	XGBoost, Random Forest	Financial Distress	Public (BIST)	R2= 0.68-0.98	Open-source, Concrete ML, public dataset

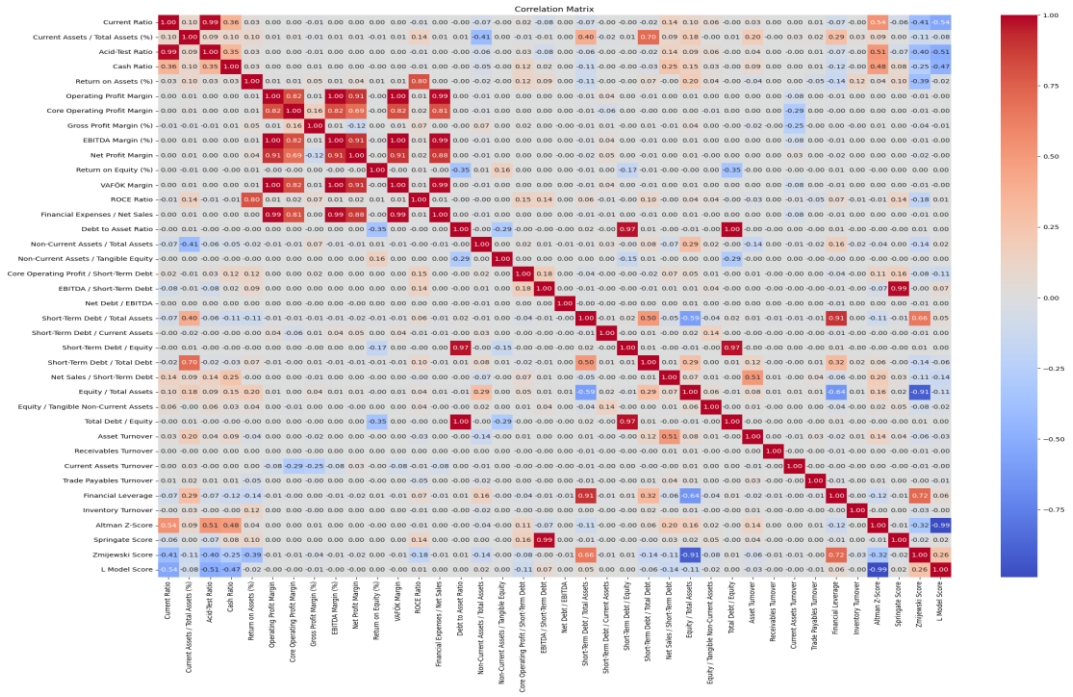


Figure 1. Correlation matrix of the financial ratios

1.2 Comparison with related work

A selection of current studies that use homomorphic encryption in machine learning applications that are sensitive to privacy or finance are summarized in Table 1. Our study is distinct from these in several ways, including the use of Concrete-ML, an open-source library designed especially for FHE inference; the application of models (XGBoost, RF) to actual public financial data (Istanbul stock exchange); and the demonstration of practical applicability with accuracy equivalent to simply models. Additionally, the solution enhances GDPR/KVKK compliance by not requiring data sharing with third-party servers.

2. Materials and Methods

2.1 Dataset

This study employs a dataset publicly available on Kaggle (Yilmaz et al 2025). The dataset consists of financial information and independent audit opinions for companies listed on Borsa Istanbul (BIST). The data were calculated from publicly disclosed annual financial reports and audit documents published via the Public Disclosure Platform (Kamuyu Aydinlatma Platformu - KAP) between 2008 and 2024. The dataset includes various financial ratios along with corresponding audit opinions, offering insight into the financial health, bankruptcy, and reporting credibility of Turkish public companies. This structured financial information serves as a foundation for evaluating financial distress and supports privacy-preserving machine learning applications within the proposed framework. The correlation matrix of the financial ratios in the dataset is displayed in Figure 1 and other detailed information and statistics are provided in (Yilmaz et al 2025).

2.2. Homomorphic Encryption

Homomorphic encryption is a new cryptographic technique that allows it to carry out calculations on

encrypted data. It was first introduced in 1978 under the name "Privacy Homomorphism" (Brickell and Yacobi 1988). However, the first FHE strategy, which enables computations on encrypted data, was demonstrated by Craig Gentry in 2009, based on a principle called "Bootstrapping" (Gentry 2009). By convention, both messages m1 and m2 and an encryption function Encrypt are given, and two functions f and f' are calculated if both of them follow conditions; they are called homomorphic, like equation 1:

$$f(m_1, m_2) = f'(Encrypt(m_1), Encrypt(m_2)) \quad (1)$$

During machine learning inference, Figure 2 illustrates a situation in which Homomorphic Encryption (HE) is utilized to protect the privacy of the data and the model.

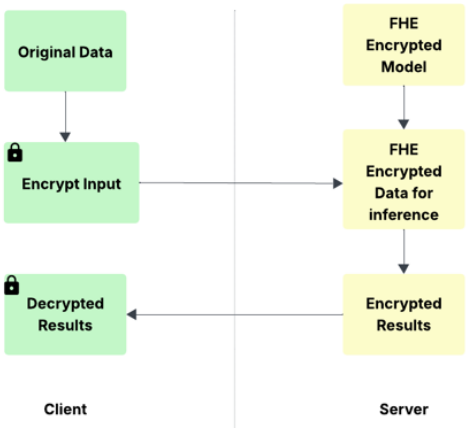


Figure 2. Homomorphic Encryption scenario

2.3. Concrete Machine Learning

Concrete ML (Frery et al 2023) is a homomorphic encryption-based open-source machine learning library that allows machine learning inference while keeping the privacy of data. Developed by Zama, this package enables model inference on encrypted data, especially through FHE techniques. Thereby, artificial intelligence applications can be created without the need to disclose sensitive data. Concrete ML utilizes basic machine learning methods such as decision trees and logistic regression and can perform the inference process entirely on encrypted data. In this regard, it offers a significant solution in areas requiring privacy-focused applications and compliance with regulations such as GDPR and KVKK (Zama 2023).

2.3.1 Random Forest

An ensemble learning method known as Random Forest was developed by combining the training of several decision trees. Within the framework of the Concrete Machine Learning approach, it is preferred to enhance the model's generalizability and provide resistance against noise in the dataset while working with concrete engineering data. Each decision tree creates different decision boundaries in the data space with the support of various subsampling strategies (bootstrap) and random feature selection. This also allows the model to naturally develop protection against overfitting (Liaw and Wiener 2002). Additionally, considering the privacy aspect of the study, the Random Forest algorithm was evaluated as a more suitable option for data security sensitivities because it can model using only sample statistics without directly needing raw data. This structure has been preferred not only for its high accuracy but also for its engineering stability and interpretability (Genuer et al. 2010).

2.3.2 XGBoost

XGBoost (Extreme Gradient Boosting) is used with an optimization-based learning approach, particularly aiming to enhance model performance in engineering problems with high complexity. The parametric control advantage of XGBoost supports more controlled modeling by reducing the risk of data leakage (Chen and Guestrin 2016). With the boosting approach, a new weak model is produced in each iteration by emphasizing the errors of the previous model, and this process cumulatively results in a stronger model (Friedman 2001). Additionally, XGBoost's regularization (L1-L2) strategies reduce overfitting by constraining model complexity, while also addressing privacy concerns (Yeom et al. 2018). Weighting data samples during training and learning according to their importance provides a significant advantage against the fact that engineering data is mostly imbalanced (Ke et al. 2017 and He and Garcia 2009))

2.4 Proposed Methodology

The proposed methodology, illustrated in Figure 3, is structured into two primary phases: the training phase and the inference phase. This two-stage design aims to facilitate privacy-preserving machine learning on sensitive financial data by leveraging FHE techniques within the Concrete ML framework developed by Zama.

The algorithm of the proposed methodology is presented in Algorithm 1. In the training phase, financial ratio data from various companies are collected in plaintext data format. These data are subjected to a series of preprocessing steps, including normalization, feature selection, and data transformation, to ensure consistency and improve model performance. Once preprocessing is complete, machine learning models are trained on the prepared dataset using the Concrete ML library. This library enables the training of models in a conventional, unencrypted environment, followed by the compilation

of the trained models into a format compatible with encrypted inference. The result is an FHE-ready model that can be deployed for secure, privacy-preserving predictions without requiring access to the raw input data

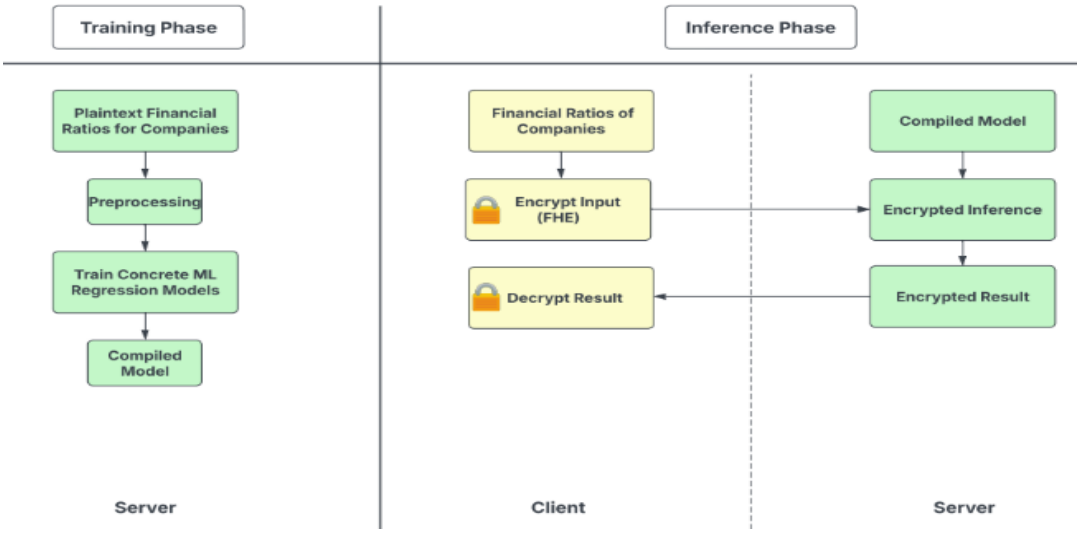


Figure 3. The proposed methodology.

Require: Financial ratio dataset X with labels y; new input sample x
Ensure: Predicted result $\hat{y}$
1: **// Training Phase (Offline)**
2: Preprocess dataset X: handle missing values, normalize features
3: Split X into training and test sets
4: Train machine learning model M on training data
5: Compile model M with Concrete ML to obtain FHE-compatible model M_{FHE}
6: Store M_{FHE} on server
7: **// Inference Phase (Client-Server with FHE)**
8: **Client:** Encrypt input $x \rightarrow x_{\text{enc}}$ using FHE scheme
9: **Client $\rightarrow$ Server:** Transmit x_{enc}
10: **Server:** Compute encrypted inference $y_{\text{enc}} \leftarrow M_{\text{FHE}}(x_{\text{enc}})$
11: **Server $\rightarrow$ Client:** Send y_{enc}
12: **Client:** Decrypt $y_{\text{enc}} \rightarrow \hat{y}$
13: **return** $\hat{y}$

Algorithm 1. Privacy-Preserving Financial Prediction Using FHE Algorithm for Proposed Methodology

The inference phase involves the application of the compiled models to new, unseen data in an encrypted setting. In this phase, the financial ratios of an individual company are first extracted and encrypted on the client side using an FHE scheme. This encryption ensures that the sensitive financial data remains confidential throughout the entire inference process. Simultaneously, the previously compiled machine learning models are hosted on a secure server. The encrypted input data is transmitted to the server, where the inference is carried out entirely in the encrypted domain, without any decryption at any stage. The server computes the prediction over the encrypted data and returns the encrypted output to the client. Upon receipt, the client decrypts the result locally using its private key.

The final output, once decrypted, is expected to be a close approximation of the prediction that would have been obtained using unencrypted (plaintext) inference. This process ensures that sensitive financial information remains protected throughout both data transmission and model execution, thereby meeting privacy requirements while still enabling meaningful predictive analytics.

Although other models, only Random Forest and XGBoost were accepted into the final comparison because of their superior performance on all measures (R^2, RMSE, and MAE). Additionally, these two models are specifically identified as being best suited for FHE based inference in the Concrete ML documentation.

Deep learning models are excluded due to technological constraints as well as performance issues unique to FHE settings. The flexibility and depth of neural network designs are severely limited by the fact that most existing FHE systems do not provide floating-point operations and

therefore model computations must be carried out using quantized integers. FHE-based systems have a substantial analyzing overhead, especially for deep learning scenarios, as the research has shown (Marcolla et al 2022 and Legiest et al 2023). Because of these limitations, our research focuses primarily at models that are empirically tested and natively supported in the Concrete ML framework.

2.5 Prototype Implementation

The prototype implementation is described in this section. To evaluate the feasibility and effectiveness of the proposed methodology, a prototype system was developed using the Concrete ML framework provided by Zama. This implementation demonstrates the complete workflow, including plaintext model training, compilation for encrypted inference, and privacy-preserving prediction on encrypted input data.

In the training phase, a dataset consisting of financial ratios extracted from publicly available company balance sheets was used. A detailed description of the dataset is provided in Subsection 2.1. Among the various machine learning models supported by Concrete ML, two algorithms—XGBoost and Random Forest—were selected for training in a plaintext environment. These algorithms were chosen due to their optimized implementation in Concrete ML and their proven success in similar tasks, as supported by findings in the literature.

After training, the models were compiled using Concrete ML's quantization and FHE-compatible compilation tools. This step converts the trained models into a form that supports secure execution over encrypted inputs. The compilation process relies on the Concrete-Numpy backend, which enables FHE. Quantization was applied automatically during compilation using various n_bits settings, ensuring that all model computations conform to the limitations of FHE-based arithmetic.

For inference, a client-server architecture was designed to reflect a realistic deployment scenario. On the client side, input financial ratios are encrypted using the FHE scheme provided by Concrete ML. These encrypted inputs are then transmitted to a server where the compiled model is deployed. Inference is performed on the encrypted data without any decryption occurring on the server. The encrypted output is sent back to the client, where it is decrypted locally to obtain the final prediction.

This prototype demonstrates the practical potential of using Concrete ML in privacy-sensitive financial decision-support systems. Performance indicators such as prediction accuracy (relative to plaintext inference), encryption and decryption times and overall inference latency were measured and are discussed in detail in this study

3. Results and Discussion

In this section, the performance evaluation of the prototype implementation is described in detail.

3.1 Experimental Setup

Every experiment was carried out in a Google Colab setting with an Intel(R) Xeon(R) CPU running at 2.20GHz, comprising 2 logical processors, a single physical core, and 13.6 GB of available memory. The experimental environment requirements are displayed in Table 2.

Table 2. Experimental environment specifications

Component	Specifications
Platform	Google Colab
Processor	Intel(R) Xeon(R) CP U @ 2.20GHz
Logical Processors	2
Physical Cores	1
Available Memory	13.6 GB

3.2 Performance Metrics

The performance of the models was examined using R^2 (R^2), which shows how well the model explains the data. Its value ranges between 0 and 1. As it gets closer to 1, it is accepted that it explains the data better. It is calculated as in equation 2 (James et al. 2013);

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} \left(y_i - \underline{y}_i\right)^2} \quad (2)$$

There, y_i as actual values, $\hat{y}_i$ values are predicted ones. $\underline{y}_i$ values are the means of the actual values and n is the number of predictions.

The Root Mean Squared Error (RMSE) indicates how much the predictions differ from the actual outcomes. It shows the magnitude of the model's prediction errors on the initial scale. Equation 3 shows the calculation of the RMSE value (Hyndman and Athanasopoulos 2018 and (Willmott and Matsuura 2005);

$$RMSE = \sqrt{\frac{1}{n}(y_i - \hat{y}_i)^2} \quad (3)$$

There, y_i as actual values, $\hat{y}_i$ values are predicted ones. n is the number of observations.

The average absolute difference between actual and anticipated values is measured using the simple and understandable regression metric known as mean absolute error. In this study, MAE is used as a complementary indicator to understand the overall prediction accuracy of FHE-compiled models without disproportionately penalizing larger errors. The metric is calculated as shown in Equation 4 (Shah,2025):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

A statistical measure called adjusted R^2 is used to determine a regression model's degree of fit while adjusting for the number of predictors. The equation for Adjusted R^2 is in 5:

$$Adjusted\ R^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$

Where p is the number of independent variables, N is the total sample size, and R^2 sample R^2 (Miles, 2005).

3.3 Model Comparison

Figures 4,5,6 and 7 each show the outcomes of experimental tests. Random Forest Regressor, XGBoost Regressor, and their privacy-preserving equivalents implemented via Concrete ML were applied to three prominent financial distress prediction models: L-Score, Altman Z-Score, and Zmijewski Score. Evaluation metrics include R^2, Mean Absolute Error (MAE), Adjusted R^2, and Root Mean Squared Error (RMSE). Concrete ML has other regression methods, but according to its documentation, Random Forest Regressor and XGBoost Regressor have been optimized.

The dataset was partitioned into three subsets to ensure robust model evaluation and prevent data leakage: 60% for training, 20% for validation, and 20% for testing. The training set was used to fit the model parameters, while the validation set was employed during the model development phase for hyperparameter tuning and performance monitoring. Importantly, the test set was held out entirely from the training and validation processes and used only for final performance evaluation. This splitting strategy ensures that model selection is based solely on the training and validation data, while the test results provide an unbiased estimate of generalization to unseen data.

Tablo 3. Comparison of Model Performance Metrics Based on Cross Validation(CV) Results

		RMSE			MAE			R-Squared			Adjusted R-Squared		
	CV	L-Score	Altman-Z	Zmijewski	L-Score	Altman-Z	Zmijewski	L-Score	Altman-Z	Zmijewski	L-Score	Altman-Z	Zmijewski
Random Forest	1	0,328	7,512	0,623	0,032	0,494	0,032	0,983	0,942	0,938	0,983	0,942	0,937
	2	0,116	3,541	0,107	0,025	0,364	0,020	0,999	0,996	0,998	0,999	0,996	0,998
	3	0,349	11,545	0,343	0,033	0,628	0,028	0,984	0,895	0,983	0,984	0,894	0,982
	4	1,544	29,136	0,202	0,059	0,919	0,023	0,891	0,730	0,992	0,891	0,728	0,992
	5	0,212	3,322	0,270	0,030	0,433	0,027	0,996	0,993	0,989	0,996	0,993	0,989
	Mean	0,510	11,011	0,309	0,036	0,568	0,026	0,971	0,911	0,980	0,971	0,911	0,980
	StDev	0,586	10,677	0,196	0,013	0,219	0,005	0,045	0,110	0,024	0,045	0,110	0,024
XGBoost	1	0,191	7,047	0,541	0,030	0,465	0,043	0,994	0,949	0,953	0,994	0,949	0,953
	2	0,430	8,002	0,101	0,032	0,496	0,035	0,992	0,981	0,998	0,992	0,981	0,998
	3	0,262	12,215	0,206	0,032	0,661	0,037	0,991	0,882	0,994	0,991	0,882	0,994
	4	0,902	16,981	0,128	0,044	0,726	0,036	0,963	0,908	0,997	0,963	0,908	0,997
	5	0,209	3,243	0,225	0,031	0,418	0,037	0,996	0,994	0,992	0,996	0,994	0,992
	Mean	0,399	9,498	0,240	0,034	0,553	0,038	0,987	0,943	0,987	0,987	0,942	0,987
	StDev	0,297	5,262	0,176	0,006	0,133	0,003	0,014	0,047	0,019	0,014	0,048	0,019
Concrete ML Random Forest	1	0,950	21,412	0,993	0,492	4,964	0,517	0,856	0,530	0,842	0,855	0,527	0,841
	2	0,689	17,797	0,837	0,456	4,483	0,619	0,980	0,905	0,859	0,980	0,904	0,858
	3	1,394	27,089	0,829	0,579	5,507	0,571	0,744	0,421	0,898	0,742	0,418	0,897
	4	0,875	26,296	0,891	0,468	5,200	0,641	0,965	0,780	0,836	0,965	0,779	0,835
	5	0,863	29,108	0,910	0,505	5,735	0,635	0,935	0,500	0,871	0,934	0,497	0,870
	Mean	0,954	24,340	0,892	0,500	5,178	0,597	0,896	0,627	0,861	0,895	0,625	0,860
	StDev	0,264	4,624	0,066	0,048	0,487	0,053	0,098	0,205	0,025	0,098	0,206	0,025
Concrete ML XGBoost	1	1,151	19,630	0,782	0,983	4,342	0,410	0,788	0,605	0,902	0,787	0,603	0,902
	2	1,173	14,242	0,481	0,994	4,146	0,366	0,941	0,939	0,954	0,941	0,939	0,953
	3	1,446	21,189	0,510	1,022	4,625	0,386	0,724	0,646	0,961	0,723	0,644	0,961
	4	1,234	24,664	0,546	1,005	4,864	0,410	0,931	0,807	0,939	0,930	0,805	0,938
	5	1,157	23,092	0,537	0,984	4,931	0,381	0,882	0,685	0,955	0,882	0,683	0,955
	Mean	1,232	20,563	0,571	0,998	4,582	0,391	0,853	0,736	0,942	0,852	0,735	0,942
	StDev	0,124	4,013	0,121	0,016	0,336	0,019	0,094	0,136	0,024	0,095	0,137	0,024

To evaluate the predictive performance and stability of each model, we conducted five-fold cross-validation and reported the mean and standard deviation (StDev) of RMSE, MAE, R-Squared, and Adjusted R-Squared across three financial distress indicators: L-Score, Altman-Z, and Zmijewski. The standard XGBoost model consistently outperformed all other models in terms of both accuracy and stability. It achieved the lowest average RMSE values (0.399 for L-Score, 9.498 for Altman-Z, and 0.240 for Zmijewski) with minimal variation across folds (StDev of 0.297, 5.262, and 0.176, respectively), indicating high reliability. The corresponding R-Squared and Adjusted R-Squared values exceeded 0.98 for all targets, confirming that XGBoost explained the vast majority of variance in each distress metric.

In comparison, the standard Random Forest model also produced competitive results but with slightly higher mean RMSEs and greater fold-to-fold variability, particularly for the Altman-Z score (StDev = 10.677). While both plain models delivered high predictive performance, the differentials in variance highlight XGBoost's superior consistency.

On the other hand, privacy-preserving models trained using Concrete ML (Random Forest and XGBoost variants)

showed significantly degraded performance. These models exhibited increased average RMSEs, often more than double those of their non-private counterparts, and reduced R-Squared scores, generally ranging from 0.85 to 0.90. Additionally, higher standard deviations across all metrics underscored their limited robustness under cross-validation. This performance degradation reflects the trade-off introduced by model quantization and encrypted inference constraints inherent to privacy-preserving computation.

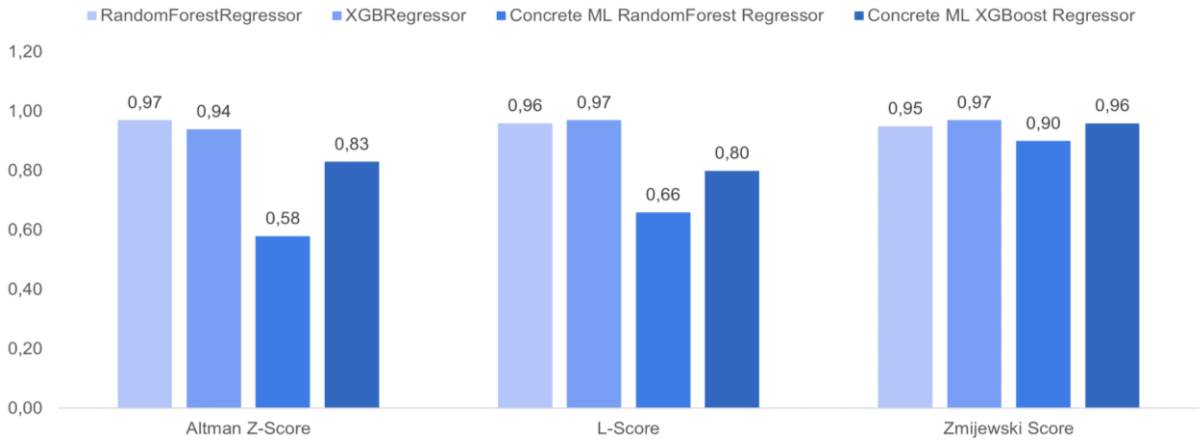


Figure 4. Comparison of R^2 Scores for Financial Distress Prediction Models (Altman Z-Score, L-Score, Zmijewski Score)

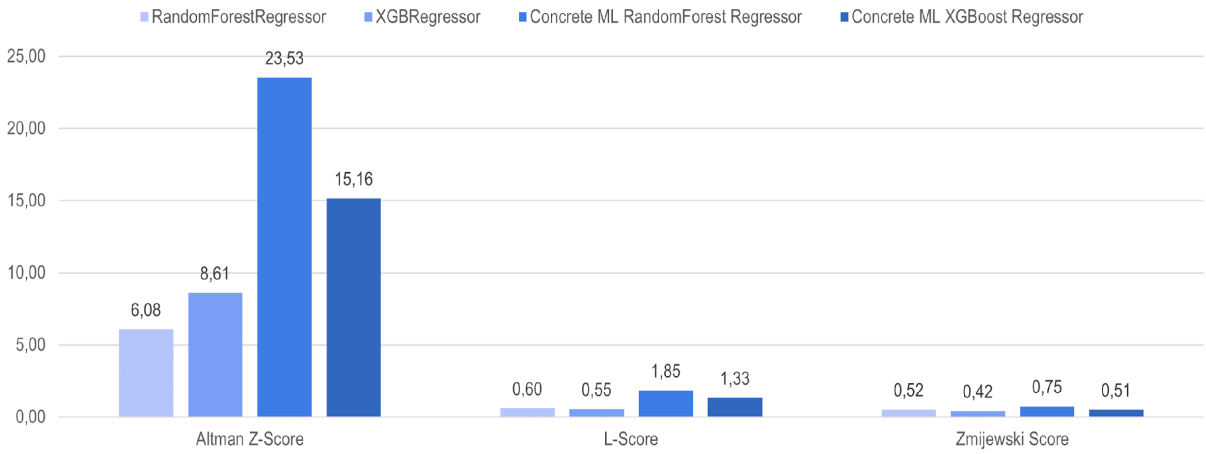


Figure 5. Comparison of RMSE Values for Financial Distress Prediction Models (Altman Z-Score, L-Score, Zmijewski Score)

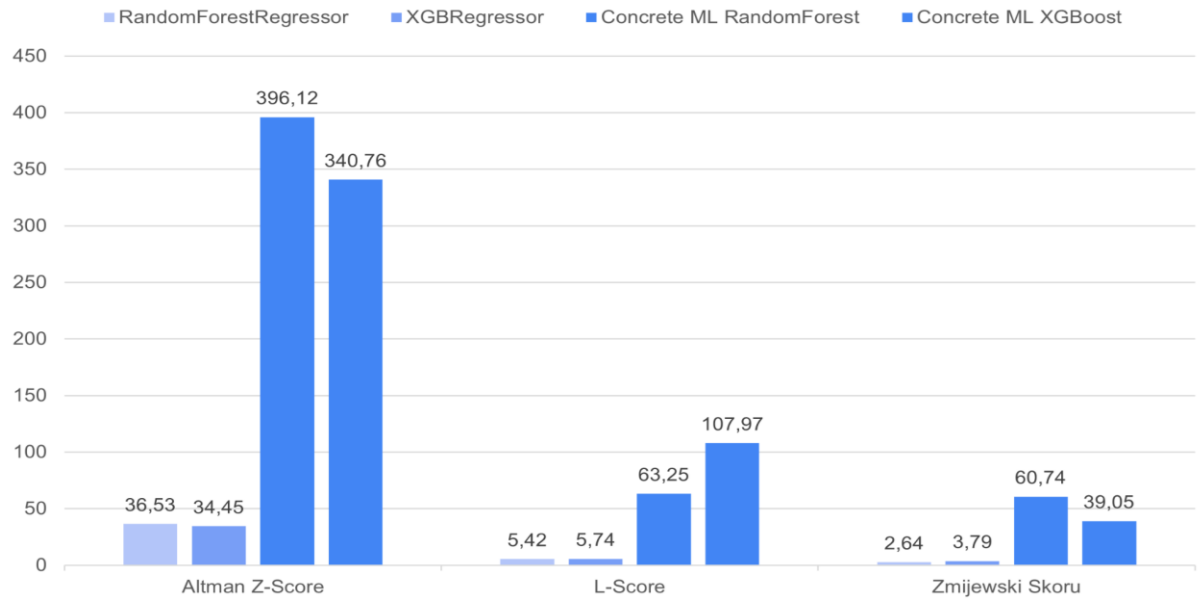


Figure 6. Comparison of MAE Values for Financial Distress Prediction Models (Altman Z-Score, L-Score, Zmijewski Score)

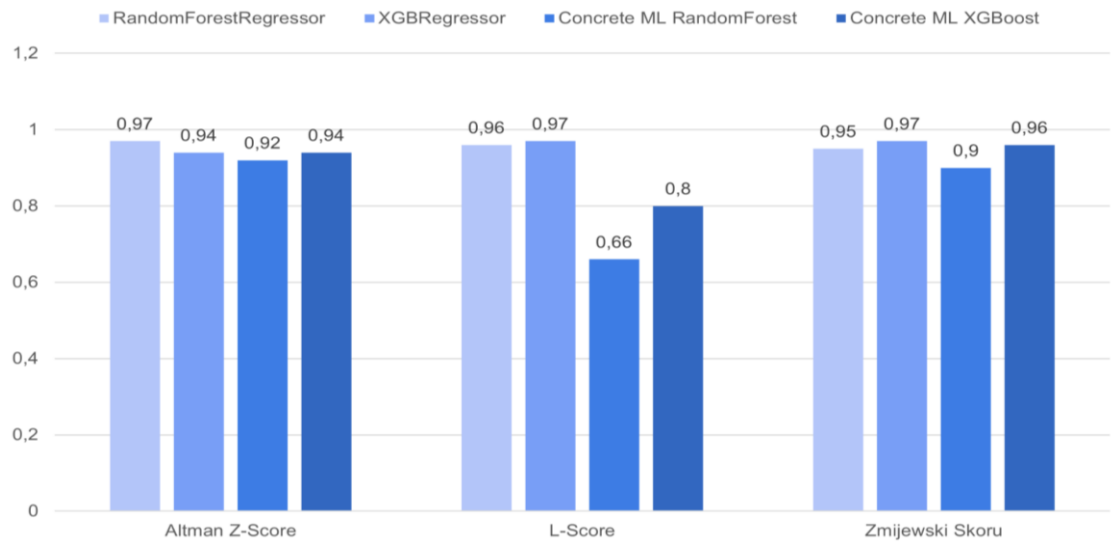


Figure 7. Comparison of Adjusted R^2 Scores for Financial Distress Prediction Models (Altman Z-Score, L-Score, Zmijewski Score)

For the L-Score, the XGBoost Regressor demonstrated the highest predictive accuracy with an R^2 value of 0.97, an RMSE of 0.55, and an MAE of 5.74. The Random Forest Regressor also performed well, achieving an R^2 of 0.96 and a slightly higher RMSE of 0.60, with an MAE of 5.42. In contrast, the privacy-preserving Concrete ML versions showed reduced performance. The Concrete ML Random Forest model achieved an R^2 of 0.66, with an RMSE of 1.85 and a significantly higher MAE of 63.25. The Concrete ML XGBoost variant improved slightly with an R^2 of 0.80 and an RMSE of 1.33, but exhibited a much larger MAE of 107.97.

In the case of the Altman Z-Score, the Random Forest Regressor again showed strong results with an R^2 of 0.97, an RMSE of 6.08, and an MAE of 36.53. The XGBoost Regressor followed with an R^2 of 0.94, an RMSE of 8.61, and an MAE of 34.45. The Concrete ML Random Forest version achieved an R^2 of 0.92 but suffered from a significantly higher RMSE of 23.53 and an MAE of 396.12. The Concrete ML XGBoost model maintained an R^2 of 0.94, while its RMSE rose to 15.16 and its MAE to 340.76.

For the Zmijewski Score, the XGBoost Regressor again attained the best results with an R^2 of 0.97, a low RMSE of 0.42, and an MAE of 3.79. The Random Forest Regressor followed closely, with an R^2 of 0.95, RMSE of 0.52, and MAE of 2.64. Among the Concrete ML models, the Random Forest version achieved an R^2 of 0.90, an RMSE of 0.75, and an MAE of 60.74. The Concrete ML XGBoost model performed slightly better, with an R^2 of 0.96, RMSE of 0.51, and MAE of 39.05.

To assess the risk of overfitting, we employed 5-fold cross-validation during model training. In this process, the training data was divided into five subsets. In each iteration, four subsets were used for training and one for validation, ensuring that each subset served as the validation fold once. The resulting cross-validation metrics were consistent across folds and closely aligned with those obtained from the held-out test set. This alignment demonstrates that the model generalizes well to unseen data and does not exhibit signs of overfitting. The strong agreement between cross-validation and test results confirms the robustness of the model's performance estimates and reinforces confidence in its predictive capability.

These findings indicate that while conventional models consistently provide superior predictive performance across all score types, the privacy-preserving Concrete ML implementations offer acceptable results. Although the encrypted models result in higher error margins, particularly in MAE, they still retain sufficient accuracy for practical applications, especially in domains where data confidentiality is a critical requirement.

Figure 8 presents the training time for each model, measured in seconds. For the Altman Z-Score, Concrete ML Random Forest Regressor and Concrete ML XGBoost Regressor required 25.77 seconds and 26.27 seconds, respectively, while the standard RandomForestRegressor and XGBRegressor completed training in 90.75 seconds and 15.57 seconds. For the L-Score, Concrete ML Random Forest Regressor and Concrete ML XGBoost Regressor took 25.08 and 26.63 seconds to train, while RandomForestRegressor completed in 103.0 seconds and XGBRegressor completed in 12.47 seconds. A similar pattern was observed for the Zmijewski Score, where Concrete ML Random Forest Regressor and Concrete ML XGBoost Regressor required 25.02 and 26.47 seconds, respectively, and the standard models XGBRegressor and RandomForestRegressor trained in 11.18 seconds and 69.10 seconds.

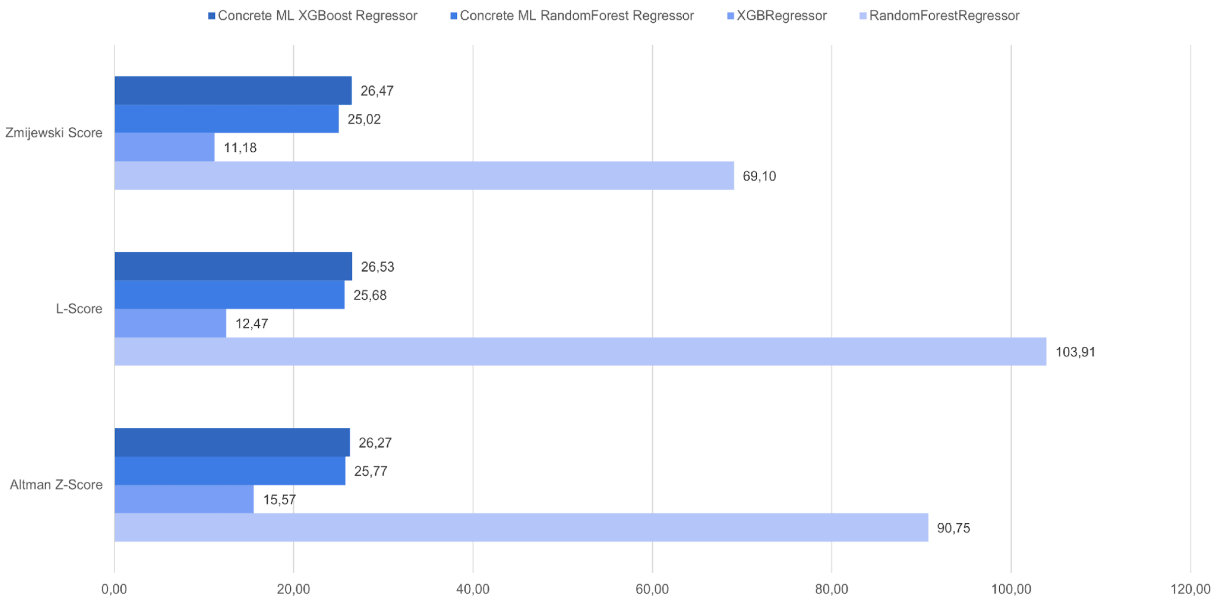


Figure 8. Training Time Comparison Across Models (Plaintext vs. Encrypted)

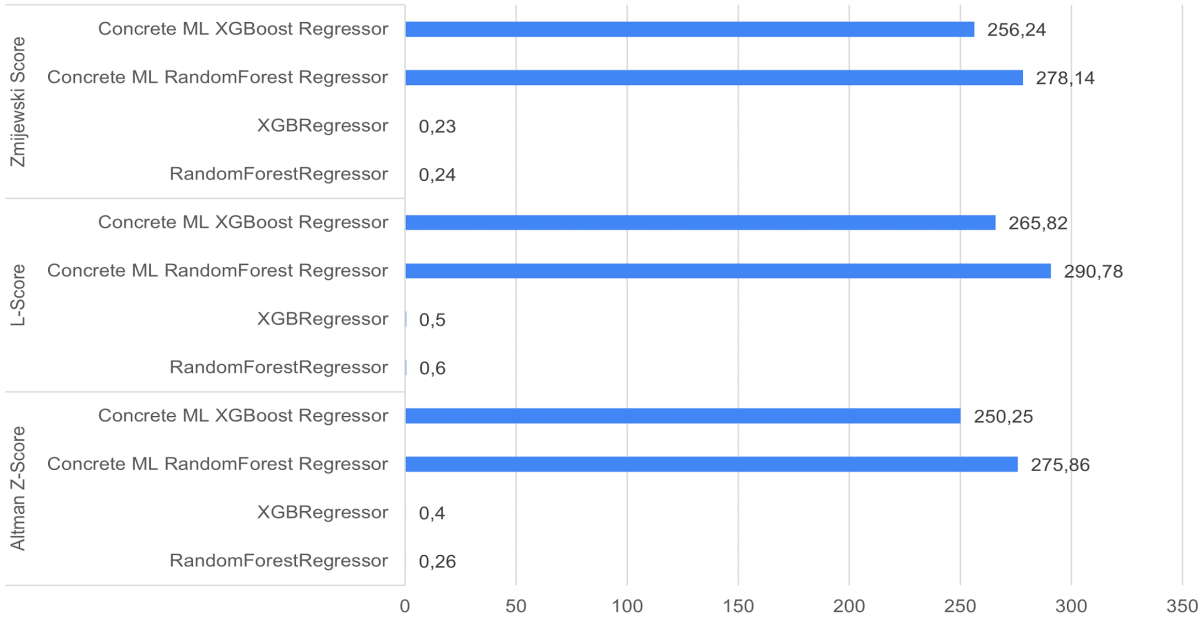


Figure 9. Inference Time Comparison Across Models (Plaintext vs. Encrypted)

Figure 9 presents the inference time in seconds. As expected, models running under encryption (Concrete ML) required more time to complete inference compared to their plaintext counterparts. For the Altman Z-Score, Concrete Random Forest and Concrete XGBoost had inference times of 275.86 seconds and 250.25 seconds, respectively, while their plaintext versions, RandomForestRegressor and XGBRegressor, achieved significantly faster times of 0.26 seconds and 0.40 seconds. Similar trends were observed for the L-Score and Zmijewski Score. These results confirm that encrypted inference introduces a notable computational overhead, which is expected in FHE-based systems.

The default parameters have been used during the tests. However, the Random Forest ve XGBoost Regressors have n_estimators 100. The optimum value of the Concrete ML Random Forest and XGBoost Regressor has been determined as n_estimator 20. While the max_depth parameter has been determined as unlimited in the Random Forest Regressor, it has been determined as 4 in the Concrete ML Random Forest Regressor. This parameter has been set to 3 in the Concrete ML XGBoost Regressor.

3.4 The Effect of the Concrete ML Data Quantization

To analyze the effect of quantization precision on model performance and computational cost, experiments were conducted by varying the n_bits parameter used during Concrete ML model compilation. Figure 10 shows how R-squared (R^2), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Adjusted R-squared (Adjusted R^2) metrics change with increasing n_bits. The R^2 and adjusted R^2 values increase consistently from 0.57 at 3

bits to 0.96 at 8 bits. This indicates that the model's ability to explain the variance in the data improves with higher bit precision. A significant improvement is observed between 5 and 6 bits, suggesting that 6 bits may represent a threshold for maintaining high model fidelity. The RMSE and MAE both decrease as n_bits increase. This trend shows that the prediction errors become smaller with greater bit-widths. Notably, RMSE drops from 2.68 at 3 bits to 0.51 at 8 bits, and MAE drops from 1.14 to 0.39 in the same range. Overall, the results demonstrate a trade-off between model size and accuracy. Lower n_bits values reduce computational costs but at the expense of accuracy. In contrast, higher n_bits values yield better performance, with 6 bits and above offering a good balance between efficiency and accuracy.

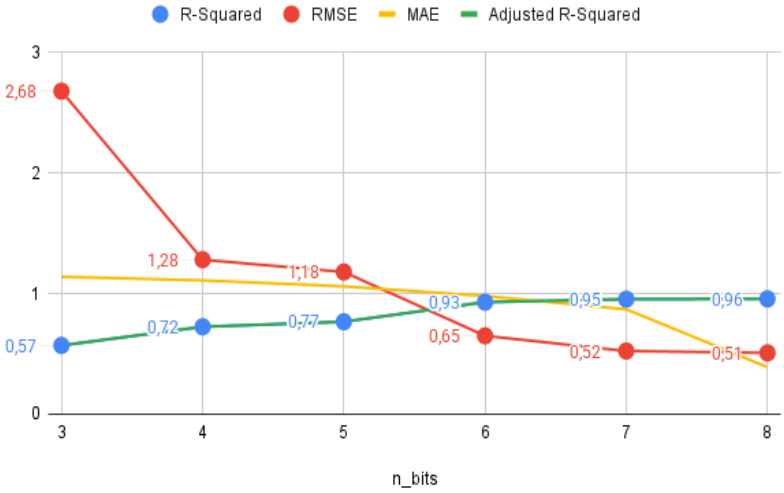


Figure 10. Impact of Quantization Precision (n_bits) on Model Performance

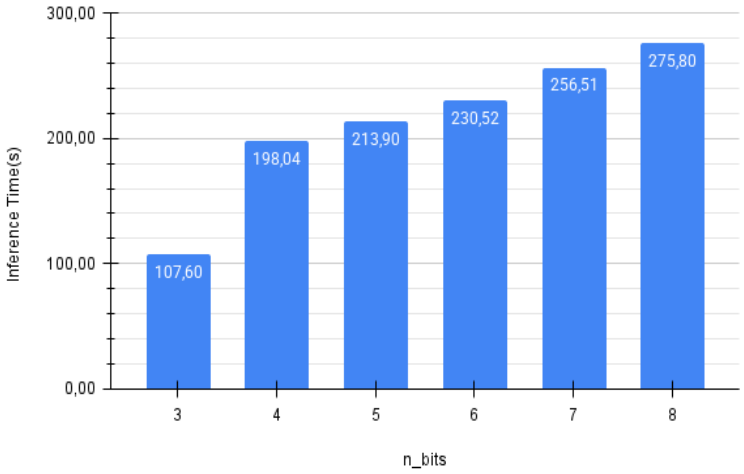


Figure 11. Impact of Quantization Precision (n_bits) on Inference Time

Figure 11 presents the inference time in seconds for each n_bits setting. As expected, increasing the number of bits results in longer inference times. With n_bits = 3, the inference time is 107.60 seconds. As n_bits increases, the time grows steadily, reaching 275.80 seconds at n_bits = 8. This increase is due to the added computational complexity introduced by higher-precision encrypted arithmetic. While higher n_bits values improve accuracy, they also increase the computational load.

These results highlight the trade-off between model accuracy and inference time when using FHE-compatible models. Selecting an appropriate n_bits value requires balancing prediction quality and system performance based on the needs of the application.

The results obtained indicate that the proposed method offers a solid alternative, especially for financial applications where data privacy is essential. One of the strongest aspects of this method is that it facilitates compliance with legal regulations by allowing all user data to be processed in an encrypted form. Further, the high accuracy of models trained on open data when generating predictions with encrypted data supports the practicality of the method. However, the significant computational costs associated with FHE-based systems stand out as a major weakness that limits the widespread use of this technology. Particularly the inference times being quite long compared to open models can lead to performance issues in clients with low processing capacity. This situation necessitates finding a balance

between the level of privacy and the computational cost. Increasing the parameters used for higher accuracy (e.g., n_bits) further extends the encrypted inference time. Therefore, this balance needs to be carefully designed according to the application scenario.

4. Conclusion

In this study, a privacy-preserving machine learning approach based on Fully Homomorphic Encryption (FHE) was developed and implemented using the Concrete ML library. The method was evaluated on three widely used financial distress prediction models: Altman Z-Score, L-Score, and Zmijewski Score. We compared the performance of standard regression algorithms (Random Forest and XGBoost) with their encrypted counterparts implemented in Concrete ML.

Experimental results show that traditional plaintext models showed better performance across all evaluation metrics. For instance, the XGBoost Regressor achieved an R^2 of 0.97, RMSE as low as 0.42, MAE of 3.79, and an Adjusted R^2 of 0.97 on the Zmijewski Score. In contrast, the encrypted version of XGBoost in Concrete ML reached an R^2 of 0.96, RMSE of 0.51, and MAE of 39.05 on the same score, indicating a clear reduction in accuracy, especially in MAE, under encryption. Similar trends were observed across other financial models. For example, the Concrete ML Random Forest model showed an R^2 of 0.66 and MAE of 63.25 for the L-Score, compared to the plaintext model's R^2 of 0.96 and MAE of 5.42.

Despite this accuracy gap, encrypted models consistently maintained reasonably high R^2 (0.66–0.96) and Adjusted R^2 values, with RMSE values remaining within an acceptable range for non-real-time applications. These results affirm the feasibility of using encrypted models for financial predictions, particularly when preserving data confidentiality is essential.

To validate model robustness and guard against overfitting, we applied 5-fold cross-validation during training. The cross-validation results closely aligned with the performance observed on the test set, indicating a well-balanced train-test split and confirming that the models generalize effectively.

Additionally, we investigated the impact of quantization precision (n_bits) on model performance and computational cost. Increasing n_bits from 3 to 8 led to a significant improvement in R^2 (from 0.57 to 0.96) and a corresponding drop in RMSE (from 2.68 to 0.51) and MAE (from 1.14 to 0.39). However, this improvement in accuracy came at the cost of increased inference time, which rose from 107.60 seconds at n_bits = 3 to 275.80 seconds at n_bits = 8. These findings highlight the trade-off between model fidelity and computational efficiency in FHE-enabled systems.

While training times for Concrete ML models were comparable to or shorter than plaintext models, inference times were substantially higher. Traditional models completed inference in under one second, whereas encrypted versions required several minutes. This computational overhead remains the most significant limitation of FHE-based approaches and must be considered when deploying such models in latency-sensitive environments.

Nevertheless, the proposed method offers a practical solution for privacy-conscious financial analytics. It enables accurate financial distress predictions while ensuring that sensitive data remains encrypted throughout processing, supporting compliance with privacy regulations such as GDPR and KVKK.

Future research will focus on optimizing encrypted inference performance through model compression, hardware acceleration, and hybrid encryption techniques. We also plan to assess the scalability of the approach by experimenting with larger datasets and more complex models.

Declaration of Ethical Standards

The authors declare that they comply with all ethical standards.

Credit Authorship Contribution Statement

Author-1: Data Colelction, data analysis/interpretation

Author-2: Data Colelction, data analysisi/interpretation

Author-3: Conceptualization, investigation, methodology and software, review and editing.

Author-4: Conceptualization, data Colelction, investigation, methodology and software, supervision and writing – review and editing.

Author-5: Conceptualization, data analysis/interpretation, methodology, writing – review and editing.

Declaration of Competing Interest

The authors have no conflicts of interest to declare regarding the content of this article.

Data Availability Statement

All data generated or analyzed during this study are included in this published article.

5. References

Ahamed, S.I. and Vadlamani, R., 2022. Privacy-Preserving Wavelet Neural Network with Fully Homomorphic Encryption. arXiv preprint, arXiv:2205.13265. https://doi.org/10.48550/arXiv.2205.13265

Bradley, S., 2024. Comparative Analysis of Two Data Privacy Regulatory Schemes: The GDPR and the CCPA. Studentworks, eRepository, Seton Hall University, USA

Brickell, E.F. and Yacobi, Y., 1988. On privacy homomorphisms (extended abstract). In *Proceedings of the Workshop on Theory and Applications of Cryptographic Techniques*, 117–125.

Basu, P. Roy, T. S., Naidu, R., & Muftuoglu, Z. l., 2021. Privacy enabled financial text classification using differential privacy and federated learning. https://doi.org/10.48550/arXiv.2110.01643

Chen, T. and Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Evren, A.G., 2023. Comparative Analysis of the European Union and Turkish Personal Data Protection Laws: Basic Principles, Legal Grounds, and Rights of Data Subjects. *Kişisel Verileri Koruma Dergisi*, **5**(2), 39–64.

Faneela, B.G., Ahmad, J., Buchanan, W.J. and Jan, S.U., 2025. Privacy-Preserving Credit Card Approval Using Homomorphic SVM: Toward Secure Inference in FinTech Applications. arXiv preprint arXiv:2505.05920. https://doi.org/10.48550/arXiv.2505.05920

Frery, J., Stoian, A., Bredehoft, R., Montero, L., Kherfallah, C., Chevallier-Mames, B., & Meyre, A., 2023. Privacy-preserving tree-based inference with fully homomorphic encryption. Cryptology ePrint Archive. https://doi.org/10.48550/arxiv.2303.01254

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, **29**(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Gentry, C., 2009. A Fully Homomorphic Encryption Scheme. Ph.D. Dissertation, Stanford University, Stanford, CA, USA.

Genuer, R., Poggi, J.-M. and Tuleau-Malot, C., 2010. Variable selection using Random Forests. *Pattern Recognition Letters*, **31**(14), 2225–2236. https://doi.org/10.1016/j.patrec.2010.03.014

He, H. and Garcia, E.A., 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, **21**(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hyndman, R.J. and Athanasopoulos, G., 2018. *Forecasting: Principles and Practice*. OTexts: Melbourne, Australia. OTexts.com/fpp2.

James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An Introduction to Statistical Learning*. **Vol. 112**, Springer, New York.

Kadhe, S.R. et al., Privacy-Preserving Federated Learning over Vertically and Horizontally Partitioned Data for Financial Anomaly Detection**,** arXiv:2310.19304. https://doi.org/10.48550/arXiv.2310.19304

Ke, G. et al., 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, Long Beach, CA, USA. pp. 3146–3154.

Lee, J.-W. et al., 2022. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, **10**, 30039–30054. https://doi.org/10.1109/ACCESS.2022.3159694

Legiest, W., Turan, F., Van Beirendonck, M., D'Anvers, J. P., & Verbauwhede, I., 2023. *Neural network quantisation for faster homomorphic encryption. 29th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 1–3.

Liaw, A. and Wiener, M., 2002. Classification and Regression by Random Forest. *R News*, **2**(3), 18–22.

Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H., & Aaraj, N. 2022. Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, **110**(10), 1572–1609. https://doi.org/10.1109/JPROC.2022.3205665

Miles, J., 2005. R-squared, adjusted R-squared. *Encyclopedia of Statistics in Behavioral Science*. https://doi.org/10.1002/0470013192.bsa526

Naresh, V.S. and Ayyappa, D., 2024. PPDNN-CRP: Privacy-preserving deep neural network processing for credit risk prediction in cloud: A homomorphic encryption-based approach. *Journal of Cloud Computing*, **13,** 149**.** https://doi.org/10.1186/s13677-024-00711-y

Pattnaik, D., Ray, S., & Raman, R. (2024). Applications of artificial intelligence and machine learning in the financial services industry: A bibliometric review. Heliyon, 10(1), e23492. https://doi.org/10.1016/j.heliyon.2023.e23492

Shah, S.B., 2025. Evaluating the effectiveness of machine learning in forecasting financial market trends: A fintech perspective. 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, pp. 1-6.

Yeom, S., Giacomelli, I., Fredrikson, M. and Jha, S., 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 268–282.

Willmott, C.J. and Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, **30**(1), 79–82.

Xu, R., Baracaldo, N. and Joshi, J., 2021, Privacy-preserving machine learning: Methods, challenges and directions, arXiv:2108.04417. https://doi.org/10.48550/arXiv.2108.04417

Internet References

1- Shinde, S., 2025. What Are Machine Learning Applications? Top 10 Industry and Real-world Use Cases. Emeritus blog. Available at: https://emeritus.org/blog/machine-learning-what-are-machine-learning-applications, (28.04.2025)

2- Yilmaz, E., Akturk, B., Civelek, D. and Buyuktanir, T., 2025. Audit opinions of Turkish Public Companies [Dataset]. Kaggle. Available at: https://www.kaggle.com/datasets/agrafintech/financial-data-of-turkish-public-companies

3- Zama, 2023. Concrete ML: Privacy-preserving machine learning using fully homomorphic encryption. GitHub repository. Available at: https://github.com/zama-ai/concrete-ml