International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(16s) 2026 pp. 1261-1274
https://doi.org/10.70917/ijcisim-2026-4656
Article

Enhancing Data Privacy in Linear Regression Through Homomorphic Encryption

Krishnakumar D[1], Sonam Mittal[2], K.R. Ramkumar[1]

[1]Chitkara Institute of Engineering and Technology, Chitkara University, Punjab, India
[2]School of Engineering and Technology, CGC University, Mohali, Punjab, India
Krishna1007cs.phd20@chitkara.edu.in, sonam.mittal1287@gmail.com and k.ramkumar@chitkara.edu.in

Abstract: Big data analysis is required to predict future trends using machine learning. A lot of processing and computation is required to make this analysis. Conventional encryption methods may help to achieve privacy during the resting stage of data over the cloud. The computation of data needs to be in its original form or needs to be decrypted before any computation, which leaves the data again at risk and makes it more vulnerable. The latest homomorphic encryption algorithms provide a way to perform the computation over data in its encrypted form and help to analyze big data without compromising the privacy and security of data, even when data is in use. In this paper, we implemented the linear regression model over plain data and encrypted data to measure and analyze the fitness, MSE, and RMSE scores of models and the space complexity of plain data and encrypted data. The results show that the fitness score of the linear regression model for plain data and encrypted data is almost the same and differs by only 0.00003551, whereas the RMSE score is 386.9949011 and 364.03753641, respectively. It is observed that the overall performance of the linear regression model improved for encrypted data, and the error is also reduced while maintaining data privacy

Keywords: Homomorphic Encryption, Camouflaged Process, Linear Regression, Data Privacy, Big Data and Cloud

1. Introduction

Over the cloud, big data floats freely, as outsourced by various clients. New threats are also arising with technological improvements. A data breach happens every day and its rate is increasing daily. Many conventional cryptographic algorithms are designed by researchers to maintain the confidentiality and privacy of the data [1] by converting it to an unreadable form. Encryption encrypts the original data (plain data) and converts it to an unreadable form (encrypted data), and decryption decrypts the encrypted data back to its original form, i.e., plain data [2]. The existing conventional cryptographic algorithms maintain privacy and confidentiality to a large extent over the cloud, but only when data is in the resting stage or stored [3]. But what about the privacy and security of data when it is in use or during its computation? To compute different operations, data must be in its original form. Computation over encrypted data cannot result in desired and accurate results.

Nowadays, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are technologies that change the perspective to work and analyze the available data, to predict future trends in different sectors [4]. Various ML and DL models are used to analyze available datasets and predict forthcoming trends. The datasets may be available freely or may not be due to their confidentiality, but researchers still want to analyze such datasets. Homomorphic Encryption (HE) algorithms are the latest cryptographic techniques that may help to preserve the privacy and confidentiality of such datasets [5]. HE is the only way to obtain the desired results while performing the computation over encrypted data itself [6].

The remaining part of this research work is organized like this: in Section 2, we present the background study. Section 3 provides the algorithm for this proposed algorithm. Section 4 defines the proposed algorithm results, and comparative analysis has been discussed for plaintext and encrypted datasets in this section. In Section 5, the paper concludes with our discussion of future scope.

2. Literature survey

In 2024, [7] authors Chen Yang and others put forward a privacy-preserving protocol for K-port clustering using fully homomorphic encryption (FHE) and built on the K-port clustering protocol to cope with the drawback of computational overhead found in traditional duration hinged on FHE for clustering by restricting multiple iterations on encrypted data to just one interaction. Using the CKKS scheme and the speed-up capability of the single-instruction-multiple-data concept, the method drastically reduces the number of rounds communicated between the data owner and the server, in comparison with the FHE clustering systems developed earlier. This protocol incorporates the mini-batch k-means method for large datasets, providing performance that is only slightly lower than the results with unencrypted data, while facilitating the processing of very large volumes of data.

In 2024, Syed Imtiaz et al. [8] combined Fully Homomorphic Encryption (FHE) and Chaotic Extreme Learning Machine (CELM) to propose reinforcement of privacy in machine learning systems, especially when sensitive information is concerned. Weights and biases are not randomly generated, but instead, some kind of logistic map is used. Although there is a marginal security compromise since hidden layer outputs have to be decrypted so as to estimate the weights, the model performs equally well or even better than the standard Extreme Learning Machines on a range of datasets. The research noted that the model is not the best for every use case; however, CELM is the best in most cases.

In 2024, A novel privacy-preserving protocol for the epigenetic pacemaker (EPM) model was proposed by Meir Goldenberg et al. [9], which is more privacy-conscious and computationally efficient than the previous methods. Their approach employs one server to perform computations securely, decreasing the chances of a corrupted server as well as the overall communications overhead during the computations. The other reason behind this proposition includes the use of symbolic algebra and number theory, which assist in enhancing the time of the protocol as well as its suitability for use in parallel processes. This recent development provides good protection of sensitive information within medical systems, but it can also be useful in many other aspects of computational biology that need to be protected from prying eyes.

In 2023, Thomas Prantl et al. [10]probed the possibility of using homomorphic encryption for secure processing of data in the cloud and focused on linear regression for that purpose. They discussed two implementations: the first where the customer provides encrypted data to the cloud for processing and the second where the customer participates in computations by performing encryption and decryption of certain parameters to control the multiplicative depth. The findings suggest that even though homomorphic computations take a significantly longer time, by as much as 80 million times, than the non-homomorphic methods, it is still possible to perform encryption within a short time, and decryption is instant.

In 2023, a new token-meriting distributed mutual exclusion algorithm was introduced in [11] by Ashish Singh et. al aimed at improving resource sharing for distributed systems. This algorithm is more scalable and has less message complexity compared to other available solutions. It characterizes message exchanges for every critical section invocation using the log functions for different loads endured. The algorithm is proposed to work on an almost complete tree topology such that the order of critical section requests is preserved while achievable properties of distributed mutual exclusion are satisfied. In future work, research will be directed towards generalized token and node loss.

In 2023, Ravital Solomon [12], alongside several others, proposed smartFHE, a framework for enabling blockchains with secure smart. contracts based on Fully Homomorphic Encryption (FHE). Unlike previous solutions which shifted the burden of computation on users off-chain, smartFHE enables miners to carry out computations privately while working on encrypted data. Users are only required to efficiently zero-knowledge prove the correctness of their private inputs. This methodology overcomes issues like concurrency and performance, displaying higher effectiveness than the existing systems. The framework seeks to improve the viability of privacy-enhancing decentralized applications.

In 2022, the authors implemented homomorphic encryption with linear regression [13]. Homomorphic encryption using the SEAL library is implemented over six different datasets to analyze the performance of linear regression over encrypted text. The authors conclude that it can be used to secure analysis without affecting the accuracy of the regression model.

In 2022, [3] authors presented the usage of an integer-based HE algorithm, DGHV, in banking data over the cloud. DGHV-like HE schemes can be used to secure computation through data sanitization and explain various challenges like growth of ciphertext and noise, computational overhead, and overall complexity, which need to be faced while implementing the HE schemes over large datasets.

In 2021, Zhang et al. [14], introduced a new scheme that allows clients to perform a secure linear regression model on a cloud server. A fair payment platform using blockchain is designed, where a client can upload the computational task over the cloud and the cloud needs to perform computation securely, which was verified by blockchain. The whole mechanism of verification did not expose any information of the client, and the scheme was publicly verifiable, secure, and efficient.

In 2018, the author used Paillier's homomorphic encryption scheme to achieve multiple regression over a medical data set of deaths due to strokes [15]. The selected HE scheme for encryption is partially homomorphic a support only addition operations over encrypted text. The author concludes that the use of HE schemes increased the overhead to a large extent, and the result shows that its processing time, for n = 100,000,000 patients, was 2.6 days.

In 2017, the author [16] implemented the linear and functional regression model. The results show that for shorter data, linear regression outperforms, whereas for more data, the functional regression model outperforms. Linear regression proved to be a low-bias, high-variance model, whereas functional regression proved to be a high-bias, low-variance model. Linear regression is inherently a high-variance model as it is unstable to outliers, so one way to improve the linear regression model is by collecting more data. Functional regression, how-

In 2016, author [17], analyzed the statistical analysis of encrypted data of five different real datasets. A matrix-based HE algorithm is used to analyze the data using linear regression. The result shows that it takes approximately 20 minutes more for secure analysis of the encrypted dataset, which consists of 30k rows with 6 different parameters as input.

In 2015, the author [4]reviewed various homomorphic encryption schemes to analyze different kinds of statistics and machine learning algorithms that can be implemented over encrypted text. The author faced many challenges like message space, computational cost, ciphertext size, and security issues, and finally introduced a new document, a high-performance Homomorphic Encryption R package, to experiment with recent homomorphic schemes.

In 2011, the author [18] proposed an approach that provides statistical analysis using linear regression and also uses homomorphic encryption to design a new protocol for linear regression analysis over encrypted text. In an experiment over a dataset of 51,016 cases with 22 covariates, the combined secure-private approach helps to analyze the data securely.

3. Method and Materials

The main objective of the research is to perform a secure analysis from an encrypted dataset without compromising the privacy and security of data. Over the cloud, a data breach may leak all sensitive and confidential information to hackers. We also study how the encryption of data affects the performance of the machine learning model. The different steps to perform the secure analysis using a fully homomorphic encryption and linear regression model are as follows:

1. Data Collection

2. Enhanced Prediction Methodology

3. Applied Model Algorithms

4. Comparative Analysis

3.1 Data Collection

The gold price dataset, obtained online, [19] contains six columns named date, country, state, Location, pure gold (24 k), and standard gold (22 k). The data range from 01/02/2006 to 10/10/2020, i.e., a total 4972 number of rows are there. Out of the specified columns, date and pure gold (24 k) price are the target attributes for implementation of the LR model and are used to forecast the gold price. 80% of the complete data is used to train the LR model, and the remaining 20% is used to test the model. A sample of data from 01/02/2020 is shown in Figure 1.

	A	B	C	D	E	F
1	**Date**	**Country**	**State**	**Location**	**Pure Gold (24 k)**	**Standard Gold (22 K)**
2	01/02/2006	India	Tamilnadu	Chennai	768	711
3	01/03/2006	India	Tamilnadu	Chennai	770.5	713
4	01/04/2006	India	Tamilnadu	Chennai	784.5	726
5	01/05/2006	India	Tamilnadu	Chennai	782.5	725
6	01/06/2006	India	Tamilnadu	Chennai	776	719
7	01/07/2006	India	Tamilnadu	Chennai	787.5	729
8	01/09/2006	India	Tamilnadu	Chennai	790	732
9	01/10/2006	India	Tamilnadu	Chennai	791	732
10	01/11/2006	India	Tamilnadu	Chennai	788	730
11	01/12/2006	India	Tamilnadu	Chennai	789	731
12	1/13/2006	India	Tamilnadu	Chennai	790	732
13	1/16/2006	India	Tamilnadu	Chennai	802	743
14	1/17/2006	India	Tamilnadu	Chennai	812	752
15	1/18/2006	India	Tamilnadu	Chennai	799	740
16	1/19/2006	India	Tamilnadu	Chennai	803	744
17	1/20/2006	India	Tamilnadu	Chennai	810	750
18	1/21/2006	India	Tamilnadu	Chennai	808.5	749
19	1/23/2006	India	Tamilnadu	Chennai	808.5	749

3.2 Enhanced Prediction Methodology

In this research, we have implemented Linear Regression (LR), an ML model that is applied to the available dataset of gold prices to predict the gold price in the future. Now, the Camouflaged Plaintext Fully Homomorphic Encryption (CpFHE) algorithm is applied to encrypt the target attribute 'Pure Gold (24 k)' from the available dataset of gold prices. After encrypting the gold price in the dataset, the LR model is applied again to predict the gold price in the future. Now, the results from both cases. Finally, a comparative analysis has been done to analyze the performance of the linear regression model over encrypted and plain datasets in terms of fitness, MSE, and RMSE score [20].

3.1 Data Collection

In this research, for encrypting the dataset, a camouflaged fully homomorphic encryption model is used. To make predictions on future trends of gold price, a simple ML model, machine learning, is used. Both are discussed below in detail.

a) Linear Regression: The linear regression model is used to analyze the relationships between independent and dependent variables. The paper analyzes the relationship between data and gold prices using simple linear regression. The model works for the dependent variable y and independent variable x, which influences the variable y and helps to predict the future trend in gold prices.

A simple linear regression model can be expressed as:

$$y = a_0 + a_1 x + e \qquad \text{Eq. (2)}$$

The variables y and x are dependent variables and independent variables, respectively. Variable a is the constant term showing the intercept of the regression line (vertical axis), and a_1 represents the slope of the regression

line. The last variable e represents the random error, used to specify the effects of random factors on the dependent variable [20].

The variables a_0 and a_1 are evaluated by regression analysis after observing the samples (x_i y_i). To evaluate the performance of the linear regression model, different model parameters are evaluated as given below:

Using Python libraries, the predefined method, the score returns the coefficient of determination of the prediction.

i. **Coefficient of determination (R^2)** [21]**:** We can define the coefficient of determination R^2 is the proportion of variation in the outcome that is explained by predictor variables, as in eq. 1. We can say that,

$$R^2 = 1 - \frac{u}{v} \qquad \text{Eq. (1)}$$

where u and v are the residual sum of squares $(y_{true} - y_{pred})^2 . sum()$ and the total sum of squares $(y_{true} - y_{true.mean})^2 . sum()$ respectively.

ii. **Mean Squared Error (MSE)**[21]*:* It is the average squared difference between the observed actual outcome values and the values predicted by the model.

iii. **Root Mean Squared Error (RMSE)** [21]**:** It is used to measure the average error made by the model in predicting the outcome of an observation. Mathematically, the RMSE is the square root of the MSE. Technically, MSE and RMSE can be expressed as in eq. 2 and eq. 3.

MSE = mean $(observed - predicted)^2$ Eq. (2)

RMSE = sqrt (MSE) Eq. (3)

b) **Proposed Homomorphic Encryption** [22][23]**:** Many homomorphic encryption algorithms exist to preserve the privacy of data. A camouflaged plaintext fully homomorphic encryption algorithm is a proposed FHE that can work integers or floating-point plaintext to encrypt it and allows computation over encrypted data with better performance; it also maintains the privacy and security of data.

In the CpFHE model, the camouflage process is applied to the original plaintext M , and then results in camouflage plaintext, M_{cam} is encrypted using a specified encryption algorithm to generate the camouflaged ciphertext, C_{cam}.

Algorithm:

Key Generation Algorithm ($KeyGen$):

i. Generate random prime numbers as P_i, where $P_i \in [2^1, 2^\lambda]$, λ, a security parameter $\in \mathbb{N}$ and $\mathbb{N} \in [1, n]$ and generate the substitution box SHE -Box, matrix of size 16×16 using P_i.

ii. Return the secret key as $S_{sk,}$ Using the random selection method.

Encryption Algorithm ($Encrypt$):

i. Input the plaintext/ message as M, where $M \in \mathbb{R}, \mathbb{Z}$, and apply the Camouflage process (Cp) to generate camouflaged plaintext as M_{cam} accordingly.

ii. Camouflage process (Cp):

   a. Find the next prime number as P_{Next}, to message M and calculate d i.e., the difference of P_{Next}, and M .

   b. Find the random error (Err) and generate. M_{cam} such as $M_{cam} \leftarrow P_{Next} + Err$.

iii. Encrypt M_{cam} to get camouflaged ciphertext as C_{cam}, as, $C_{cam} \leftarrow M_{cam} \times S_{sk}$.

Decryption Algorithm ($\boldsymbol{Decrypt}$):

i. Input C_{cam} And divide it by S_{sk} And obtain the M_{cam}.

ii. Apply the Reverse Camouflage process (CP_{rev}) to generate the original plaintext, M.

iii. Reverse Camouflage process (CP_{rev}):

a. $P_{Next} \leftarrow M_{cam} - Err$

b. $M \leftarrow P_{Next} - d.$

And retrieves the original plaintext M.

Evaluation Algorithm ($\boldsymbol{Eval}$):

Generate two camouflaged ciphertexts as $C_{cam}\ and\ C'_{cam}$ for two different plaintexts $M\ and\ M'$, using an Encryption algorithm (Encrypt), respectively.

i. Evaluate the $C_{cam}\ and\ C'_{cam}$ using the evaluation function as f_{eval}, and it consists of algebraic addition or multiplication operations.

ii. Calculate the result of the applied operation, as C_{eval} using f_{eval}.

a. $C_{eval} \leftarrow C_{cam} + C'_{cam}\ or\ C_{eval} \leftarrow C_{cam} \times C'_{cam}$

iii. Also, find the result of the applied operation as M_{eval} over the different plaintext, $M\ and\ M'$ using f_{eval}.

a. $M_{eval} \leftarrow M + M'\ or\ M_{eval} \leftarrow M \times M'$

iv. If $C_{eval} = M_{eval}$, for respective f_{eval} Then homomorphism is achieved, or else homomorphism fails.

4. Experiment and Results

4.1 Performance Parameters

Performance metrics considered for the evaluation of linear regression models are as follows:

i. R^2 Value

ii. MSE Score

iii. RMSE Score

Apart from these, an encryption algorithm is implemented to evaluate the linear regression model over encrypted text. Therefore, parameters to evaluate the encryption algorithm are, also taken into consideration.

i. Encryption Time (Time Complexity)

ii. Memory Required (Space Complexity)

iii. Secret Key Size

4.2 Experiment I – Linear Regression over Plaintext

In this experiment, the linear regression model is implemented over a given dataset to evaluate it. For evaluation, a ten-day prediction of the gold price has been done. Table 1, given below shows the same along with the R^2, MSE, and RMSE score.

Table 1 Different Performance Metrics Values for Linear Regression Model Over Plaint Dataset

Date	R^2	MSE	RMSE	Predicted Price (Rs.)
11/10/2022	0.8368	149765.05	386.9949	4449.56
12/10/2022	0.8368	149765.05	386.9949	4450.14
13/10/2022	0.8368	149765.05	386.9949	4450.72
14/10/2022	0.8368	149765.05	386.9949	4451.29
15/10/2022	0.8368	149765.05	386.9949	4451.87
16/10/2022	0.8368	149765.05	386.9949	4452.45
17/10/2022	0.8368	149765.05	386.9949	4453.03
18/10/2022	0.8368	149765.05	386.9949	4453.61
19/10/2022	0.8368	149765.05	386.9949	4454.194
20/10/2022	0.8368	149765.05	386.9949	4454.772

Figures 2 and 3 show the scattering of the gold price day-wise for the train and test dataset. The gold price varies to the date variable over a plain dataset using the linear regression model.

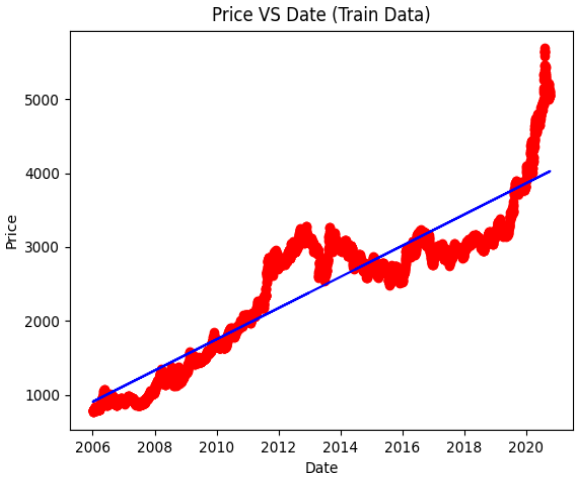

Figure 2 Scattering of Gold Price for Train Dataset over Plain Dataset using LR Model

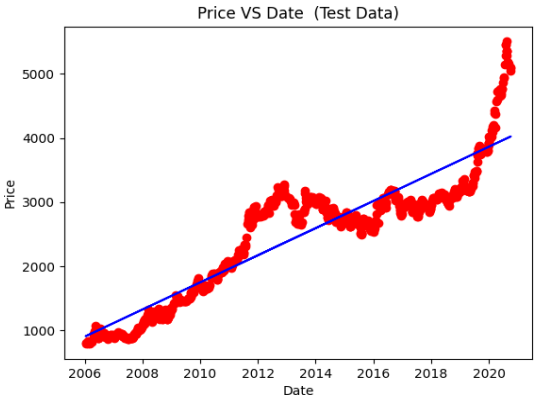

Figure 3 Scattering of Gold Price for Test Plain Dataset over Plain Dataset using LR Model

4.3 Experiment II – Linear Regression over Encrypted Text

In this experiment, the linear regression model is implemented over an encrypted dataset. The given dataset is encrypted using the proposed FHE algorithm named CpFHE. Table 2 shows the encryption time to encrypt the given dataset comprised of 4971 records. It also shows the secret key and random error required to perform the encryption of a given data set through the proposed CpFHE algorithm. Using the specified key and random error, the data set is encrypted to obtain the encrypted dataset over which a linear regression model is executed to predict the future gold price. Day-wise, a day's prediction of the gold price has been done as given in Table 2. The average key size is 3.60 digits and the additional time to encrypt the data is about 77.2603802 seconds.

Table 2 Different Performance Metrics Values for CpFHE Algorithm

Date	Key	Error	Plain Memory	Encryption Memory	Encrypted Predicted Price (Rs.)	Encryption Time (Sec.)
11/10/2022	1823	0.91	2257	2490	4434.1217	77.10798454
12/10/2022	59	0.857	2257	2490	4434.7010	77.12251377
13/10/2022	1579	0.552	2257	2490	4435.2803	77.09040761
14/10/2022	383	0.424	2257	2490	4435.8596	77.20115495
15/10/2022	1129	0.794	2257	2490	4436.4389	77.31921005
16/10/2022	2791	0.405	2257	2490	4437.0182	77.49230194
17/10/2022	673	0.105	2257	2490	4437.5975	77.46020031
18/10/2022	2549	0.022	2257	2490	4438.1768	77.42366576
19/10/2022	2297	0.862	2257	2490	4438.7562	77.22177792
20/10/2022	1889	0.299	2257	2490	4439.3355	77.16458511

Table 3 Different Performance Metrics Values for Linear Regression Model Over Encrypted Dataset

Date	R^2	Encrypted MSE	Encrypted RMSE	Decrypted MSE	Decrypted RMSE
11/10/2022	0.8367790	498726428386.6860	706205.6559	273574539.1107	363.4765
12/10/2022	0.8367790	522387851.8239	22855.8056	8854007.5298	363.5295
13/10/2022	0.8367790	374156509643.5620	611683.3410	236957867.2926	363.8345
14/10/2022	0.8367790	1168984407923.9900	1081195.8231	418840681.6914	363.9815
15/10/2022	0.8367790	191283244423.0580	437359.3996	169427119.1848	363.5925
16/10/2022	0.8367790	791791865924.7030	889826.8741	344706927.0236	363.6375
17/10/2022	0.8367790	67970298000.2123	260711.1390	100995962.0365	364.2815
18/10/2022	0.8367790	975054108252.9270	987448.2813	382524146.5555	364.3645
19/10/2022	0.8367790	791791865924.7010	889826.8741	344706926.9106	363.5245
20/10/2022	0.8367790	535491968344.7450	731773.1673	283479049.4086	364.0875

Table 3 given below shows the R^2, MSE, and RMSE score, for day-wise prediction. The data illustrate a stable model performance at all dates, with the mean R^2 of 0.8368, which is equivalent to the explanation of around 83.7%

variance in the outcome. Noticeable are the MSE and RMSE values during encryption, which vary significantly, with really high peaks on October 14, 2022. This indicates that encryption greatly boosts error variance. The magnitude of the errors is lower and stabilizes when decrypted. Decrypted RMSE stays almost constant around 363.5, which indicates that the temporal stability of the predictive accuracy of the model is sound, though the encryption process adds error variance but retains the general error trend. The peak on October 14 should be further studied because it may point to an anomaly within the data.

Fig. 3 and 4, show the scattering of the gold price day-wise for the train and test dataset where the gold price also varies to date variable over the encrypted dataset.

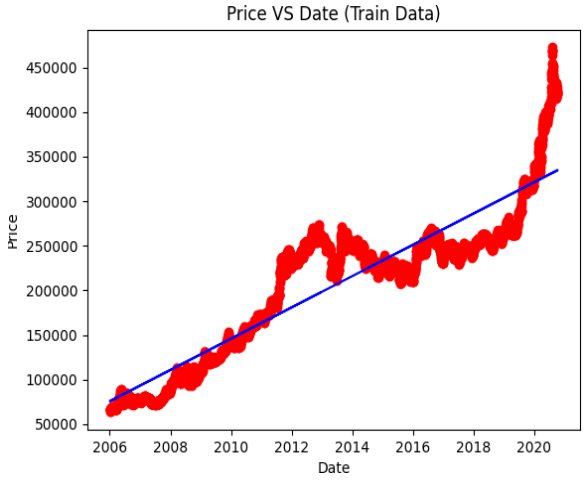

Figure 4 Scattering of Gold Price for Train Dataset over Encrypted Dataset using LR Model

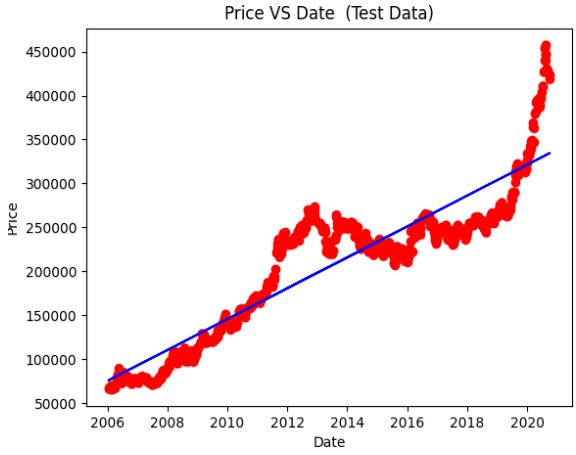

Figure 5 Scattering of Gold Price for Test Dataset over Encrypted Dataset using LR Model

4.4 Comparative Analysis

In this section, comparative analysis has been done for linear regression, an ML model for a plain dataset, and an encrypted dataset. Fig. 6 and Table 4 show the analysis of the prediction of gold price for both the implemented model through experiment I and experiment II. The graph shows the day-wise analysis of predicted gold price over the test dataset to analyze how linear regression model performance varies for the plain dataset and encrypted dataset.

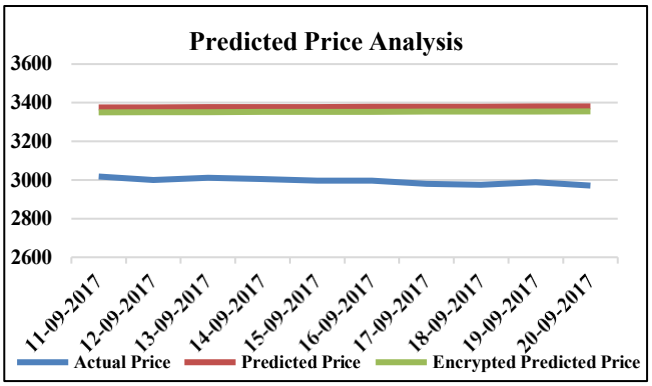

Figure 6 Analysis of Predicted Price over Plain and Encrypted Dataset

Table 4 Analysis of Predicted Price over Plain and Encrypted Dataset

Date	Actual Price (Rs.)	Predicted Price (Rs.)	Enc (Pred) Price (Rs.)
11/09/2017	3018	3375.609718	3359.497956
12/09/2017	3000	3376.188668	3360.077268
13/09/2017	3012	3376.767618	3360.656579
14/09/2017	3005	3377.346569	3361.235891
15/09/2017	2996	3377.925519	3361.815203
16/09/2017	2997	3378.504469	3362.394515
17/09/2017	2980	3379.083419	3362.973827
18/09/2017	2975	3379.66237	3363.553139
19/09/2017	2989	3380.24132	3364.132451
20/09/2017	2971	3380.82027	3364.711763
Sum:	29943	33782.14994	33621.04859
Average:	2994.3	3378.214994	3362.104859

Fig. 7 and 8 show the comparison of R^2values and predicted gold price respectively after comparing results from experiment I and experiment II.

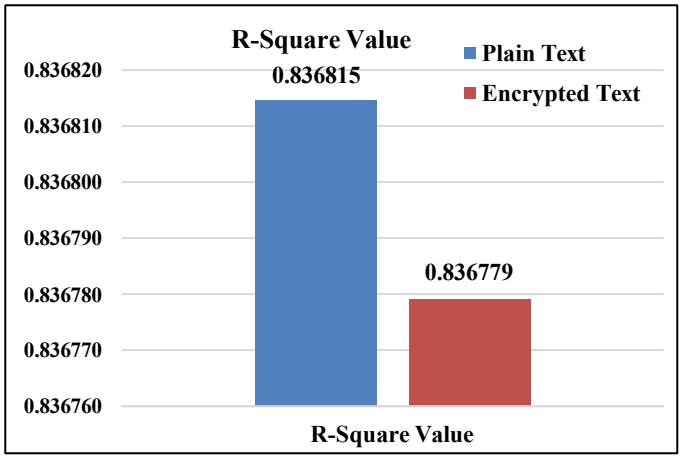

Figure 7 Comparison of R-Squared Values

The space complexity, i.e., the memory required to store the gold price is 2257 and 2290 bytes, from the plain dataset and encrypted dataset respectively. The encrypted gold price dataset requires more space as compared to the plain dataset.

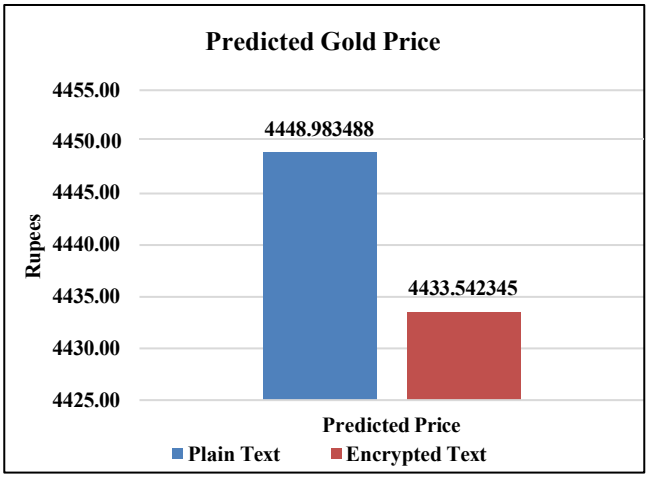

Figure 8 Comparison of Predicted Gold Price

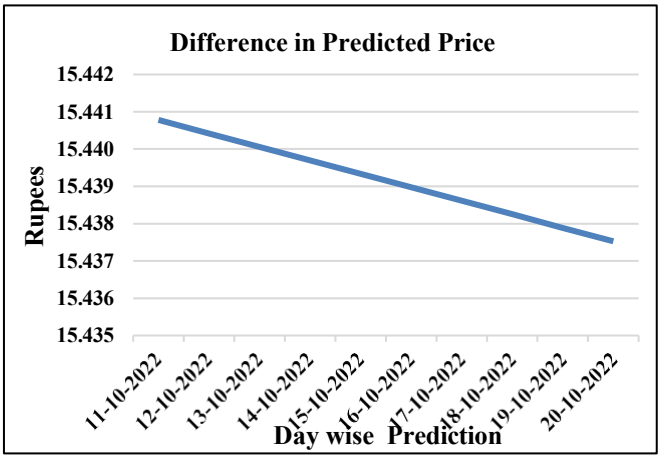

Figure 9 Analysis of Difference in Prediction Gold Price

Fig. 9, shows the difference in future predicted god price values a after applying linear regression model over the plain dataset and encrypted dataset, referring from Table 1 and Table 2 respectively. A day-wise analysis shows that the difference is decreasing and gives more accurate and better results.

Fig. 10, shows the difference in RMSE score after applying a linear regression model over a plain dataset and encrypted dataset, referring from Table 1 and Table 3 respectively.

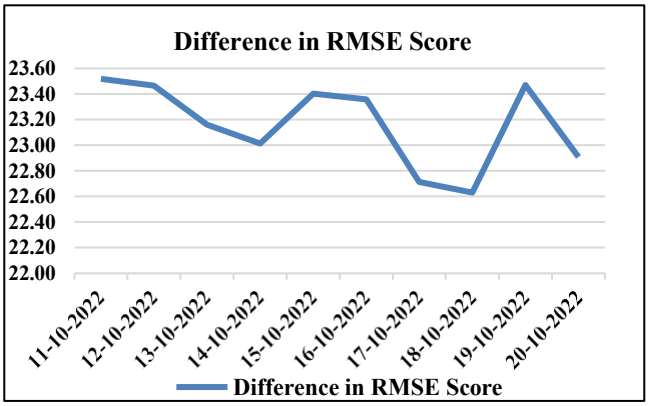

Figure 10 Analysis of Difference in RMSE Score

Table 5, show the R^2 values, RMSE score, Predicted Gold Prices, and Memory required for plain and encrypted datasets. R^2 values for both the models are 0.836815 and 0.836779 respectively and the difference among them is 0.000036 that is negligible. RMSE score is also less for the LR model over the encrypted dataset. The predicted gold price over the encrypted dataset is closer to the actual price so it can be said that the linear regression model performs better on an encrypted dataset instead of a plain dataset. The difference between them is about Rs. 15.44 only.

Table 5 Comparative Analysis

Performance Metrics	R^2 Value	RMSE Score	Predicted Price	Memory
LR over Plain Dataset	0.836815	386.9949	4448.9834	2257
LR over Encrypted Dataset	0.836779	363.8498	4433.5423	2490

5. Conclusion and Future Scope

The paper also focuses on techniques for securing data during analysis. The CpFHE algorithm securely encrypts the data, and implementing the LR model over the encrypted dataset enables the successful execution of the proposed algorithms with improved performance. Table 5, shows the R^2 values, RMSE score, Predicted Gold Prices, and Memory required for plain and encrypted datasets. R^2 values for both models are 0.836815 and 0.836779, respectively, and the difference between them is 0.000036, which is negligible. RMSE score is also less for the LR model over the encrypted dataset. The predicted gold price over the encrypted dataset is closer to the actual price, so it can be said that the linear regression model performs better on the encrypted dataset instead of the plain dataset. The difference between them is about Rs. 15.44 only.

The encrypted gold price dataset requires more memory as compared to the plain dataset and it can be ignored. The additional time is also there to encrypt the data which is about 77.26 seconds. When we compare the increased space and time complexity and privacy of sensitive data, the cost of increased memory space can be neglected but the privacy of data is the primary factor that needs to be considered as the first preference in big data over clouds. Losing sensitive and private data may lead to unrecoverable loss.

The proposed methodology to secure data analysis may help in the future to analyze smaller datasets over the cloud securely and privately. It may further be tested and implemented for more complex datasets to check their accuracy and to evaluate other performance metrics depending on the different machine and deep learning models.

REFERENCES

1. S. Mittal and K. R. Ramkumar, “Research perspectives on fully homomorphic encryption models for cloud sector,” J. Comput. Secur., vol. 29, no. 2, pp. 135–160, 2021, doi: 10.3233/JCS-200071.
2. R. A., “Cryptographic Basics,” in An Introduction to Cryptograpy, 2nd ed., K. H. Rosen, Ed. New York: Taylor & Francis, 2016, pp. 79–130.
3. S. Mittal, P. Jindal, and K. R. Ramkumar, “Data privacy and system security for banking on clouds using homomorphic encryption,” in 2021 2nd International Conference for Emerging Technology, INCET 2021, 2021, pp. 1–6, doi: 10.1109/INCET51464.2021.9456345.
4. L. J. M. Aslett, P. M. Esperança, and C. C. Holmes, “A review of homomorphic encryption and software tools for encrypted statistical machine learning,” pp. 1–21, 2015, [Online]. Available: http://arxiv.org/abs/1508.06574.
5. H. Abdinasibfar, C. Nuoskala, and A. Michalas, “The HHE Land: Exploring the Landscape of Hybrid Homomorphic Encryption,” ACM Comput. Surv., vol. 58, no. 12, pp. 304:1-304:35, 2026, doi: 10.1145/3807507.
6. S. Mittal and K. R. Ramkumar, “Understanding integer-based fully homomorphic encryption,” AIP Conf. Proc., vol. 2357, no. 1, p. 120005, 2022, doi: 10.1063/5.0080604.
7. C. Yang, J. Chen, W. Wu, and Y. Feng, “Optimized Privacy-Preserving Clustering with Fully Homomorphic Encryption,” Cryptol. ePrint Arch., vol. 2, 2024, [Online]. Available: https://eprint.iacr.org/2024/1141.
8. S. I. Ahamed and V. Ravi, “Privacy-Preserving Chaotic Extreme Learning Machine with Fully Homomorphic Encryption,” Lect. Notes Networks Syst., vol. 997 LNNS, pp. 599–623, 2024, doi: 10.1007/978-981-97-3242-5_40.
9. M. Goldenberg, L. Mualem, A. Shahar, S. Snir, and A. Akavia, “Privacy-preserving biological age prediction over federated human methylation data using fully homomorphic encryption,” Genome Res., p. gr.279071.124, 2024, doi: 10.1101/gr.279071.124.
10. T. Prantl et al., “Performance Impact Analysis of Homomorphic Encryption: A Case Study Using Linear Regression as an Example,” pp. 284–298, 2023, doi: 10.1007/978-981-99-7032-2_17.
11. A. S. Parihar and S. K. Chakraborty, “A new resource sharing protocol in the light of token-based strategy for distributed system,” Int. J. Comput. Sci. Eng., vol. 26, no. 1, pp. 78–89, 2023, doi: 10.1504/IJCSE.2023.129149.
12. R. Solomon, R. Weber, and G. Almashaqbeh, “smartFHE: Privacy-Preserving Smart Contracts from Fully Homomorphic Encryption,” Proc. - 8th IEEE Eur. Symp. Secur. Privacy, Euro S P 2023, pp. 309–331, 2023, doi: 10.1109/EuroSP57164.2023.00027.
13. B. Chen and X. Zheng, “Implementing linear regression with homomorphic encryption,” Procedia Comput. Sci., vol. 202, pp. 324–329, 2022, doi: 10.1016/j.procs.2022.04.044.
14. H. Zhang, P. Gao, J. Yu, J. Lin, and N. Xiong, “Machine Learning on Cloud with Blockchain: A Secure, Verifiable and Fair Approach to Outsource the Linear Regression for Data Analysis,” IEEE Trans. Netw. Sci. Eng., pp. 1–12, 2021, doi: 10.1109/TNSE.2021.3110101.
15. H. Kikuchi, C. Hamanaga, H. Yasunaga, H. Matsui, H. Hashimoto, and C. I. Fan, “Privacy-preserving multiple linear regression of vertically partitioned real medical datasets,” J. Inf. Process., vol. 26, pp. 638–647, 2018, doi: 10.2197/ipsjjip.26.638.
16. M. Holmstrom, D. Liu, and C. Vo., “Machine Learning Applied to Weather Forecasting,” Meteorol. Appl, vol. 10, no. 2, pp. 1–5, 2016.
17. W. Lu, S. Kawasaki, and J. Sakuma, “Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data,” Cryptol. ePrint Arch., 2016, doi: 10.14722/ndss.2017.23119.
18. R. Hall, S. E. Fienberg, and Y. Nardi, “Secure multiple linear regression based on homomorphic encryption,” J. Off. Stat., vol. 27, no. 4, pp. 669–691, 2011.
19. “Gold Dataset.” https://codeload.github.com/bhargav-joshi/Gold-Price-Prediction-with-Python/zip/refs/heads/main.
20. S. Rong and Z. Bao-Wen, “The research of regression model in machine learning field,” MATEC Web Conf., vol. 176, pp. 8–11, 2018, doi: 10.1051/matecconf/201817601033.
21. “Linear Regresion Libraries.” https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#:~:text=The best possible score is,of shape (n_samples%2C n_features).
22. S. Mittal and S. Singh, “Data Analytics over Encrypted Data from Fully Homomorphic Encryption,” 2023 IEEE 4th Annu. Flagsh. India Counc. Int. Subsections Conf. Comput. Intell. Learn. Syst. INDISCON 2023, pp. 1–5, 2023, doi: 10.1109/INDISCON58499.2023.10269997.
23. S. Mittal, “Fully homomorphic encryption-based optimal key encryption for privacy preservation in the cloud sector,” J. Inf. Secur. Appl., vol. 91, no. April, p. 104048, 2025, doi: 10.1016/j.jisa.2025.104048.