Privacy-Preserving Sentiment Analysis with Fully Homomorphic Encryption

HUNJAE "TIMOTHY" LEE, Southern Methodist University, USA

ACM Reference Format:
Hunjae "Timothy" Lee. 2018. Privacy-Preserving Sentiment Analysis with Fully Homomorphic Encryption. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In this paper, we demonstrate a way of performing privacy-preserving sentiment analysis using Fully Homomorphic Encryption (FHE). Sentiment analysis is a machine learning technique that extracts subjective information from textual data. As such, sentiment analysis can be performed on customer reviews, emails, and more to extract and analyze sentiment information of said data. Although a powerful tool, sentiment analysis in its current form, like other machine learning models, requires transparency and availability of data. This means that in industries and fields where data privacy is paramount, they cannot rely on third-party supercomputers or cloud architectures to perform sentiment analysis. This is where FHE comes in. FHE, as proven and demonstrated by Craig Gentry in 2009, allows for computation on encrypted data without the need for decrypting it first [Gentry 2009]. In other words, sentiment analysis performed with FHE will ensure data privacy even if the sentiment analysis computations are being performed by untrusted third parties.

1.1 Motivation

Privacy of data is becoming an ever-important topic for computer scientists and consumers alike. In the field of machine learning in particular, data privacy has largely been nearly impossible to accomplish. As a result, much focus of FHE research has been on privacy-preserving machine learning. However, the field of FHE is still very new with practical and some theoretical limitations of FHE largely preventing practical implementation and adoption of FHE in machine learning. Despite this, FHE maintains to be a robust and fast-growing field with progress being made every year toward practical privacy-preserving machine learning. The motivation for this paper is to add to the ever-growing research of FHE by demonstrating sentiment analysis evaluation on homomorphically encrypted data. The aim of this paper is not to introduce fundamental mathematical improvements to FHE, but rather to apply existing FHE technologies in order to investigate the feasibility and practicability of FHE in the field of sentiment analysis. Likewise, this paper also aims to unearth and investigate areas of FHE that pose a challenge toward practical implementation in the context of sentiment analysis and machine learning.

Author's address: Hunjae "Timothy" Lee, hunjael@smu.edu, Southern Methodist University, Dallas, Texas, USA, 75205.

0730-0301/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

1.2 Sentiment Analysis

Sentiment analysis is a Natural Language Processing (NLP) technique within the domain of machine learning that aims to extract subjective information from textual data. Sentiments can be classified into binary or multiclass categories as well as regressed into continuous values. Sentiment analysis can be used in a variety of fields such as customer-feedback analysis, political analysis, and customer service analysis.

1.3 Fully Homomorphic Encryption

Homomorphic encryption is a type of encryption scheme that allows for arithmetic and logical computations to be performed on encrypted data without the need for decryption. This can be achieved between two sets of encrypted inputs as well as one set each of encrypted and plaintext inputs. While there are many variations of homomorphic encryption such as Partially Homomorphic Encryption (PHE) and Somewhat Homomorphic Encryption (SHE), this paper uses FHE as it is considered the most powerful form of homomorphic encryption.

2 RELATED WORK

Since Gentry's breakthrough at Stanford, research in FHE and homomorphic encryption in general have seen an explosive growth. Within FHE, there are two main approaches when dealing with noise that accumulates from encrypted arithmetic operations: bootstrapped and leveled. Bootstrapped schemes such as Zama's Concrete library partially decrypts and re-encrypts the ciphertexts periodically to reduce noise and allow for unlimited number of encrypted computations at the steep cost of speed and efficiency [Zam [n. d.]]. Leveled schemes such as the CKKS scheme on the other hand allow for limited number of encrypted computations but maintains higher speed of computation [Cheon et al. 2016]. We take the leveled approach in this paper as bootstrapped schemes in their current state are too computationally taxing for large inputs. Specifically, we use TenSEAL, a Microsoft SEAL variant that supports matrix and tensor operations on real numbers [Benaissa et al. 2021].

2.1 FHE in Machine Learning

With the emergence of FHE libraries that support tensors and matrix operations, much attention in FHE research have shifted toward privacy-preserving machine learning. Groups like Zama and Openmined [ope [n. d.]] have demonstrated deep and convolutional neural network inference in recent years. Zama's whitepaper on deep neural network inference with FHE [Chillotti et al. 2021] has even shown that the depth of a neural network is not a limiting

factor for FHE inference. However, many areas of machine learning are still infeasible and challenging to achieve with FHE. Back propagation and gradient calculations for example, are too computationally expensive for FHE in its current state, making training of machine learning models a difficult endeavor. Another challenge for FHE is activation function calculation as the non-linearity of most activation functions like the sigmoid function make it challenging to be computed in FHE. However, recent works have demonstrated the viability of polynomial approximations of activation functions that allow for FHE computation with near equivalent performance [Chiang 2022].

2.2 FHE in Natural Language Processing

FHE in the NLP domain has drawn interest as of late particularly in areas of banking services and SPAM detection. A recent paper from KB Bank has investigated homomorphic inference of text embedding similarity with an approximate cosine similarity function [Kim et al. [n. d.]]. Another work demonstrated the feasibility of SPAM detection with homomorphic inference [Badawi et al. 2020]. While both of these recent works provide positive contribution, FHE in NLP is still a field in its infancy and much work needs to be done for FHE to be integrated practically into the many facets of NLP. As an addition to the contribution made by these two works and more, this paper aims to demonstrate sentiment analysis on word embeddings using GloVe [Pennington [n. d.]] and Bing Liu's repository of positive and negative words [Liu [n. d.]]. By performing sentiment analysis, which is one of many facets of NLP not explicitly covered by these previous works, this paper aims to investigate the feasibility of FHE within the context of sentiment analysis and NLP at large from a different angle and provide our findings.

3 METHODS

In this paper, we use GloVe as our 300-dimensional word embedding and Bing Liu's repository of words with positive and negative sentiments to train a logistic regression model. The logistic regression model written in pytorch [Paszke et al. 2019], is first implemented in plaintext to serve as a baseline performance metric as well as to later be used as a building-block for our encrypted model. The performance of the plaintext logistic regression model is then compared with the encrypted version. Because the aim of this paper is to investigate feasibility of FHE adoption into sentiment analysis, we are not necessarily aiming for the highest possible accuracy and performance in the plaintext model, but rather consistency between both plaintext and encrypted models. In order to build an encrypted version of a plaintext logistic regression model capable of performing sentiment analysis with the GloVe word embedding, steps are taken to mitigate inherent challenges of FHE and to enhance performance. These steps are outlined in the following subsections along with an overview of the general architecture of the models.

3.1 Architecture for Sentiment Analysis

In this study, we use GloVe 6B and Bing Liu's compiled words with positive and negative sentiments. Even though we are using the smaller version of GloVe at the cost of performance, it is acceptable as long as the same version is used for both plaintext and encrypted models since our objective is to compare the performance between them. Once the embedding is loaded, Bing Liu's sentiment lexicon is parsed and used to train the plaintext logistic regression model. The high-level overview of the encrypted architecture is akin to a client-server interaction in web development. First, the client (in our case the data owner), encrypts sensitive data and sends it to a server (untrusted third-party computing service). Second, the server performs evaluation on encrypted data and sends back the encrypted results. Lastly, the client decrypts the results and either uses them as is or performs further operations if applicable. This architecture relies on a model pre-trained in plaintext even though it is later deployed for encrypted evaluation. Therefore, this architecture is suitable in cases where there is enough data without privacy issues available for training but needs to be deployed in areas where incoming data for evaluation does have privacy concerns. As an intuitive example, this type of architecture can be used for SPAM detection in company emails where there are enough old and irrelevant emails in plaintext to train a model but real-time SPAM detection on current employees' emails needs to be evaluated with FHE to protect their privacy.

3.2 Logistic Regression in Plaintext

The base logistic regression model in plaintext is taken largely from TenSEAL's documentation [OpenMined 2021]. It uses a Binary Cross Entropy (BCE) loss function (1) and back propagates with Stochastic Gradient Descent (SGD) (2). There isn't a great deal of mathematical intuition or motivation behind the choice to use BCE and SGD. These are simply functions that are standard and often used with logistic regression and as long as a reasonable accuracy is achieved using them, the plaintext logistic regression can successfully be used as a base for the encrypted version.

$$-(y \log(p) + (1 - y) \log(1 - p)) \tag{1}$$

$$\mathbf{w} \leftarrow \mathbf{w} + \eta\Big((y^{(i)} - g(\mathbf{w}^T \mathbf{x}^{(i)}))\mathbf{x}^{(i)} - 2C \cdot \mathbf{w}\Big), \text{ where } i \in M \tag{2}$$

The training algorithm takes around 4500 epochs with a learning rate of 0.001 to converge. The validation accuracy for this plaintext model is at around 91% which is more than enough to compare and test against the encrypted model.

3.3 Logistic Regression Evaluation with FHE

The encrypted logistic regression model is a feedforward model that uses trained weights from the plaintext model. The feedforward layer computes matrix operations with weights and biases against encrypted data input and outputs encrypted results to be returned. Note that the weights and biases are in plaintext; many FHE libraries including TenSEAL support plaintext-ciphertext computation. These results are then evaluated by decrypting them and comparing their accuracy against the plaintext model's evaluation. Accuracy of a model is the primary performance metric of choice in this paper. Though other performance metrics such as f1 and precision scores as well as further statistical comparisons are of great importance in gauging the validity and true performance of machine learning models, we have determined that for the scope of

this paper, accuracy of the encrypted model as compared to that of the plaintext model is a good enough starting point to open up the discussion for feasibility of FHE in sentiment analysis.

3.4 Sigmoid Approximation

Activation functions are generally difficult to achieve with FHE due to their non-linearities. Though theoretically possible, the complex mathematical operations that are required to compute non-linear functions make it completely impractical in practice. In this paper, we use John Chiang's polynomial approximation of the sigmoid to work around the non-linearity of the traditional sigmoid activation function [Chiang 2022]. Chiang uses the least square of the gradient difference between the polynomial approximation and the activation function to "fit" a low degree polynomial function to a sigmoid function within the range [-8,+8] as seen in equation 3.

$$\int_a^b (f(x) - p(x))^2 + (f\prime(x) - p\prime(x))^2 dx \tag{3}$$

This results in equation 4, a near-equivalent, practically computable alternative to a traditional sigmoid so long as the range of data is stable.

$$\sigma(x) = 1.1110537229 + 0.5 * x + 0.054235537 * x^2 \tag{4}$$

4 RESULTS AND CONCLUSION

First, before anything else can be achieved, feasibility of FHE evaluation of a model must be verified. Feasibility of encrypted evaluation in this context is a relatively broad term as the state of FHE and its limitations must be considered. Therefore, it is expected that encrypted evaluation will not be equivalent to plaintext model performance. Though accuracy of a model is not necessarily a holistic representation of its performance, it nonetheless gives an adequate picture on its feasibility and shows that encrypted evaluation is working as intended. As shown in Table 1, encrypted evaluation of our logistic regression model for sentiment analysis does perform as intended and results in comparable accuracy compared to that of the plaintext model across epochs. It can also be seen that as the number of epochs grow, encrypted model's accuracy slowly catches up to that of the plaintext model albeit still falling short by around 1.2% at the end. Seeing model accuracy of around 90% against a completely encrypted test dataset is more than enough to determine that FHE evaluation of our model is feasible and can be used for further analysis.

Once the feasibility of our encrypted evaluation model is verified, a few practical tests are done to see the sentiment scores for varying imdb reviews on the recent 2023 Mario movie for an intuitive look at how our models perform [IMD [n. d.]]. The results for this are in Table 2 where the first entry is a positive review, second entry a neutral review slightly more negative than positive, and lastly a negative review with no room for positive sentiment. Each entry has plaintext and FHE sentiment scores attached where positive scores denote positive sentiment prediction and negative scores denote negative sentiment prediction. Accuracy scores aside, when used against real, practical inputs, both models show signs of imperfection. For example, they both seem to struggle with differentiating between neutral and negative reviews. However, this is not completely unexpected as our logistic regression models were never fine-tuned for accuracy. What is of much greater interest is seeing whether both plaintext and FHE models are consistent in identifying positive, neutral, and negative sentiments. Even though the values are different, both plaintext and encrypted evaluations follow the same trend with the encrypted evaluation being more biased on the negatives. Because of the black-box nature of FHE computations, it is not always possible to get a comprehensive understanding on why some encrypted computations deviate slightly from expected values. Often, the noise that accumulates from encrypted arithmetic operations can unpredictably influence the results even if said noise is not big enough to break the ciphertext. But broadly speaking, our encrypted evaluation can demonstrably differentiate between positive and negative sentiments in sentences.

In our paper, we have added to a growing list of FHE research and investigations by demonstrating feasibility of sentiment analysis in FHE with GloVe word embeddings as well as outlining an applicable client-server architecture for secure and streamlined computation. Though outside the scope of this investigation, results from this paper also opens opportunities to take our encrypted evaluation model further and introduce encrypted fine-tuning, further reducing the computational load on the client-side and improving the encrypted model without fully training an encrypted model from scratch. However, the real purpose of this paper is the fact that an investigation into sentiment analysis with FHE is well-positioned to serve as an entry point into larger, more complex NLP tasks. Sentiment analysis can be re-fitted for SPAM detection in emails, personality recognition, political sentiment, and more. Overall, FHE as a field of research is still in its early stages and investigating and proving feasibility in any aspect of machine learning are crucial to the progress of FHE. As data become ubiquitous and the digital age becomes a part of everyday life, so too grows the need for data privacy. Practical privacy-preserving machine learning with FHE is a necessary step towards a future where breakthrough data analysis and machine learning applications are handled without violating the privacy of personal, medical, or corporate entities. We plan for an ongoing contribution to FHE research in these exciting new areas and hope to see others making progress toward a more practical FHE.

ACKNOWLEDGMENTS

This is an ongoing research by members of Human and Machine Intelligence Game Lab at SMU led by Dr. Corey Clark.

Table 1. Accuracy against Epochs

Epochs	Plaintext Accuracy	FHE Accuracy
500	79.72%	74.36%
1000	87.01%	83.25%
1500	87.82%	85.18%
2000	89.42%	86.70%
2500	89.82%	87.50%
3000	90.38%	87.91%
3500	90.14%	88.38%
4000	91.19%	90.30%
4500	91.26%	90.06%

Table 2. IMDB Movie Review Sentiment Scores

Review	Plaintext Score	FHE Score
This movie felt like a beautifully animated amusement park ride	0.42	-0.15
Not high art or anything, but it ticks off almost everything for what Mario should be at least	-0.10	-0.63
easily one of the worst movie Illumination produced so far	-0.07	-0.62

REFERENCES

[n. d.]. https://www.zama.ai/company

[n. d.]. https://github.com/OpenMined

[n. d.]. https://m.imdb.com/title/tt6718170/reviews?ref_=tt_urv

Ahmad Al Badawi, Louie Hoang, Chan Fook Mun, Kim Laine, and Khin Mi Aung. 2020. Privft: Private and fast text classification with Homomorphic encryption. *IEEE Access* 8 (2020), 226544–226556. https://doi.org/10.1109/access.2020.3045465

Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. 2021. TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption. arXiv:2104.03152 [cs.CR]

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2016. Homomorphic Encryption for Arithmetic of Approximate Numbers. Cryptology ePrint Archive, Paper 2016/421. https://eprint.iacr.org/2016/421 https://eprint.iacr.org/2016/421.

John Chiang. 2022. On Polynomial Approximation of Activation Function. arXiv:2202.00004 [cs.LG]

Ilaria Chillotti, Marc Joye, and Pascal Paillier. 2021. Programmable bootstrapping enables efficient homomorphic inference of Deep Neural Networks. *Lecture Notes in Computer Science* (2021), 1–19. https://doi.org/10.1007/978-3-030-78086-9_1

Craig Gentry. 2009. *A fully homomorphic encryption scheme.* Ph. D. Dissertation. Stanford University. crypto.stanford.edu/craig.

Donggyu Kim, Garam Lee, and Sugwoo Oh. [n. d.]. *Toward Privacy-preserving Text Embedding Similarity with Homomorphic Encryption* ([n. d.]).

Bing Liu. [n. d.]. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

OpenMined. 2021. TenSEAL: A Library for Homomorphic Encryption Operations on PyTorch. https://github.com/OpenMined/TenSEAL. Accessed: May 8, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* Curran Associates Inc., Red Hook, NY, USA.

Jeffrey Pennington. [n. d.]. https://nlp.stanford.edu/projects/glove/