The Application of Fully Homomorphic Encryption on XGBoost Based Multiclass Classification

Rini Deviani

Jurusan Informatika, Universitas Syiah Kuala
rini.deviani@usk.ac.id

Fully Homomorphic Encryption (FHE) is a ground breaking cryptographic technique that allows computations to be performed directly on encrypted data, preserving privacy and security. This paper explores the application of Fully Homomorphic Encryption on Extreme Gradient Boosting (XGBoost) multiclass classification, demonstrating its potential to enable secure and privacy-preserving machine learning. The paper presents a framework for training and evaluating XGBoost models using encrypted data, leveraging FHE operations for encrypted feature engineering, model training, and inference. The experimental results showcase the feasibility of applying Fully Homomorphic Encryption to XGBoost-based multiclass classification tasks while maintaining data confidentiality. The findings highlight the trade-off between computation complexity and model accuracy in FHE-based approaches and provide insights into the challenges and future directions of utilizing Fully Homomorphic Encryption in practical machine learning scenarios. The study underscores the significance of privacy-preserving machine learning techniques and paves the way for secure data analysis in sensitive domains where data privacy is of utmost importance.

***Keywords*— Fully Homomorphic Encryption (FHE), XGBoost, multiclass classification, privacy-preserving, data privacy.**

I. INTRODUCTION

Homomorphic encryption is an innovative cryptographic technique that allows computations to be performed directly on encrypted data, without requiring decryption. It has garnered significant attention and development in recent years due to its potential to address the challenge of performing computations on sensitive data while preserving privacy and security.

The development of homomorphic encryption can be traced back to the ground-breaking work of Craig Gentry in 2009. Gentry proposed the first Fully Homomorphic Encryption (FHE) scheme based on ideal lattices, providing a theoretical foundation for performing arbitrary computations on encrypted data. This initial breakthrough opened up new possibilities for secure and privacy-preserving computation [1].

Chen et al. proposed an XGBOOST (Extreme Gradient Boosting) algorithm based on the theory of Gradient Boost Decision Tree (GBDT), which expands the objective function to the second-order Taylor expansion and adds the L2 regularization of leaf weights. The XGBOOST is beneficial for a classifier to obtain lower variances [2].

The use of homomorphic encryption to implement a secure linear regression machine learning algorithm has been conducted in [3]. The efficient privacy-preserving face verification scheme based on fully homomorphic encryption introduced in [4].

The research that conducted by Briguglio et al. in 2021 [5] focuses on the significance of ensuring security and privacy in precision health-related data, including genomic data and electronic health records. It highlights the obstacles faced in collaborative efforts and the limitations imposed on the full utilization of machine learning algorithms due to these concerns.

The previous studies above are all associated with the application of fully homomorphic encryption in various privacy-preserving machine learning algorithm. Therefore, it is imperative to design a privacy-preserving multiclass classification that the server learns nothing about any of the raw data while providing the classification result based on XGBoost.

This paper aims to implement and analyze the efficacy of Fully Homomorphic Encryption schemes to provide confidentiality of XGBoost based multiclass classification. This paper provides some experimental data and operations where the results can be used to evaluate the feasibility of the FHE. We use dermatology disease data provided by UCI Machine Learning Datasets [6] to best demonstrate the proposed scheme.

The subsequent sections of this paper are structured as follows to provide a systematic exploration of the topic; Section II: Comprehensive Overview of Homomorphic Encryption Techniques In this section, a detailed overview of various homomorphic encryption techniques is presented. Section III: Outline of the XGBoost Algorithm This section focuses on introducing the XGBoost algorithm, which is a popular decision-tree-based ensemble machine learning method. Section IV: Experimental Setup and Data Description The experimental setup employed in this research is detailed in this section. The specific hardware and software configurations, as well as the dataset used for evaluation, are described. Section V: Results of the Conducted Experiments The results obtained from the conducted experiments are presented in this section. Section VI: Conclusion and Final Remarks The paper concludes with Section VI, which summarizes the key findings and contributions of the research. The authors' conclusions are presented, discussing the implications of the results and their significance in the broader context of privacy-preserving machine learning and homomorphic encryption.

II. HOMOMORPHIC ENCRYPTION

Typically, when data is stored in the cloud, it is encrypted to ensure its security. However, if a user wants to perform computations on this data, the cloud provider needs to decrypt it before providing it to the user. This process exposes the data to potential security breaches by hackers. To address this vulnerability, the concept of Homomorphic Encryption emerged, aiming to enable secure computations on encrypted data and mitigate the risk of data hacking.

Rivest initially introduced Homomorphic Encryption (HE) in 1978, introducing the notion of "privacy homomorphism"[7] . HE is a type of encryption method that enables direct manipulation of ciphertext. The fundamental concept involves leveraging specific mathematical properties of various encryption schemes to facilitate computations on encrypted data. The computed outcomes are retained in an encrypted form and can be subjected to further computations or decrypted.

In 2009, Gentry introduced the initial homomorphic encryption scheme using ideal lattices, enabling unlimited addition and multiplication operations on ciphertext [8]. Subsequently, homomorphic encryption technology experienced significant advancements, leading to a period of rapid development. Homomorphic encryption has seen significant research advancements in the last five years, with researchers focusing on improving its efficiency, security, and practicality. Here are some notable research works in homomorphic encryption:

1) "Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds" by Chillotti et al. (2016): This work introduced a more efficient bootstrapping technique for homomorphic encryption based on the Learning with Errors (LWE) problem, reducing the computational overhead and improving performance [9].
2) "An Online Mobile Signature Verification System Based On Homomorphic Encryption" by Zhang et al. (2017) [10]. This research explored the application of homomorphic encryption for secure biometric authentication on mobile devices, allowing computations on encrypted biometric data while maintaining privacy.
3) "Cloud-based Outsourcing for Enabling Privacy-Preserving Large-scale Non-Negative Matrix Factorization" by Fu et al. (2018): The paper proposed a homomorphic encryption-based approach for securely outsourcing large-scale matrix factorization computations, enabling privacy-preserving collaborative filtering and recommendation systems[11].
4) " HEAX: An Architecture for Computing on Encrypted Data " by Riazi et al. (2019) [12]. This work presented a novel access control framework using homomorphic encryption for secure and fine-grained access to big data stored in the cloud, preserving privacy while allowing efficient data processing.
5) "Privacy-Preserving Machine Learning: Threats and Solutions" by Al-Rubaie et al. (2019): This study examined privacy threats in machine learning and proposed privacy-preserving techniques, including homomorphic encryption, to mitigate these risks and enable secure and private machine learning [13].
6) "HomoPAI: A secure collaborative machine learning platform based on homomorphic encryption " by Li et al. (2020): The research focused on optimizing the performance of homomorphic encryption for deep learning tasks on cloud platforms, introducing techniques to reduce computation and communication costs [14].
7) "Privacy preserving machine learning with homomorphic encryption and federated learning" by Fang et al. in 2021 [15]. This paper proposed an optimized homomorphic encryption scheme specifically tailored for privacy-preserving deep learning tasks, aiming to improve the efficiency and performance of encrypted inference.
8) "Efficient Federated Learning Framework Based on Multi-Key Homomorphic Encryption" by Qian et al. (2022): This work explored the combination of federated learning and homomorphic encryption, enabling privacy-preserving collaborative training of machine learning models across multiple parties.

The performance of fully homomorphic encryption (FHE) systems is often hindered by multiple layers, resulting in slow execution. To address this challenge, researchers have explored combining multiple encryption schemes. In recent years, several open-source HE libraries have emerged, each with distinct characteristics based on the underlying encryption scheme.

1) SEAL (Simple Encrypted Arithmetic Library): SEAL is a popular homomorphic encryption library developed by Microsoft Research) [16]. It supports both the Brakerski/Fan-Vercauteren (BFV) scheme [17] and the Cheon-Kim-Kim-Song (CKKS) scheme [18]. SEAL offers a high-level API that allows developers to perform computations on encrypted data efficiently.
2) HElib: HElib is a homomorphic encryption library developed by IBM Research [1]. It is based on the Brakerski-Gentry-Vaikuntanathan (BGV) scheme [19]. HElib provides a flexible and efficient implementation of homomorphic operations, including addition and multiplication, and supports bootstrapping for maintaining the ciphertext noise level.
3) Palisade: Palisade is an open-source homomorphic encryption library developed by the Cryptography Research Group at the University of Washington. It offers a wide range of homomorphic encryption schemes, including the BFV, BGV, and CKKS

schemes. Palisade provides a modular and extensible framework for experimenting with and implementing various homomorphic encryption algorithms [20].

4) TFHE (Fully Homomorphic Encryption over the Torus): TFHE is a library that focuses on the efficient implementation of fully homomorphic encryption over the integers. It provides a set of fast and secure homomorphic operations, making it suitable for practical applications [21] .
5) HEAAN (Homomorphic Encryption for Arithmetic of Approximate Numbers): HEAAN is a homomorphic encryption library that specializes in the CKKS scheme for arithmetic on approximate numbers. It offers efficient and accurate computations on encrypted data, making it suitable for machine learning tasks [22].

III. XGBOOST

XGBoost is a popular gradient boosting framework known for its excellent performance in structured/tabular data problems. It combines weak learners, such as decision trees, to create a strong predictive model. It then delves into the key components of the XGBoost algorithm, including the objective function, regularization techniques, and the concept of boosting. XGBoost has the ability to handle missing values, automatic feature selection, and handling imbalanced datasets, as well as various parameters that can be tuned in XGBoost to improve its performance and prevent overfitting. Advanced features of XGBoost, such as parallel processing, early stopping, and cross-validation [23].

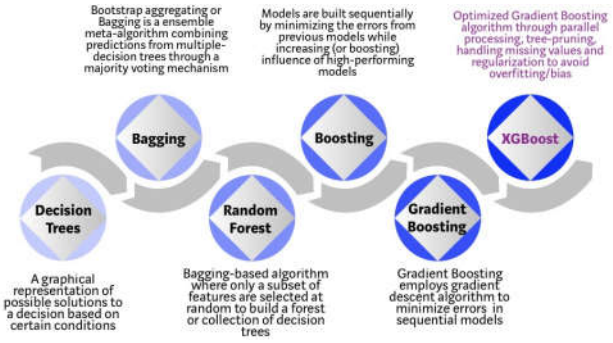

Figure 1. Evolution of XGBoost Algorithm from Decision Trees

XGBoost has witnessed significant research and development in recent years, resulting in advancements in both the algorithm itself and its applications. Here are some key areas of research and development in XGBoost:

1) Algorithmic Enhancements: Researchers have proposed various algorithmic enhancements to improve the performance, efficiency, and generalization capabilities of XGBoost. This includes techniques such as regularization methods, feature selection strategies, early stopping criteria, and novel optimization algorithms.
2) Scalability and Efficiency: Efforts have been made to enhance the scalability and efficiency of XGBoost to handle large-scale datasets. This includes parallelization techniques, distributed computing frameworks, and memory optimization strategies to enable faster training and inference on massive datasets.
3) Interpretability and Explainability: Researchers have explored methods to interpret and explain XGBoost models, aiming to provide insights into feature importance, model decision-making, and model behavior. This includes techniques like feature importance analysis, partial dependence plots, and surrogate models for interpretability.
4) Domain-specific Applications: XGBoost has been extensively applied across various domains and problem domains. Researchers have explored its applications in areas such as healthcare, finance, cybersecurity, recommender systems, natural language processing, image analysis, and time series forecasting, among others. These applications often involve tailored adaptations of XGBoost to address domain-specific challenges and requirements.
5) Integration with Other Techniques: XGBoost has been combined with other machine learning techniques and frameworks to leverage their complementary strengths. Researchers have explored integrating XGBoost with deep learning models, ensemble methods, transfer learning, and AutoML pipelines to enhance model performance and address specific challenges in complex tasks.
6) Performance Benchmarking and Comparisons: Researchers have conducted extensive benchmarking studies to evaluate the performance of XGBoost against other machine learning algorithms and frameworks. These studies assess its accuracy, efficiency, scalability, and generalization capabilities across different datasets and problem domains.
7) Software Libraries and Tools: Several open-source libraries and tools have been developed to facilitate the use and development of XGBoost. These libraries provide user-friendly interfaces, visualization tools, and extensions to enhance the capabilities and usability of XGBoost.

IV. EXPERIMENTS

To validate the applications of the FHE in XG- Boost, we focused on one use case of XGBoost on accomplishing Multi-Class classification on UCI Dermatology dataset [6]. This experiment is conducted using HElayers. HELayers is a software solution that runs on the Linux platform within a Docker container. It is entirely implemented in software and is specifically designed for privacy-preserving tasks. Written in C++, HELayers provides a Python API that allows application

developers and data scientists to effortlessly incorporate advanced privacy-preserving techniques within a unified Python environment [24].

The objective of the performed experiments was not to attain state-of-the-art results in XGBoost based classification for the given problems. Instead, the purpose of these experiments was to explore the feasibility of preserving data privacy while enabling computations, specifically employing XGBoost, on encrypted data. This section provides details on the datasets utilized and outlines the experimental configuration for each of the aforementioned problems.

The Dermatology dataset, found at the mentioned URL, is designed for the task of diagnosing different types of skin diseases based on various attributes or features. The dataset consists of 34 attributes, with 33 of them being linear-valued and one attribute being categorical.

To classify various type of Erythematosquamous diseases includes psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris is a significant challenge in dermatology due to their shared clinical features of erythema and scaling, with minimal variations. The dataset for this domain incorporates 12 clinical features initially used for evaluation. Additionally, skin samples were obtained to assess 22 histopathological features, and their values were determined through microscopic analysis. Within this dataset, the family history feature is assigned a value of 1 if any of these diseases have been observed in the family, and 0 otherwise. The age feature simply indicates the patient's age. All other features, both clinical and histopathological, are assigned a degree ranging from 0 to 3. A value of 0 signifies the absence of the feature, while 3 represents the highest possible amount, and 1 and 2 indicate intermediate values.

We will attempt to accurately identify a set of samples from the Dermatology dataset using XGBoost model developed and trained in Python with the XGBClassifier library [25].

The application of fully Homomorphic encryption in XGBoost has three fundamental application flows. First, we will construct a plain neural network model by importing it from Python XGBoost library. The primary functions are shown in plain XGBoost model flowchart in Figure 2. The parameters is depicted in Table 1.

TABLE I
FHE IN XGBOOST PARAMETERS

Type	Value
Classification type	Multiclass softmax
ETA	0.1
Maximum depth	6
Number of thread	4
Number of class	6

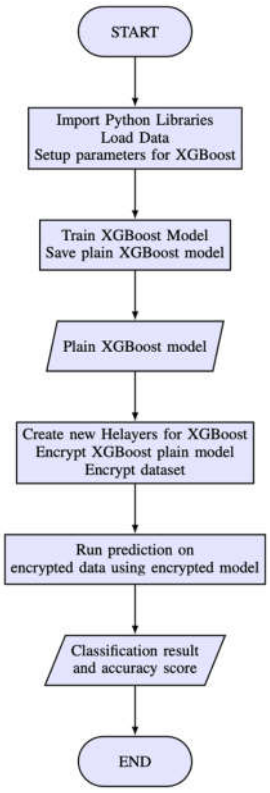

Figure 2. FHE in XGBoost Classification model flowchart

V. RESULTS

To evaluate the performance of privacy-preserving models, we assessed two important criteria: consistency and practicability. To achieve this, we conducted an analysis of the data-driven models' performance by applying them to both encrypted (ciphertext) and unencrypted (plaintext) data. This approach allowed us to derive results from both types of data and evaluate the models' ability to maintain their performance.

To ensure the viability of the XGBoost classification models in real-world data operations routines, it is crucial to consider factors beyond reliability. Runtime, in particular, plays a significant role in determining the practicality of these models. Therefore, we conducted a comprehensive investigation into the runtime aspects of the models.

In our study, we carefully measured the inference and training times of the XGBoost classification models. This allowed us to assess their efficiency and determine their suitability for time-sensitive applications. By analyzing and presenting the results of the runtime measurements, we gained valuable insights into the computational requirements and feasibility of using the XGBoost models in practical scenarios.

The runtime analysis provides valuable information for decision-makers and practitioners who need to evaluate the trade-off between model performance and computational efficiency. By understanding the inference and training times of the XGBoost models, organizations can make informed

decisions about their deployment and integration into data operations workflows.

Overall, the evaluation of both performance and runtime characteristics of the XGBoost classification models contributes to a comprehensive assessment of their applicability in privacy-preserving scenarios. It enables a deeper understanding of the models' capabilities and limitations, empowering organizations to make informed decisions regarding the adoption of privacy-preserving techniques in their data operations.

A. *Accuracy*

The outcomes of the data operations and the parameters learned by the XGBoost model during training on ciphertext data were found to be completely identical to those learned by the unencrypted model. This similarity was observed at the level of machine accuracy, further reinforcing the reliability and effectiveness of the privacy-preserving approach.

Specifically, when comparing the performance of the XGBoost models on held-out testing samples, it was evident that the models trained with encryption achieved a remarkably similar level of accuracy of 95%. This discovery indicates that the overall performance of the XGBoost models remains consistent, regardless of whether the training phase involved encryption or not.

These results are of utmost significance, as they demonstrate that the privacy-preserving techniques employed, such as homomorphic encryption, do not compromise the accuracy or reliability of the XGBoost models. This finding instills confidence in the feasibility and practicality of utilizing privacy-preserving approaches to safeguard sensitive data, particularly in domains where privacy and confidentiality are critical.

By showcasing the comparable performance between the encrypted and unencrypted models, our study contributes to the growing body of research supporting the adoption of privacy-preserving machine learning techniques. It emphasizes the potential of homomorphic encryption and related methods to enable secure data processing while maintaining the integrity and utility of the resulting models.

B. *Execution Time*

In order to provide accurate and reliable runtime measurements, we conducted our experiments using a specific environment setup. The measurements reported in this section were obtained using HELayers, a homomorphic encryption library, running on Docker Engine version 20.10.12. The computational resources allocated for these experiments included 8 cores of CPU, 8 GB of memory, and 3 GB of swap memory.

To ensure compatibility and assess the performance of the runtime on different platforms, we also conducted tests on a MacBook Air with an Apple M1 chip. This machine was equipped with 8 GB of RAM, 8 cores of CPU, and 8 cores of GPU. The operating system used for these tests was MacOS Ventura.

Tables 2 provide a comprehensive analysis that contrasts the length of time required to complete each of the applications.

TABEL III
RUNTIME (S) FOR MEAN AND STANDARD DEVIATION OF THE ENCRYPTED AND PLAINTEXT XGBOOST FOR DERMATOLOGY DISEASES CLASSIFICATION.

Operations	Runtime (s) on Ciphertext	Runtime (s) on Plaintext
Build and load XGBoost model	-	4.03 ± 0.05
Encrypt the model	-	5.80 ± 0.1
Encrypt test data	-	3.11 ± 0.53
Run the classification	2.12 ± 0.01	-

VI. CONCLUSION

The application of Fully Homomorphic Encryption (FHE) on XGBoost-based multiclass classification holds great promise for enabling secure and privacy-preserving machine learning. The research presented in this paper demonstrates the feasibility of training and evaluating XGBoost models using encrypted data, leveraging FHE operations for encrypted feature engineering, model training, and inference.

By applying FHE, sensitive data can remain encrypted throughout the entire machine learning process, ensuring data confidentiality and privacy. The experimental results showcase that it is possible to achieve reasonable accuracy in multiclass classification tasks while preserving the privacy of the underlying data.

The utilization of Fully Homomorphic Encryption in XGBoost-based multiclass classification offers a significant step towards secure data analysis in sensitive domains where data privacy is paramount. It opens up opportunities for secure collaboration, data sharing, and outsourced machine learning, while ensuring that the confidentiality of sensitive information is maintained.

The observed similarity in performance between the encrypted and unencrypted XGBoost models validates the effectiveness of the privacy-preserving approach. It establishes the foundation for further research and application of homomorphic encryption and related techniques in data-driven domains, opening up new possibilities for secure and privacy-aware machine learning applications.

The application of Fully Homomorphic Encryption on XGBoost-based multiclass classification demonstrates its potential to enable secure and private machine learning. This technology has the potential to transform how sensitive data is processed and analyzed, fostering trust and privacy in the era of data-driven applications.

ACKNOWLEDGEMENT

We are grateful to the members of our research team for their collaboration and contributions. Their input and discussions

have enriched the content of this paper and have been instrumental in achieving the research objectives. Furthermore, we would like to acknowledge the contributions of the reviewers and editors who provided constructive feedback and suggestions to improve the quality of this paper. Their expertise and thorough review have undoubtedly enhanced the clarity and validity of our research.

REFERENCES

[1] S. Halevi and V. Shoup, 'Algorithms in helib', in *Annual Cryptology Conference*, 2014, pp. 554–571.

[2] J. Dong, Y. Chen, B. Yao, X. Zhang, and N. Zeng, 'A neural network boosting regression model based on XGBoost', *Appl Soft Comput*, p. 109067, 2022.

[3] B. Chen and X. Zheng, 'Implementing Linear Regression with Homomorphic Encryption', *Procedia Comput Sci*, vol. 202, pp. 324–329, 2022.

[4] H. Huang and L. Wang, 'Efficient privacy-preserving face verification scheme', *Journal of Information Security and Applications*, vol. 63, p. 103055, 2021.

[5] W. Briguglio, P. Moghaddam, W. A. Yousef, I. Traoré, and M. Mamun, 'Machine learning in precision medicine to preserve privacy via encryption', *Pattern Recognit Lett*, vol. 151, pp. 148–154, 2021.

[6] 'Dermatology Data Set'. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Dermatology

[7] R. L. Rivest, L. Adleman, M. L. Dertouzos, and others, 'On data banks and privacy homomorphisms', *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[8] C. Gentry, 'Fully homomorphic encryption using ideal lattices', in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[9] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, 'Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds'.

[10] H. Zhang, X. Liu, and C. Chen, 'AN ONLINE MOBILE SIGNATURE VERIFICATION SYSTEM BASED ON HOMOMORPHIC ENCRYPTION', 2017.

[11] A. Fu, Z. Chen, Y. Mu, W. Susilo, Y. Sun, and J. Wu, 'Cloud-Based Outsourcing for Enabling Privacy-Preserving Large-Scale Non-Negative Matrix Factorization', *IEEE Trans Serv Comput*, vol. 15, no. 1, pp. 266–278, 2022, doi: 10.1109/TSC.2019.2937484.

[12] M. Sadegh Riazi, K. Laine, B. Pelton, and W. Dai, 'HEAX: An architecture for computing on encrypted data', in *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, Association for Computing Machinery, Mar. 2020, pp. 1295–1309. doi: 10.1145/3373376.3378523.

[13] M. Al-Rubaie and J. M. Chang, 'Privacy-Preserving Machine Learning: Threats and Solutions', *IEEE Secur Priv*, vol. 17, no. 2, pp. 49–58, Mar. 2019, doi: 10.1109/MSEC.2018.2888775.

[14] Q. Li *et al.*, 'HomoPAI: A secure collaborative machine learning platform based on homomorphic encryption', in *Proceedings - International Conference on Data Engineering*, IEEE Computer Society, Apr. 2020, pp. 1713–1717. doi: 10.1109/ICDE48307.2020.00152.

[15] H. Fang and Q. Qian, 'Privacy preserving machine learning with homomorphic encryption and federated learning', *Future Internet*, vol. 13, no. 4, 2021, doi: 10.3390/fi13040094.

[16] 'Microsoft SEAL (release 3.2)'. 2019. [Online]. Available: https://github.com/

[17] J. Fan and F. Vercauteren, 'Somewhat practical fully homomorphic encryption', *Cryptology ePrint Archive*, 2012.

[18] J. H. Cheon, A. Kim, M. Kim, and Y. Song, 'Homomorphic encryption for arithmetic of approximate numbers', in *International Conference on the Theory and Application of Cryptology and Information Security*, 2017, pp. 409–437.

[19] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, 'Fully Homomorphic Encryption without Bootstrapping', 2012.

[20] 'Palisade Homomorphic Encryption Software Library'. https://palisade-crypto.org/

[21] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, 'TFHE: Fast Fully Homomorphic Encryption over the Torus'.

[22] J. H. Cheon, A. Kim, M. Kim, and Y. Song, 'Homomorphic encryption for arithmetic of approximate numbers', in *International Conference on the Theory and Application of Cryptology and Information Security*, 2017, pp. 409–437.

[23] V. 'Morde and 'Anurag Setty' 'Venkat', 'XGBoost Algorithm: Long May She Reign!' https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d (accessed Aug. 26, 2022).

[24] 'ibmcom/helayers-pylab-s390x'. [Online]. Available: https://hub.docker.com/r/ibmcom/helayers-pylab-s390x

[25] 'https://xgboost.readthedocs.io/en/stable/python/python_api.html'.