T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DOCTORAL THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

Mhd Raja ABOU HARB

ADVANCING PRIVACY AND SECURITY IN MACHINE LEARNING THROUGH HOMOMORPHIC ENCRYPTION AND EXPLAINABLE AI

SUPERVISOR
Asst. Prof. Baris CELIKTAS

İSTANBUL, March 2026

T.C.

IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DOCTORAL THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

Mhd Raja ABOU HARB

(21COMP9001)

ADVANCING PRIVACY AND SECURITY IN MACHINE LEARNING THROUGH HOMOMORPHIC ENCRYPTION AND EXPLAINABLE AI

SUPERVISOR

Asst. Prof. Baris CELIKTAS

İSTANBUL, March 2026

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DOCTORAL THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

Mhd Raja ABOU HARB

(21COMP9001)

ADVANCING PRIVACY AND SECURITY IN MACHINE LEARNING THROUGH HOMOMORPHIC ENCRYPTION AND EXPLAINABLE AI

Date: 5th of March 2026

Thesis Supervisor: Asst. Prof. Baris CELIKTAS, Isik University

Jury Members: Asst. Prof. Emine EKIN, Isik University

Asst. Prof. Sahin AYDIN, Isik University

Asst. Prof. Mehmet Tahir SANDIKKAYA, İstanbul Technical University

Asst. Prof. Erturk ERDAGI, İstanbul Medeniyet University

İSTANBUL, March 2026

ÖZET

MAKİNE ÖĞRENİMİNDE GİZLİLİĞİN VE GÜVENLİĞİN İLERLETİLMESİ: HOMOMORFİK ŞİFRELEME VE AÇIKLANABILIR YAPAY ZEKÂ ÜZERINE BİR ÇALIŞMA

Bulut tabanlı makine öğrenimi çözümlerinde, özellikle sağlık ve finans gibi hassas alanlarda veri gizliliği kritik bir öneme sahiptir. Gizlilik koruması ile yüksek model başarımı arasında denge kurmak ise güncel bir zorluktur. Bu çalışmada, homomorfik şifreleme yöntemleriyle korunan veriler üzerinde çalışan, gizlilik odaklı bir Yapay Sinir Ağı (YSA) yaklaşımı öneriyoruz. Şifreli ortamlarda hesaplama maliyeti yüksek olan Sigmoid ve Tanh gibi doğrusal olmayan aktivasyon fonksiyonlarını verimli yönetmek amacıyla, hafif YSA tabanlı tahminciler geliştirdik. Elde edilen sonuçlar, önerdiğimiz tahmincilerin geleneksel polinom ve parçalı doğrusal yöntemlere göre üstün olduğunu; Ortalama Karesel Hatayı (MSE) %96 oranında azalttığını göstermektedir. MNIST veri kümesinde elde edilen %97,70 doğruluk ve 0,9997 AUC değerleri, yöntemin etkinliğini kanıtlamıştır. Gerçek dünya senaryolarında, QEEG verileriyle disleksi tespitinde düz metin çıkarımına kıyasla ihmal edilebilir performans kayıplarıyla (%2,66 doğruluk, %3,86 AUC) başarı sağlanmıştır. UCI Kalp Hastalığı veri kümesinde ise düz metin performansıyla eşdeğer %85,25 doğruluk elde edilmiştir. Ayrıca, şeffaflığı artırmak amacıyla şifreli çıkarımlara SHAP tabanlı açıklanabilirlik entegre edilmiştir. Bulgularımız, önerilen modelin gizlilik, yüksek performans ve açıklanabilirlik gereksinimlerini başarıyla dengelediğini ve hassas sektörler için güçlü bir çözüm sunduğunu ortaya koymaktadır.

Anahtar Kelimeler: Gizlilik koruyucu makine öğrenimi, Homomorfik şifreleme, Yapay sinir ağları, Açıklanabilir yapay zekâ, SHAP, şifreli çıkarım.

ABSTRACT

ADVANCING PRIVACY AND SECURITY IN MACHINE LEARNING THROUGH HOMOMORPHIC ENCRYPTION AND EXPLAINABLE AI

The importance of data privacy in cloud-based Machine Learning is paramount, particularly in sectors such as healthcare and finance. Balancing robust privacy protection with high model accuracy remains a significant challenge. In this study, we propose a privacy-preserving framework utilizing ANNs on homomorphically encrypted data. To mitigate the computational complexity of non-linear activation functions (Sigmoid and Tanh), we developed lightweight, ANN-based estimators specifically designed for encrypted environments. Our experimental results demonstrate that these estimators significantly outperform traditional polynomial and piecewise linear methods, reducing MSE by up to 96% while improving accuracy and F1-scores. Our method achieved 97.70% accuracy and 0.9997 AUC on the MNIST dataset, validating its effectiveness. In real-world applications, we applied the approach to dyslexia detection using QEEG data, observing only minor performance degradation (2.66% accuracy, 3.86% AUC) compared to plaintext inference. Furthermore, a case study on the UCI Heart Disease dataset yielded 85.25% accuracy in encrypted inference, matching plaintext performance. Finally, we integrated the SHAP algorithm to ensure transparency for encrypted outputs. Our findings confirm that this approach successfully balances privacy, performance, and explainability, making it highly suitable for sensitive ML applications.

Keywords: Privacy-preserving machine learning, Homomorphic encryption, Artificial neural networks, Explainable AI, SHAP, encrypted inference.

ACKNOWLEDGEMENT

First of all, I would like to thank my family, especially my parents and grandparents, they have always loved, supported and encouraged me endlessly. They have believed in me and brought me up. Without them this wouldn't be possible.

I want to express my gratitude to the head of the department and my thesis supervisor for their great work and excellent management which have resulted in this success.

I dedicate this work to my dear Türkiye. I am grateful to Türkiye for providing me with the chance to fulfil my dreams here and for my academic and personal growth.

Lastly, I would like to give special thanks to Alan Turing, the “father of the computer”. Without Turing’s revolutionary theories and developments, we would not have had the necessary technological infrastructure in place to create this thesis. I am endlessly inspired by Turing’s legacy, which constantly motivates me to do research and to recognize the invaluable historical contributions of these incredible people.

To all who have walked this journey with me, thank you for being a part of this milestone.

Mhd Raja ABOU HARB

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS LIST

AI: Artificial Intelligence

ANN: Artificial Neural Network

API: Application Programming Interface

ASIC: Application-Specific Integrated Circuit

AUC: Area Under Curve

BFV: Brakerski/Fan-Vercauteren

CAGR: Compound Annual Growth Rate

CKKS: Cheon-Kim-Kim-Song

CNN: Convolutional Neural Network

CNN-LSTM: CNN- Long Short-Term Memory

CSP: Cloud Service Provider

CV: Cross Validation

DLT: Distributed Ledger Technology

DP: Deferential Privacy

EHR: Electronic Health Record

EU: European Union

GDPR: General Data Protection Regulation

GPU: Graphics Processing Unit

FFT: Fast Fourier Transform

FHE: Fully Homomorphic Encryption

FL: Federated Learning

FN: False Negative

FP: False Positive

HHS: Health and Human Services

HE: Homomorphic Encryption

HIPAA: Health Insurance Portability and Accountability Act

IoT: Internet of Things

IP: Internet Protocol

IT: Information Technologies

KMS: Key Management Service

KVKK: Kişisel Verileri Koruma Kurumu

LIME: Local Interpretable Model-agnostic Explanations

ML: Machine Learning

MLaaS: Machine Learning as a Service

MNIST: Modified National Institute of Standards and Technology

MSE: Mean Squared Error

NIST: National Institute for Standards and Technologies

PHE: Partially Homomorphic Encryption

PHI: Protected Health Information

PPML: Privacy-Preserving Machine Learning

QEEG: Quantitative Electroencephalography

Ring-LWE: Ring Learning With Errors

ROC: Receiver Operating Characteristic

RSA: Rivest–Shamir–Adleman

SGD: Stochastic Gradient Descent

SHAP: SHapley Additive exPlanations

SHE: Somewhat Homomorphic Encryption

SMPC: Secure Multi-Party Computation

SQL: Structured Query Language

SSN: Social Security Number

TEE: Trusted Execution Environment

TILLS: Test of Integrated Language and Literacy Skills

TN: True Negative

TP: True Positive

TPU: Tensor Processing Unit

UCI: University of California, Irvine

US: United States

USD: United States Dollar

XAI: eXplainable AI

CHAPTER 1

1. INTRODUCTION

1.1. BACKGROUND AND CONTEXT

Cloud computing has been on a significant rise over the last decade, due to its cost-effectiveness, which has been particularly attractive for saving operational costs and reducing risks, such as regulatory non-compliance. One of the many emerging facets of cloud computing is Machine Learning as a Service (MLaaS), which offers the ability to access sophisticated machine learning algorithms without requiring an in-depth technical expertise or the large computing infrastructure otherwise necessary to support it. MLaaS brings unprecedented scale and efficiency to businesses looking to take advantage of machine learning, allowing them to quickly and cost-effectively deploy models. However, this growing dependence on MLaaS has created growing concerns with data privacy and security, especially in sensitive fields like healthcare and finance, where data breaches can be particularly costly.

To combat these issues, substantial research has been directed towards Privacy-Preserving Machine Learning (PPML), which seeks to maintain privacy while upholding model performance. A few notable PPML methods include Deferential Privacy (DP), which reduces the influence of the addition or removal of a single data point to help protect the privacy of individuals (Wood et al., 2018), Federated Learning (FL), which is a prominent technique that enables distributed training on decentralized data while maintaining the privacy of original records (Zhu et al., 2024), and Homomorphic Encryption (HE), which is another major approach that allows computations to be performed directly on encrypted data, thus ensuring privacy throughout the processing stages (Park & Lim, 2022). These methods represent crucial advancements in the protection of

data privacy without hampering the utility of machine learning in cloud-based settings.

Privacy-preserving machine learning has seen significant progress in recent years, but the application of HE to Artificial Neural Networks (ANNs) is still in its early stages, mainly due to the need for non-linear activation functions like Sigmoid and Tanh in neural networks to model complex relationships in the data. These non-linear activation functions involve non-polynomial operations that are not natively supported by HE. Previous work has used either polynomial or piecewise linear approximations of these functions. While these approximations are feasible in encrypted space, they often suffer from either limited precision or are specific to the function being approximated. We focus on the recently proposed ANN-based activation function estimators, a technique that leverages the universal approximation theorem of ANNs to better model the non-linear function with higher accuracy and flexibility. Our solution involves training lightweight ANN estimators using plaintext data to emulate non-linear activation functions during encrypted inference rather than performing direct approximations. This provides (as we will prove in this thesis) improved accuracy, generalization, and ease of integration over previous work.

While privacy forms the bedrock of secure Machine Learning (ML) applications, transparent and interpretable models remain essential in highly regulated environments. The use of black-box models, such as ANNs, can lead to a trust problem, as stakeholders are not likely to accept models that cannot explain their decisions. As a remedy, explainable AI (XAI) has been suggested as a possible solution to provide interpretability of the predictions, which would increase user trust and help meet legal and compliance requirements, such as the General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA). Implementing explainability features into homomorphic encrypted inference presents greater difficulties because encrypted data processing itself is a complex procedure. In the present study, we combine the SHapley Additive exPlanations (SHAP) framework with the proposed method in order to increase the interpretability of privacy-preserving

ANN models. Our approach would provide strong data confidentiality and meaningful explainability, which could lead to an increased trust in MLaaS solutions hosted on cloud platforms.

1.2. PROBLEM STATEMENT

The increasing popularity of MLaaS on cloud-based infrastructures has raised significant privacy issues in certain industries, more specifically across sectors such as health services and financial systems where information privacy is of utmost important. Although HE can process data in encrypted form it struggles to combine with neural networks because of the high computational cost of non-linear activation functions. Moreover, the opaque nature of ANNs also hinders the interpretability of encrypted results, which is crucial for both trust development and regulatory compliance in these sensitive fields. This research tackles these challenges through the combination of HE with ANN-based estimators to protect privacy and Explainable AI techniques like SHAP for better encrypted model interpretability.

1.3. OBJECTIVES OF THE STUDY

The main goal of the study is to design a privacy-preserving machine learning framework that enhances data security within the cloud-based MLaaS model. The study specifically intends to:

- Explore and analyse the increasing trend of cloud computing adoption and the consequences associated with the usage of MLaaS, especially in critical sectors, including the healthcare services and financial systems.
- Employ state-of-the-art methods to manage the increased need for privacy.
- Elevate the security and privacy level of MLaaS implementation by applying HE to address the risks associated with sensitive data processing through machine learning inferences.

- Provide a solution that could effectively protect the sensitive data from potential breaches without affecting the model performance through HE application integrated with ANNs.
- Enhance transparency and interpretability of machine learning models by utilizing XAI techniques, e.g., SHAP, for building trust and supporting adherence to data protection regulations.

1.4. RESEARCH HYPOTHESIS

This study proposes that the inclusion of homomorphic encryption and XAI in MLaaS platforms could improve the privacy and security of MLaaS systems, while ensuring adequate model performance and interpretability. The resulting increase in trust would in turn lead to greater adoption of machine learning solutions deployed in the cloud, with specific use cases in the healthcare domain.

1.5. SCOPE AND CONTRIBUTIONS

In this thesis, a privacy-preserving machine learning framework that leverages HE and XAI is developed for addressing privacy, security, and transparency issues in MLaaS deployments. The study aims to address fundamental problems in machine learning within cloud environments to meet the needs of healthcare sector by implementing secure and interpretable machine learning solutions. The major contributions of this thesis are:

1. Privacy-preserving machine learning framework: In this thesis, an approach is proposed to apply homomorphic encryption on ANN models for making machine learning inference secure. The proposed framework ensures that the sensitive data are fully protected during the processing stage, which enhances privacy for machine learning in cloud environment.
2. Non-linearity approximation using ANN estimators: A critical aspect of

implementing HE in neural networks is the efficient processing of non-linear activation functions. In this thesis, ANN based estimators are proposed to approximate complex non-linear functions (Sigmoid, Tanh, etc.) in encrypted domain. The approach not only enhances the model's accuracy but also outperforms the previous activation functions estimators in the literature such as polynomial or piecewise approximations in terms of efficiency.

3. Integration of explainable Artificial Intelligence (AI) for encrypted inferences: As transparency in machine learning becomes increasingly important, particularly in the regulated industry, this thesis combines the SHAP algorithm to enable interpretability for encrypted inferences. By providing interpretability to the encrypted model predictions, this work addresses the combined needs of secure ML and explainability to promote the wide adoption of machine learning in cloud environments.

1.6. STRUCTURE OF THE THESIS

The thesis is organized into eight chapters, each serving a specific purpose in presenting and discussing the research on privacy-preserving machine learning by leveraging homomorphic encryption and explainable AI. The chapters develop in a logical sequence that forms a complete picture through a step-by-step progression of ideas and discoveries. Chapter 1: Introduction: This chapter introduces the research topic and provides background information to establish the context and motivation for the study. It highlights the increasing adoption of cloud-based MLaaS and its widespread use in industries that handle sensitive data such as healthcare and finance. The chapter clearly states the problem statement, research objectives, scope, and contributions. It also provides a brief overview of the thesis structure to orient the reader. Chapter 2: Literature Review: This chapter presents an in-depth analysis of the existing literature and research in areas related to the proposed solution. It provides a comprehensive review of previous works and technological advancements. It starts by discussing the benefits of cloud-based machine learning and MLaaS and then highlights the data privacy concerns associated with this computing paradigm. It then discusses

the past works that have been proposed for privacy-preserving machine learning, with a focus on the application of homomorphic encryption to ANNs. The chapter also reviews previous works that have proposed solutions for the non-linear activation functions in the encrypted domain. It concludes by reviewing prior works that have leveraged explainable AI techniques, such as SHAP, to interpret machine learning models in sensitive and regulated sectors like healthcare. Chapter 3: Preliminaries: This chapter provides an overview of the mathematical foundations and key concepts required to understand the proposed solution. It starts by introducing the basic principles of homomorphic encryption. It then provides a brief overview of ANNs, focusing on the key aspects related to activation functions and their non-linear nature. The chapter also includes a subsection on Explainable AI techniques, providing an introduction to the SHAP algorithm and its application in interpreting machine learning models. To provide regulatory perspective, this research-oriented chapter also includes a discussion on data privacy regulations relevant to the study, such as GDPR, HIPAA, and Turkish data protection laws, highlighting their implications in MLaaS contexts. Chapter 4: Proposed Methodology: This chapter describes the design and implementation of the proposed privacy-preserving machine learning solution. It includes the system architecture, data flow diagrams, and the integration of homomorphic encryption with ANN models. This chapter also details the proposed ANN-based estimators to approximate the non-linear activation functions in the encrypted domain. In addition, the chapter describes the application of explainable AI methods to interpret encrypted model predictions and provide insights into the decision-making process. A comprehensive description of the datasets used in the study, the experimental setup, and the evaluation metrics are also provided to validate the proposed solution. Chapter 5: Experimental Setup: In this chapter, the experiments conducted to validate the proposed solution are described in detail. It includes the various scenarios tested and the evaluation measurements used to assess the performance of the proposed solution. This chapter provides a clear view of the experimental design, including the encryption parameters, datasets, and computational setup used in the experiments. Chapter 6: Results: This chapter presents the results and outcomes of the implementation and evaluation of the proposed solution. It starts with the performance results obtained using the Modified National Institute of Standards and Technology (MNIST) dataset and the comparison with other estimators. To show the real-world applicability of the proposed framework, it is evaluated on two important healthcare datasets, including a

Quantitative Electroencephalography (QEEG) dataset for dyslexia detection and the University of California, Irvine (UCI) heart disease dataset. These applications on important biomedical classification problems of different modalities in healthcare domain show the flexibility of the proposed method under privacy-preserving settings. The chapter also presents the result of the explainability framework that used the SHAP method to interpret the MLaaS models. Since SHAP explanations were performed on high-dimensional QEEG dataset, these interpretations were only applied to this dataset. Chapter 7: Discussion: In this chapter, the results are discussed and interpreted in the context of existing literature. It highlights the significance and implications of the findings for privacy-preserving ML solutions in sensitive and regulated sectors. The chapter also discusses the limitations of the study, such as the computational overhead and scalability issues, and presents opportunities for improving and extending the proposed framework. Chapter 8: Conclusion and Future Work: This chapter provides a summary of the key findings and contributions of the research. It emphasizes the impact of the proposed solution in privacy-preserving machine learning and explainable AI. The chapter concludes with recommendations for future work to further advance the research in the areas of privacy, efficiency, and transparency in cloud-based MLaaS applications.

CHAPTER 2

2. LITERATURE

2.1. CLOUD-BASED MACHINE LEARNING

The current definition of cloud computing by the National Institute for Standards and Technologies (NIST) in publication 800-145 is a “model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (NIST, 2011). As the definition of cloud computing suggests, it is about providing generalized computational resources to a large variety of users, with minimal effort and high immediacy. The use of cloud infrastructure has the potential to dramatically lower the capital expenses, personnel, and overall costs of running Information Technologies (IT) infrastructure by taking advantage of the enormous economies of scale of cloud computing companies. The definition highlights the primary value proposition of cloud computing. It is about offering to a user a variety of resources, seamlessly and with a high level of immediacy. The use of cloud computing infrastructure helps to lower the capital investments, the number of required staff members, and the associated costs, as well as lowering the long-term recurring operational costs. Moreover, a key advantage of cloud adoption is the reduction in regulatory compliance expenses for the adopting organization, as many of the compliance responsibilities and associated costs are shifted directly to the cloud providers.

These advantages have contributed to the rapid adoption of cloud solutions in various industries, such as healthcare, finance, and retail, where scalability and cost-effectiveness are crucial. MLaaS has gained popularity alongside cloud

computing development as firms offer platforms to build and launch ML applications swiftly without a lot of experience in this field. This approach enables organizations to leverage artificial intelligence technologies without significant technical expertise or infrastructure.

According to a report by Mordor Intelligence, the MLaaS market is projected to grow from 33.75 billion United States Dollars (USD) in 2024 to 154.59 billion USD by 2029, exhibiting a Compound Annual Growth Rate (CAGR) of 35.58% during the forecast period (Mordor Intelligence, 2024). This growth is fuelled by the expanding use of cloud-based services, the widespread integration of Internet of Things (IoT) devices, and the growing demand for automation across different industries.

Cloud machine learning offers several benefits, but its growing popularity has exacerbated privacy and security concerns. This is especially the case for confidential data such as patient medical records and financial transactions. The minimal effort required for provisioning and releasing new resources leads to data processing on external servers that may fail to meet essential confidentiality and security standards. For this reason, maintaining sensitive data while still being able to leverage cloud scalability has become an urgent need.

Take healthcare as an example: moving to the cloud has significantly enhanced data accessibility, storage, and operational efficiency. However, it also introduced a variety of security vulnerabilities that have the potential to be actual threats. Sivan and Zukarnain (2021) provided data as evidence that storing and sharing Electronic Health Record (EHR) data, sensitive patient data, with third parties in the cloud exposes healthcare organizations to a variety of security concerns. These risks include data breaches and unauthorized access, and disclosure of health data to identity theft as a result of shared personal information. These difficulties are compounded by weaknesses in encryption, lack of access control, and incorrect configuration management with cloud storage, which might all lead to unintended data exposure. Raja and Chopra (2024) also noted that insecure Application Programming Interfaces (APIs) and authentication systems that are not as strong as they should be both make it more

likely that bad actors will use Structured Query Language (SQL) injection and denial-of-service attacks to prevent critical healthcare services. Vulnerability to cyber threats is further highlighted by Jaber (2024), who emphasizes that cloud systems storing or handling sensitive medical information are particularly susceptible to cyberattacks. The use of strong encryption, multi-factor authentication, and constant monitoring are just a few examples of mitigating measures that have been recommended to decrease the risk of these attacks. These difficulties are compounded by the inherently interconnected and distributed nature of cloud environments, which further magnifies privacy challenges, increasing the attack surface and complicating data provenance tracking. Consequently, strict adherence to comprehensive regulations like HIPAA is paramount for safeguarding patient confidentiality and maintaining public trust. In order to ensure healthcare data confidentiality when using cloud technology, organizations need to implement rigorous encryption alongside robust access controls and constant monitoring systems to handle potential risks. In response to these problems, this study was proposed to apply privacy-preserving technologies, particularly homomorphic encryption and explainable AI, to boost both security and transparency in a cloud-based healthcare setting.

2.2. OVERVIEW OF PPML

PPML's goal is to make machine learning possible without having to sacrifice the confidentiality or integrity of sensitive data. The field has gained traction in recent years due to a growing trend of using machine learning for use cases involving sensitive data in healthcare, finance, and government services, while also facing security attacks and new privacy regulations. The earliest related works to PPML include HE by Rivest et al. (1978) and Secure Multi-Party Computations (SMPCs). Dwork and Roth (2014) also formalized the idea of DP, a technique to provide a mathematically rigorous privacy guarantee to individual data points processed with a model. Throughout the 2010s, as more and more machine learning models were being trained and put to use for real-

world applications, research started to focus on how to adapt those privacy-preserving techniques to the typical workflows for model training and inference to provide better protections against emerging attacks on machine learning models, such as membership inference and model inversion (Shokri & Shmatikov, 2015). In present day, PPML is continuing to evolve further with the rise of cloud-based MLaaS solutions, as well as data protection laws such as GDPR, HIPAA, and Kişisel Verileri Koruma Kurumu (KVKK, Turkish Personal Data Protection Authority).

PPML consists of a family of techniques that aim to address the privacy risks with a certain trade-off to the performance of the models. HE and SMPC provide strong privacy by allowing the computation on either encrypted data or data distributed among participants without revealing the data itself (Cao et al., 2023; Guo et al., 2019).

Privacy risks can also be countered by perturbation-based methods such as DP. DP works by adding statistical noise to a dataset or to the results of a query on a dataset. The effect of each person's data is masked in the output of the model, preventing it from affecting the utility of the entire dataset by a significant amount (Mehta et al., 2025). Another method is anonymization, in which the personal information that can be used to trace the source of the data is removed or obfuscated. However, there are many examples of anonymization failing to ensure privacy; when it is combined with external public sources of information, it is still possible to re-identify the subjects in anonymized datasets (Al-Rubaie & Chang, 2018; Mehta et al., 2025). DP, on the other hand, can be adjusted so that re-identification is infeasible, but this comes at the cost of accuracy.

FL provides a decentralized approach to PPML. It enables collaborative learning processes on distributed datasets without exposing raw data. However, FL is susceptible to attacks, such as gradient inversion, leading to a surge in research on hybrid solutions. These combine FL with HE to improve privacy (Su et al., 2024).

Conversely, other studies have pointed out the deficiencies of the current

PPML solutions. For instance, Al-Rubaie & Chang (2018) believed that the prevalent approaches are not flexible enough to fit the different ML models, particularly in the increasingly complex ML architectures. In addition, Xu et al. (2021) mentioned that the communication and computation bottlenecks should be properly tackled for scaling up the privacy-preserving techniques.

In the spirit of the above, our research extends these efforts by harnessing HE to provide secure inferences on ANNs within the context of privacy-preserving machine learning frameworks. HE empowers computations on encrypted data, thus maintaining the privacy of sensitive information throughout the inference phase. Such capabilities are not only pivotal but essential in domains with stringent privacy mandates, such as healthcare and financial services, where they must stand up to rigorous privacy regulations and provide robust defences against privacy violations. While executing neural network operations under encryption, particularly non-linear activation functions, presents substantial hurdles; our contribution overcomes these constraints by devising efficient approximations and introducing innovative methodologies tailored for HE. Focusing on the inference phase of ANNs, our work strikes a balance between privacy and computational overhead, showcasing the feasibility of deploying secure machine learning models in practical settings. This research contributes to the growing literature on the convergence of homomorphic encryption and ANNs, pushing the boundaries of secure MLaaS with an emphasis on competitive performance benchmarks.

2.3. CHALLENGES OF NON-LINEARITY IN HOMOMORPHICALLY ENCRYPTED ML MODELS

Modelling ANNs has been a challenge with HE. One main hurdle has been the injection of non-linearities, which are necessary for the performance of the model. Non-linear functions, such as Sigmoid and Tanh, rely on complex operations (exponentials and square roots) which are not natively available in most HE schemes, which only support addition, subtraction, and multiplication.

For this reason, efficient approaches for approximate non-linear function evaluation have been an active area of research.

To overcome the challenges of computing non-linear activation functions within HE, some earlier studies proposed designing ANNs without activation functions entirely (Aremu & Nandakumar, 2023). Another approach involved constructing linear ANNs (Sarkar et al., 2021; T'Jonck et al., 2022) or using linear approximations of activation functions (Almutairi et al., 2023). While these methods simplify the computation under HE, they compromise the core advantages of ANNs by eliminating the non-linear features that are essential for capturing complex data patterns.

With the aim of adding non-linear behaviours, some studies applied the activation function at the final layer after the decryption process (Marcel et al., 2023). Nevertheless, the approach falls short because it restricts the learning of sophisticated relationships across the network's internal layers. By keeping the hidden layers linear, the model loses its capacity to model intricate decision boundaries, which is a fundamental strength of non-linear activation functions in ANNs.

Additional studies have attempted to use Trusted Execution Environments (TEEs) for activation function calculations. TEEs offer a protected, separate area within a processor for sensitive operations, enabling computations to be performed in a way that both data and code are safe, even from a possibly compromised operating system. Secure enclaves and remote attestation in TEEs assure that processed data can be kept confidential and computation integrity is maintained in untrusted environments (Natarajan et al., 2023).

Another approach explored in earlier studies involves the use of non-colluding dual clouds (Baryalai et al., 2016; Yao et al., 2021; Lei et al., 2023). In this method, one Cloud Service Provider (CSP) performs the classification process, while a second CSP generates key pairs for each client and processes the activation functions on the encrypted weight sums computed by the first cloud. The second cloud, having access to the decryption keys, decrypts the sums, applies the activation functions, and re-encrypts the results before sending

them back to the first cloud. Although this approach addresses the non-linear activation challenge for ANNs on HE data, it requires a level of trust in the CSP handling the key pairs and increases the potential attack surface.

Table 2.1 summarizes the previous works that aimed to estimate activation functions for implementation on HE data.

Table 2.1 Summary of Previous Works on Activation Function Estimation for Homomorphic Encryption-based ANNs.

Reference	Model /framework	HE scheme	Activation function handling	Datasets	Performance	Limitations
Usman et al. (2025)	HoRNS-CNN: privacy-preserving deep CNN with RNS-FHE for neuro-biomarker classification	RNS-FHE; energy-efficient, FPGA-friendly, 3-moduli set	ReLU replaced by degree-3 Taylor polynomial; BN for stability	OpenNeuro & C-BIRD MRI (patch-based, 148 subjects, dyslexia vs. control)	91.4% accuracy (encrypted), 400k features/hr, 42.4% energy saving	Decryption slower than encryption; tested only on binary task; cipher expansion; hardware eval only
Allavarpu et al. (2025)	Neural network for credit risk analysis integrating TenSEAL and Torch	CKKS (via TenSEAL library)	Uses approximate arithmetic for NN layers; does not explicitly detail specialized polynomial approximations for activation functions	Real-world financial datasets from multiple countries	Comparable accuracy with/without HE (e.g., minor F1 drop); validated resilience in credit-risk classification	Overhead associated with encrypted data operations; mainly tested on tabular financial data rather than large-scale image or deep networks

Table 2.1 (Next) Summary of Previous Works on Activation Function Estimation for Homomorphic Encryption-based ANNs.

Reference	Model /framework	HE scheme	Activation function handling	Datasets	Performance	Limitations
Lee Junghyun (2024)	Polynomial-approximation-based inference on large-scale CNNs (ResNet, VGG)	RNS-CKKS	Proposes composite minimax and weighted least-squares polynomials for ReLU and max-pooling; allows evaluation of deep nets on encrypted data	ImageNet, CIFAR	Can reach ~77% top 1 on ImageNet (ResNet-152) via polynomial-based ReLU under encryption	Expensive bootstrapping for deeper networks; polynomial management is complex; still large run-time overhead relative to plaintext
Xiong et al. (2024)	CNN with "Self-Learnable Activation Function" (SLAF) for HE	CKKS-based fully homomorphic approach	Learns polynomial-like activation parameters to reduce approximation error under HE constraints	UTKFace (face recognition)	Accuracy gains of ~0.88–3.15% over standard polynomial methods, ~4.87–9.67% over CryptoNets for face-verification	Tested on a single face dataset; learning custom activations adds overhead; large-scale or more complex tasks not extensively shown
Song & Shi (2024)	ReActHE	CKKS	Scaled power function on residues	Heart Failure, Stroke, Hepatitis C, TCGA, SARS-CoV-2, Yeast Genomics	COVID dataset: 98.87%, Heart Failure: 91.14%, Stroke: 89.27%	High computational cost, challenges in optimizing deep networks

Table 2.1 (Next) Summary of Previous Works on Activation Function Estimation for Homomorphic Encryption-based ANNs.

Reference	Model /framework	HE scheme	Activation function handling	Datasets	Performance	Limitations
Lin et al. (2024)	CrossNet	CKKS	2nd-degree polynomials	MNIST, CIFAR-10	MNIST, CIFAR-10 & MNIST: 98.70\%, CIFAR-10: Competitive with state-of-the-art	Computational and communicatio n overhead, client dependency
Shi and Zhao (2023)	Binary convolutional neural network (BCNN) over a vector-based homomorphic encryption (VHE)	Proposed ‘‘efficient integer vector’’ VHE scheme	Binarizes activation to ±1 (sign) for simpler polynomial overhead	MNIST	~93.75% train accuracy, ~86% test accuracy with reduced overhead	Binarization lowers representation al capacity; mostly tested on simpler MNIST tasks
Khan & Michalas (2023)	CNN with Chebyshev Polynomials	HE with Chebyshev Polynomials	Low-degree Chebyshev polynomials	MNIST	MNIST: 98.5%	Limited generalizabilit y due to CNN dependency
Nguyen et al. (2023)	HeFUN	CKKS	Polynomial approximatio n for activation functions like ReLU	MNIST, CIFAR-10	MNIST: 98.9%, CIFAR-10: 85.7%	Computational complexity due to polynomial approximation s, increased latency for larger models
Lee et al. (2022)	ResNet-20 with RNS-CKKS	RNS-CKKS	Minimax polynomial for ReLU, SoftMax	CIFAR-10	CIFAR-10: 92.43% ± 2.65%	High memory requirements, complexity

Table 2.1 (Next) Summary of Previous Works on Activation Function Estimation for Homomorphic Encryption-based ANNs.

Reference	Model /framework	HE scheme	Activation function handling	Datasets	Performance	Limitations
			approximation			due to bootstrapping
Hong et al. (2022)	Shallow Neural Network (SNN)	CKKS	Polynomial approximation of SoftMax	Cancer Genome Atlas (TCGA)	TCGA: 85%	Challenges with polynomial approximation of SoftMax
Xiong et al. (2020)	Encrypted CNN Ensemble	BFV	Low-degree polynomial approximations for non-linear operations	MNIST	Accuracy: 96%-97% with CNN ensemble	High computational overhead; time-intensive due to ensemble structure
Vizitiu et al. (2020)	Deep neural network on encrypted medical data (classification tasks)	Matrix Operation for Randomization or Encryption (MORE)	Approximates non-linear layers with rational polynomial expansions or direct floating-point homomorphic support; focuses on ReLU-like replacements	MNIST, medical data (e.g., X-ray coronary angiography)	Comparable classification accuracy to plaintext; overhead grows with network depth, but demonstrated feasible for real medical scenarios	Encryption overhead increases computation time: specialized scheme (MORE) not as widely adopted as CKKS/BFV; potential memory overhead
Izabachène et al. (2019)	Fully masked Hopfield neural network for face recognition	GSW-type FHE variant	Avoids standard ReLU by using a Hopfield-style network	Small face-recognition sets (e.g., ORL, Yale)	~0.6s per recognition on standard CPU; secure evaluation feasible for	Specialized for Hopfield nets; limited scalability to large modern DNNs;

Table 2.1 (Next) Summary of Previous Works on Activation Function Estimation for Homomorphic Encryption-based ANNs.

Reference	Model /framework	HE scheme	Activation function handling	Datasets	Performance	Limitations
			with discrete sign-like updates; no standard polynomial activation		face recognition	specialized non-standard activation approach
Hesamifard et al. (2018)	CryptoDL Framework	HELib (Leveled HE)	Polynomial approximations for Sigmoid, ReLU, and Tanh functions	UCI datasets, MNIST, CIFAR-10	MNIST: 99%, CIFAR-10: 91.5% with polynomial approximation	High computational overhead, limited to low-degree polynomials for practicality
Hesamifard et al. (2017)	Neural network with polynomial approximations for activation functions	HELib	Polynomial approximations of Sigmoid and ReLU	MNIST, UCI datasets (Crab, Fertility, Climate)	MNIST: 99.1% with polynomial approximations for ReLU	High computational overhead due to polynomial approximations; noise growth management is needed during computations

2.4. PRIVACY-PRESERVING AND EXPLAINABLE AI IN HEALTHCARE: INSIGHTS FROM DYSLEXIA DETECTION

Several recent studies have explored PPML techniques in healthcare, including a few targeted at dyslexia detection. Scheibner et al. (2022) conducted an ethical/legal analysis of applying HE and Distributed Ledger Technology (DLT) to health data sharing. Their interviews with experts in Switzerland

highlighted the feasibility of HE and blockchain for secure patient data collaboration but noted that these efforts remain conceptual – no actual ML model was implemented, and the work focused on regulatory and ethical challenges rather than technical solutions. In a broader survey, Munjal & Bhatia (2023) reviewed HE techniques in healthcare applications. They compared partially, somewhat, and fully homomorphic encryption schemes (largely based on lattice cryptography) and documented their use in areas such as EHR analytics and genomic studies. Fully homomorphic encryption can protect sensitive medical data but incurs high implementation costs and faces challenges when integrating with existing workflows. This is a recurring theme in PPML, that increased privacy often comes at the expense of reduced efficiency and must be carefully engineered.

Focusing specifically on dyslexia detection, early work by Usman and colleagues pioneered PPML approaches using neuro data. Usman & Muniyandi (2020) introduced CryptoDL, a Convolutional Neural Network (CNN) that operated on homomorphically encrypted brain Magnetic Resonance Imaging (MRI) scans to distinguish dyslexic individuals from controls. They employed a residue number system-based encryption scheme allowing arithmetic on 7-bit pixel values and achieved an accuracy of about 73.2% on the encrypted MRI data. This demonstrated the viability of dyslexia classification without decrypting sensitive neuroimaging data, albeit with a moderate accuracy on a small dataset (45 subjects). In follow-up research, Usman et al. (2022) improved on this approach by integrating an RNS-enabled Fully Homomorphic Encryption (FHE) scheme with a CNN and more advanced activation function handling. They leveraged degree-3 Taylor polynomial approximations for non-linear activations in an encrypted model and even used pre-trained CNN features. As a result, the encrypted dyslexia classifier reached about 93% accuracy on an fMRI-based dyslexia dataset – a performance much closer to plaintext models. This high accuracy, accompanied by only a minor drop from the unencrypted baseline, showcases the progress in HE-based model inference for dyslexia. However, the authors noted that scaling such solutions to larger datasets remains

challenging, as the cryptographic overhead grows and slight accuracy reductions may occur with more complex data.

Work has also been done using FL and DP in addition to HE for medical data privacy preservation. Mercier et al. (2022) is one such example, benchmarking several PPML techniques on a time-series health dataset (a public ECG5000 ECG signal set). They use a 1D CNN (AlexNet-based) and compare it under different privacy arrangements (pure HE, pure DP, pure FL, and hybrid configurations). The results show that FL with DP is able to preserve data while maintaining high performance (weighted F1-score ~92.5% for the ECG task) and that an FHE was computationally infeasible to scale, becoming a performance bottleneck with significant overhead that made it infeasible to train and use on longer time-series. This work was not dyslexia-specific, but there are takeaways for this space: a hybrid solution of the privacy-preserving techniques has real trade-offs with each other. FL and DP can decentralize and obfuscate data to protect privacy, with a smaller loss to accuracy, but HE, while secure, has the potential to be a speed and scalability bottleneck. This information can help design privacy-preserving dyslexia detection.

In addition to privacy-preserving work, other studies have focused on XAI techniques for dyslexia detection. XAI approaches were applied to explain what features drive model decisions to non-expert end users, typically clinicians and educators in the dyslexia context. Gallego-Molina et al. (2024) introduced an EEG-based dyslexia classifier based on a CNN- Long Short-Term Memory (CNN-LSTM) hybrid model. The authors used LIME to interpret their results and demonstrated the local interpretability of their model. The study was based on quantitative EEG recordings from 15 children with dyslexia and 33 control subjects and used a feature extraction method for obtaining spatiotemporal representations of neural synchrony. The interpretable model uncovered brain-region-specific patterns in EEG recordings associated with dyslexia – for example, decreased phase synchronization in left temporal and frontal regions, indications of compensatory mechanisms in right-hemisphere areas, and distinct oscillatory dynamics in occipital and parietal regions. Overall, the explainable

EEG analysis of neural synchrony by Gallego-Molina et al. (2024) revealed neurophysiological markers of dyslexia in agreement with those known from the literature, while retaining a reasonable accuracy. The interpretable model could classify between dyslexic and typical readers with a balanced accuracy of approximately 83%. Another work in this category is Robaa et al. (2024), which introduced an XAI framework for dyslexia detection from handwriting data. The authors used transfer learning based on transformer models and achieved high precision (~99.6%) for the classification task of detecting dyslexic handwriting patterns. More importantly, the handwriting features (letter spacing, sizing irregularities) relevant to the model's predictions were identified using Grad-CAM visualizations. This explainability is important for educators and other non-expert end users to gain trust in AI-assisted dyslexia screening by providing the rationale for a given prediction. As for privacy preservation, the explainable models introduced by Gallego-Molina et al. and Robaa et al. did not use privacy-enhancing methods, and all data were analysed in plaintext in centralized systems. In the real world, EEG recordings and handwriting samples are considered sensitive personal information, and so, in these works, the confidentiality of the raw data was not prioritized while interpretability was attained. Table 2.2 below provides a summary of major works addressing the problem of privacy in dyslexia detection.

Table 2.2 Summary of Previous Works in Privacy-Preserving and Explainable AI in the Healthcare Sector.

Reference	Dataset Details	ML Model (Estimators)	Privacy	Explainability	Security Aspects	Key Contributions
Usman & Muniyandi (2020)	MRI brain images (45 subjects)	CNN (RNS-based HE estimator)	Yes	No	None	Early demonstration of encrypted MRI-based dyslexia classification (~73.2% accuracy)
Usman et al. (2022)	fMRI brain images	CNN (RNS-FHE,	Yes	No	None	High accuracy (~93%) in encrypted

Table 2.2 (Next) Summary of Previous Works in Privacy-Preserving and Explainable AI in the Healthcare Sector.

Reference	Dataset Details	ML Model (Estimators)	Privacy	Explainability	Security Aspects	Key Contributions
		polynomial estimator)				inference, feasibility demonstration
Robaa et al. (2024)	Handwriting samples	Transformer-based model	No	Yes	None	Explainable detection of dyslexia via handwriting analysis, achieved high precision (~99.6%)
Gallego-Molina et al. (2024)	EEG signals (33 controls, 15 dyslexics)	CNN-LSTM	No	Yes	None	Interpretable EEG features, identified neurophysiological markers
Our Proposal	QEEG data (200 participants, 100 dyslexic)	ANN (CKKS-HE, ANN estimator)	Yes	Yes	Threat modelling + Game-theoretic defence	Combines privacy-preserving HE with SHAP-based explainability, minimal accuracy trade-off (accuracy in HE inferences 90.03%)

2.5. RESEARCH GAPS IN PRIVACY-PRESERVING AND EXPLAINABLE MACHINE LEARNING

The high time complexity of computation over encrypted data (Table 2.1) is the major problem observed in many PPML methods. In order to cope with the inefficiency, most existing methods rely on low-degree polynomial approximations of non-linear activation functions, which can be directly expressed in HE schemes such as Cheon-Kim-Kim-Song (CKKS) and Brakerski/Fan-Vercauteren (BFV). However, these polynomial approximations are insufficient in approximating the S-shaped complex functions, such as

Sigmoid and Tanh, resulting in low expressiveness and accuracy of the encrypted neural network models.

Moreover, several works demonstrated good results with complex noise management techniques - in particular frequent bootstrapping. This, however, introduces a high computational and architectural overhead, making the technique less practical to deploy in a real-world setting. One interesting research direction is to design an architecture that can offer security and accuracy guarantees while avoiding heavy bootstrapping.

The third major gap concerns the approximation of non-linear activation functions. While polynomial-based approaches have been proposed, they are still limited in terms of accuracy and flexibility. For this reason, this work proposes an ANN-based estimator. It was inspired by the Universal Approximation Theorem (Hornik et al., 1989). This solution allows a more accurate and flexible approximation of the Sigmoid and Tanh activation functions. In contrast to polynomial approximations, this solution is not limited by its intrinsic bias. Moreover, the employed ANN structure is simple and easy to compute, achieving a good trade-off between computational cost and approximation accuracy. Overall, the proposed estimator is not susceptible to bootstrapping complexity, making it well-adapted to PPML contexts.

Beyond these methodological research gaps, domain-specific gaps remain unaddressed as well. As illustrated in Table 2.2, the intersection of HE and XAI has attracted very limited attention in healthcare settings (specifically concerning dyslexia detection). While privacy-preserving solutions (Usman et al., 2022) and interpretability methods (Ter-Minassian et al., 2024) have been proposed separately, to the best of our knowledge, no prior study has been found to comprehensively address both. This is a major gap, as a single, unified framework is needed for truly secure and trustworthy clinical adoption. Furthermore, computational overhead has emerged as a key bottleneck in HE-based deep learning (Mercier et al., 2022; Munjal & Bhatia, 2023) but remains largely unaddressed.

This thesis bridges these gaps in the literature by putting forth a unified framework that simultaneously integrates CKKS-based HE and SHAP-based XAI. The framework is evaluated on a balanced QEEG dataset of 200 participants (100 dyslexic

and 100 controls) and made to support both secure and interpretable dyslexia detection. We demonstrate that the proposed solution can achieve competitive performance (90.03% accuracy) in encrypted inference, while avoiding the high overheads of bootstrapping observed in prior work.

Finally, unlike the majority of prior efforts in the space, we introduce a formal threat model as well as game-theoretic defence analysis against adversarial risks, including model inversion, membership inference, and access-pattern leakage. As such, this work is among the first to jointly address the aspects of privacy, interpretability, efficiency, and security in the context of healthcare-focused PPML.

CHAPTER 3

3. PRELIMINARIES

3.1. ANNs

ANNs are among the most widely and powerful applied tools in artificial intelligence that mimic organic nervous systems, such as the human brain. Mathematically, they model nonlinear input–output behaviour and learn to reproduce the brain's adaptable mechanisms. Networks are engineered to recognize patterns and model data in order to solve a wide range of complicated computational problems.

ANNs can be formally defined by three components: a neuron model, an architecture, and a learning algorithm. The neuron model is the network's basic information-processing unit. The architecture is the overall network structure or how neurons are interconnected, including a set of links, which have assigned connection weights that determine the strength of the interconnections. The learning algorithm is the method used to train the network, where learning proceeds by successive changes of the connection weights based on input–output pairs, allowing the model to learn the task and generalize it to both training data and new instances.

The Universal Approximation Theorem states that a feedforward ANN with a single hidden layer and a finite number of neurons can approximate continuous functions on compact subsets of R^n, given appropriate activation functions and sufficient training (Hornik et al., 1989). The Universal Approximation Theorem demonstrates that ANNs possess the ability to learn nonlinear dynamics together with complex data relationships and patterns. Even though more layers are generally needed to make this process computationally efficient and scalable in practice, the Universal Approximation Theorem

provides the theoretical basis that ANNs have the representational power to do so. This theoretical assurance is one reason why ANNs are used in a variety of applications, including privacy-preserving machine learning.

3.1.1. Mathematical Foundation of ANNs

As described by Alpaydin (2021), an ANN is a function $f(x;\theta)$ that maps an input vector or feature set x to an output y, which can represent either classification or regression. This mapping is achieved using a set of parameters θ, consisting of weights and biases that the model learns during training. The processing within a neuron involves combining inputs through weighted summation, adding a bias, and applying an activation function σ, which introduces the non-linearity into the model. For a single neuron, the output y is computed similarly to the equation:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \qquad (3,1)$$

Where w_i are the weights, x_i are the input features, b is the bias, and σ is the activation function (such as Sigmoid or Tanh).

For more complicated ANN topologies, the output can be computed recursively such that the output of each layer is the input of the next. The output of an ANN with L layers can be computed as:

$$f(x;\theta) = \sigma_L(W_L.\sigma_{L-1}(W_{L-1}.\dots\sigma_1(W_1.x + b_1)\dots + b_{L-1}) + b_L) \qquad (3,2)$$

Where, W_l and b_l are the weights and biases for the layer l respectively, and σ_l is the activation function used in that layer.

3.1.2. Learning Process of ANNs

The learning process in an ANN revolves around optimizing its hyperparameters, represented as $\theta = \{W, b\}$, to boost performance while minimizing the loss $L(y, y')$, where y denotes the actual value and y' is the estimated value for it. Among loss functions, the Mean Squared Error (MSE) is

commonly applied, as it measures the Euclidean distance separating observed and estimated values.

Optimization methods based on gradients, such as Stochastic Gradient Descent (SGD), are often used to train ANNs. The first step is forward propagation, where inputs are fed into the network layer by layer. Neurons process the information by computing a weighted sum of the inputs, adding a bias term, and then applying an activation function to generate a prediction.

Then, backpropagation is used to adjust the model parameters using the error information. The backpropagation step calculates the gradient of the parameters using the chain rule. The calculated gradients then determine the direction of the correction. The optimization algorithm (SGD in most cases) updates the weights and biases in a magnitude according to both the gradients and a predefined learning rate.

The steps of the forward propagation, loss calculation, backpropagation, and weight update loop are repeated for a certain number of epochs. The loop minimizes the loss of the model, thus leads to model convergence, learns the critical structure from the training set, and transfers this to the unknown data for prediction. The ANN training process is fundamentally based on forward propagation, backpropagation, and gradient-based update steps such as SGD (Epelbaum, 2017).

3.1.3. Activation Functions in ANNs

An important component of ANNs is the activation function. Activation functions are the key ingredients that introduce non-linear behaviour into the neural networks to increase the accuracy of approximations for complex data sets. They are used to convert sums of weighted inputs to outputs from the neurons, essentially acting as gatekeepers for signal flow through the network. Tanh and Sigmoid are common examples of activation functions.

An example of a common Activation function is the Sigmoid function (also known as the logistic function). This is a continuously increasing function, with elements of both linear and nonlinear (Haykin, 2009). The equation for the

Sigmoid function is below:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3,3}$$

Where x is the input of the neuron. The sigmoid function will take an input value and give an output value that will be between 0 to 1. The output of the Sigmoid function is shown in Figure 3.1. From the figure, it can be seen that the function takes an 'S' shape. It is mainly used in the output layer for binary classification problems.

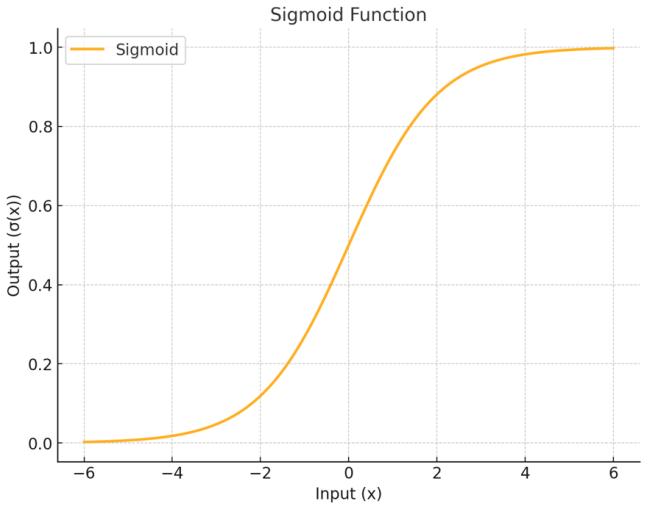

Figure 3.1 Sigmoid Function.

Another popular activation function is Tanh or hyperbolic tangent. It is a scaled and shifted version of the Sigmoid function, defined as:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3,4}$$

Inputs passed through the Tanh function are scaled to a range from −1 to 1, which results in outputs distributed around zero, which can be beneficial for gradient-based learning (Goodfellow et al. 2016). It is frequently used in hidden

layers of neural networks. Figure 3.2 shows the output of the Tanh function.

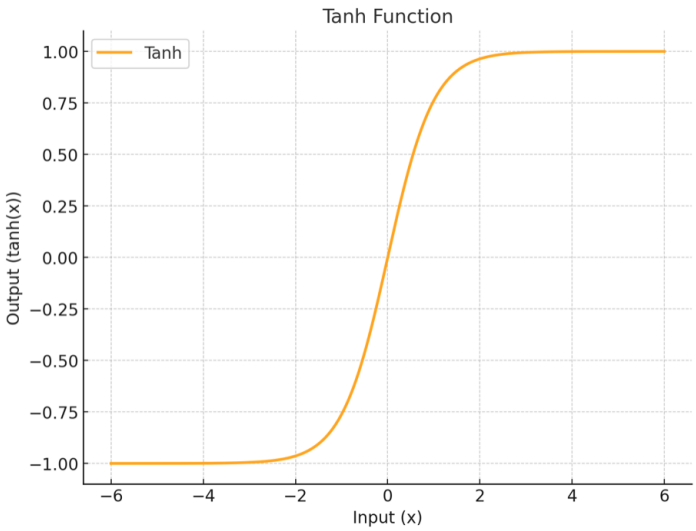

Figure 3.2 Tanh Function.

3.2. HE

HE represents a cryptographic approach that enables users to perform calculations directly on encrypted material, which precludes the need for decryption (Chen et al. 2018). It is a viable tactic for privacy preservation in an untrusted environment because it ensures that the private data is secured while it is being processed. As a result of the fact that it maintains data confidentiality, HE is particularly appropriate for MLaaS platforms. By using HE, one can easily implement sophisticated machine learning techniques while still safeguarding the privacy of the user.

A standard HE system is composed of representations of plaintext and ciphertext, the encryption and decryption processes, and the homomorphic operations. Encryption transforms plaintext to ciphertext using an encryption key, while decryption recovers plaintext from ciphertext using a decryption key. The distinguishing property of HE is the ability to perform arithmetic operations (such as addition, multiplication) directly on ciphertexts, producing an encrypted result that, when decrypted, matches the result of performing the same operation

on the plaintexts. For example, given $E(a)$ and $E(b)$, which represent ciphertexts of a and b respectively, then $E(a) \times E(b)$ is the ciphertext representation of $a \times b$.

HE schemes can be categorized into three classes depending on the type of the supported operations on the cyphertext mode: Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), and FHE (Acar et al., 2018). PHE allows an unlimited number of single type of operations to be carried out directly on encrypted data: either repeated additions or repeated multiplications. For example, the Rivest–Shamir–Adleman (RSA) scheme allows an unlimited number of multiplications, and the Paillier scheme allows an unlimited number of additions. A limitation of PHE is that it is not suited to a task that requires both addition and multiplication.

On the other hand, SHE facilitates addition and multiplication operations on the ciphertext mode. However, this is only possible for a limited number of operations before decryption becomes impossible. This is due to the fact that every operation increases the noise on the ciphertexts. SHE has more computing capabilities than PHE, and it also serves as a step toward FHE.

FHE extends both partially and somewhat Homomorphic Encryption schemes by enabling unlimited encrypted operations through both additive and multiplicative functions while preserving security. This makes it suitable for very complex workloads, including training and inference of machine learning models. The limitation is that it is computationally and memory intensive, which makes this type of scheme hard to implement in real world scenarios.

The CKKS scheme is employed in this work to provide homomorphic encryption capabilities, a scheme particularly suitable for computations involving real numbers and floating-point arithmetic. This scheme provides the capability to perform approximate numerical operations on data without decryption, making it ideal for applications in machine learning where precise calculations on non-integer values are essential, such as during the inference phase of neural networks with activation functions like Sigmoid and Tanh.

3.2.1. Mathematical Foundation of CKKS

The CKKS scheme is considered levelled FHE designed for approximate computations. Although it enables both additive and multiplicative operations on ciphertexts, its capabilities are limited by the growth of noise with each operation, making it suitable for computations involving a bounded number of operations (Mahmoud, 2025).

The CKKS scheme encodes data as polynomials over the ring $R = Z[X]/(X^N + 1)$, where N is a power of 2, and $X^N + 1$ represents the cyclotomic polynomial. This ring structure ensures efficient handling of arithmetic operations. The foundation of encrypted computation lies in representing polynomial coefficients as integers modulo a sufficiently large prime q. To encode plaintext data, a scaling factor Δ is introduced to maintain the precision of real or complex values during the transformation into polynomial coefficients.

CKKS encryption is based on the Ring Learning With Errors (Ring-LWE) problem (Cheon et al. 2017). It is done by encoding a plaintext polynomial and then adding ciphertext pairs (c_0, c_1) to encrypt the original data. A small error term is added in this encryption process using the public key to prevent the possibility of decryption by an adversary.

The scheme allows homomorphic addition and multiplication of ciphertexts. Additions are straightforward and performed by adding corresponding components of the ciphertexts. Multiplications, however, are more complex, involving polynomial multiplication and leading to noise growth. CKKS employs techniques such as modulus switching and rescaling to mitigate noise. Modulus switching reduces the modulus q, while rescaling divides the ciphertext by the scaling factor Δ, keeping noise within acceptable limits without significant loss of precision.

CKKS is a practical levelled FHE scheme used for privacy-preserving computations that require homomorphic operations on encrypted real or complex numbers. This scheme can be used in scenarios such as encrypted machine learning and privacy-preserving data analysis. The CKKS scheme is

both efficient and accurate, making it ideal for practical applications. Additionally, CKKS supports the multiplication of encrypted data, allowing it to be used in a wide range of applications.

For a more detailed explanation of the underlying HE scheme and its setting with more details on the mathematical foundations, please refer to (Cheon et al., 2017).

3.3. XAI

XAI refers to a set of techniques or approaches in the field of artificial intelligence that are aimed at providing some interpretability or transparency in a machine learning system from the perspective of a human observer. Models like neural networks are known to have a high accuracy rate but are sometimes criticized as “black boxes”. While these types of models can be accurate, they lack interpretability, which may hinder trust and adoption for important or sensitive applications, such as in the medical field. XAI allows one to understand how and why a model is making its decisions. Arrieta et al. (2020) define XAI as “the ability of a model to produce details or reasons that make its functioning clear or easy to understand for a given audience”. The demand for XAI exists for various reasons, such as allowing the interpretability of black box models, increasing trust by providing explanations for decisions, facilitating the diagnosis of errors and model debugging, and addressing ethical and societal issues by ensuring fairness, accountability, and transparency (Dwivedi et al., 2023).

3.3.1. XAI Types

In general terms, XAI can be divided into two broad categories: model-agnostic methods and model-specific methods (Kurek et al., 2023). Model-agnostic approaches, such as SHAP and LIME, can be applied to any model and do not require access to the model's internal workings. These methods are flexible and can provide post-hoc explanations for a wide range of models, but

they may be less faithful to the model's reasoning process. On the other hand, model-specific methods are designed to work with a particular algorithm or class of algorithms, and they aim to improve interpretability by incorporating it into the model's architecture. Examples of model-specific methods include decision trees or attention mechanisms. These methods are typically more faithful to the model's reasoning process but may have lower predictive power. These categories of methods are sometimes referred to as pre-hoc and intrinsic explainability.

3.3.2. SHAP

SHAP is one of the most widely used techniques to explain machine learning model predictions. At its core, SHAP assigns an importance value, known as a Shapley value, to each feature of a given instance to explain the model prediction. SHAP is based on the concept of Shapley values from cooperative game theory that provide a unified and consistent approach for attributing the output of a prediction among its input features (Lundberg & Lee, 2017). SHAP calculates the marginal contribution of each feature across all possible feature subsets by computing the change in the prediction when the feature is added or removed from the subset. SHAP can be used to provide local or global explanations. Local explanations refer to the specific predictions made by a model on a single instance, while global explanations refer to the overall behaviour of a model across a dataset.

SHAP was chosen as the explainability technique in this work due to its sound theoretical basis and its adaptability. The mathematically rigorous and consistent explanations it can provide make it a strong choice for high-stakes domains like the privacy-preserving dyslexia detection system in this thesis, where the issues of fairness, accountability, and transparency are most pronounced. In addition, SHAP is model agnostic, meaning it can be applied to any machine learning model without requiring any modification to the underlying algorithm. This makes SHAP an ideal candidate for our proposed framework, as the model operates on the homomorphically encrypted data.

3.4. CHALLENGES WITH NON-LINEAR FUNCTIONS IN HE

As mentioned above, HE schemes allow performing fundamental arithmetic operations (e.g., addition and/or multiplication) directly on encrypted data without the need to decrypt the message. While basic mathematical computations through these operations function adequately for simple expressions, they face significant obstacles when applied to non-linear activation functions necessary for successful ANN performance (Khan & Michalas, 2023).

In fact, one of the main challenges is related to the intrinsic complexity of non-linear functions. Non-linear activation functions include more advanced mathematical operations like exponentiation and division that HE schemes typically cannot support directly because each step in their approximation adds noise to the ciphertext. This often results in a rapid growth of ciphertext noise levels, which makes precision preservation during decryption problematic.

It is important to note that without non-linear activation functions, ANNs are no better than simple linear models, and thus they would not have much expressive power and would not be able to learn non-linear decision boundaries. Non-linear activation functions provide ANNs with the non-linearity needed to learn the complex patterns and relationships in the data. This implies that if we try to apply HE to an ANN model directly (without an efficient approximation to these functions), the model will perform poorly and will have low accuracy.

In order to be able to use HE with ANNs, this issue must be addressed by finding a way to approximate the function that is a good enough trade-off between accuracy and the resulting complexity. This is an active area of research in the broader area of PPML as of today.

3.5. PREVIOUS SOLUTIONS FOR NONLINEARITY IN HOMOMORPHICALLY ENCRYPTED ANN

In this section, we explore the current methods for approximating non-linear activation functions, which are essential for the performance of neural

networks. These approximations are particularly important when working with homomorphically encrypted data, where direct computation of non-linear functions is challenging. We focus on two widely used approaches: polynomial estimators and piecewise linear approximations. Each method provides a different balance between computational complexity and approximation accuracy, making them suitable for various encrypted computing scenarios.

3.5.1. Polynomial Approximation

The core idea of this method is to use a polynomial function as an approximation for non-linear activation functions such as Sigmoid and Tanh functions. Polynomial functions are faster to compute as they don't require using the exponent operation. The most important part is that it is easier to manage the noise budget because of the low complexity of operation. But there is a trade-off between the order of the polynomial function and the accuracy of the approximation. Higher-order polynomials can provide better behaviour of the non-linear functions, but they increase the complexity of the operations. In the HE senses, it increases the computation time and grows more noise in the encrypted domain. As such, the common choice is a second-degree polynomial that results in a trade-off between low computational power and ease of implementation. The drawback of using a second-degree polynomial is that it may not be able to give a good approximation of a non-linear function, which can lead to low precision of the function. The remedy for this is to use a higher-order polynomial for better approximation of the non-linear function, but the noise growth in the encrypted domain during the operations does not allow it.

In this work, we used second-degree polynomial estimators for Sigmoid and Tanh, which is a common way used in the literature. For the Sigmoid activation function, we use the polynomial approximation $Sigmoid_{poly}(x) \approx 0.5 + 0.197x - 0.004x^2$, while for the Tanh activation function, the approximation is defined by the equation $Tanh_{poly}(x) \approx 0.0 + 1.0x - 0.165x^2$. These forms were chosen to provide a balance between simplicity and

accuracy, allowing for efficient computation under homomorphic encryption constraints while maintaining reasonable approximations of the Sigmoid and Tanh curves. They were determined through empirical analysis of the functions. Figure 3.3 and Figure 3.4 show the polynomial approximations for Sigmoid and Tanh and comparison with the correct values for these activation functions respectively.

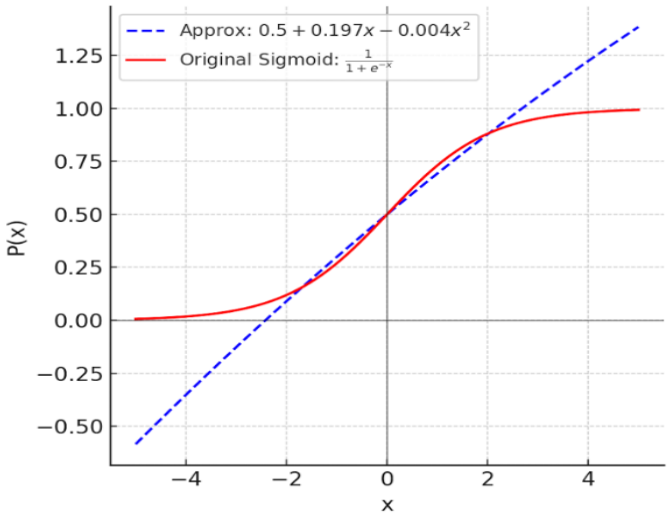

Figure 3.3 Sigmoid Polynomial Approximation and the Original Sigmoid.

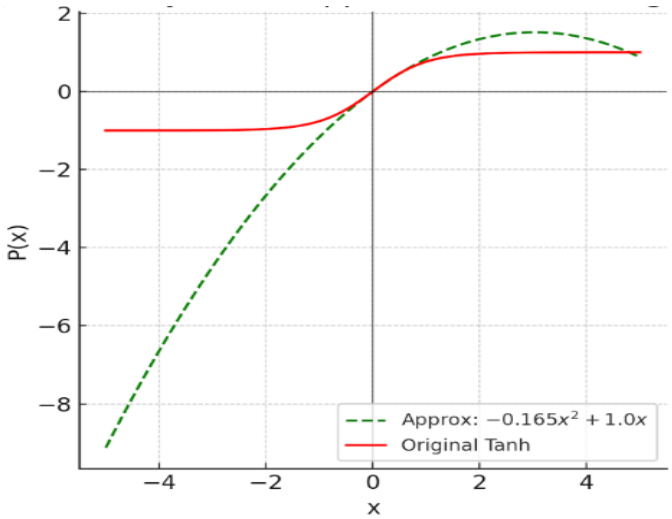

Figure 3.4 Tanh Polynomial Approximation and the Original Tanh.

3.5.2. Piecewise Linear Approximation

This approximation can be understood as approximating a non-linear function with multiple linear segments, each over some range of values. It makes this approach very amenable for encrypted domains, as it can often be a good trade-off between expressivity and computational complexity. This trade-off is because each linear segment can fit different regions of a non-linear function.

A key challenge of this approach is determining the appropriate range for each segment when working with homomorphically encrypted data. Because ranges cannot be discerned under encryption, determining optimal breakpoints for piecewise approximation becomes particularly challenging. One solution to this problem is to use the ranges calculated on plaintext data during training and use those ranges for inference on encrypted data. This method does not lose any approximation fidelity but does add in additional complexity and needs careful calibration.

Algorithm 3.1 details the computations of the nonlinear activation functions using piecewise linear approximations. The predefined breakpoints for the Sigmoid activation function were set to $[-6.0, -3.0, 0.0, 3.0, 6.0]$, while for the Tanh activation function, the breakpoints were $[-3.0, -1.0, 0.0, 1.0, 3.0]$.

Algorithm 3.1 Piecewise Linear Approximation for Nonlinear Activation Functions with HE

Input: Encrypted tensor x, breakpoints, and segment coefficients (a, b).

Output: Encrypted tensor with approximated nonlinear values.

1. Initialize encrypted result tensor $r \leftarrow 0$.
2. **For each segment** i **do**
3. Retrieve coefficients (a, b) and define the segment range.
4. Create a mask for x based on the segment range using approximate Heaviside functions.
5. Compute the segment approximation $s_i \leftarrow a.x + b$.
6. Update $r \leftarrow r + mask\,.\,s_i$.
7. **End for**

8. **Return** r.

Algorithm 3.1 approximates the nonlinear activation functions using a piecewise linear approach that is compatible with HE. The range of input values is divided into segments, each defined by a breakpoint range and a linear equation, which allows for efficient approximation of the nonlinear activation function when direct computation is not feasible due to the constraints of encrypted data processing.

The first step in this process is to initialise an output tensor to store the result. Each segment is then processed as a linear approximation on its assigned range. Since the data is encrypted and cannot be compared directly, we use Heaviside function approximation to create masks that identify which inputs are part of each segment.

For values less than or equal to the first breakpoint, the algorithm creates a mask to select those values. For values greater than the last breakpoint, a different mask is applied. For intermediate ranges, two masks are used to define the boundaries of the segment, ensuring that only values within the specified range are affected by the corresponding linear function.

Each segment's linear approximation can be computed directly using the multiplication and addition operators, which are both supported by FHE schemes. The masked segment is then added to the result, summing all of the contributions of each segment.

3.6. REGULATORY FRAMEWORKS AND DATA PRIVACY

As cloud adoption and the use of MLaaS continue to grow, organizations that operate in the cloud or deal with large amounts of sensitive data are looking for ways to better protect that data and prevent breaches. Addressing the privacy issues surrounding the collection and use of massive amounts of personal data, regulatory bodies around the world have created a number of data security frameworks.

These regulations aim to secure individuals' rights by holding organizations accountable for how they handle, store, and process personal information. They help to build trust between organizations and the individuals whose data they collect, use, and store. These frameworks create a set of rules and guidelines that organizations need to follow in order to better protect user privacy, especially when handling sensitive data. This is particularly important in industries such as healthcare and finance, where privacy breaches can have serious consequences. Additionally, these rules create transparency and privacy safeguards while providing clear standards and direction for organizations navigating the complexities of digital data privacy.

With the increasing reliance on cloud-based machine learning, organizations face the dual challenge of ensuring their models perform effectively while also meeting stringent data privacy regulations. In this section, we explore three critical regulatory frameworks that play a significant role in shaping data privacy practices: the GDPR in the European Union, HIPAA for the health sector in the United States, and the Law on the Protection of Personal Data, KVKK, in Türkiye.

3.6.1. GDPR

The General Data Protection Regulation, the GDPR, entered into force on the 25th of May 2018, and is one of the most stringent data protection laws globally. The EU drafted the GDPR to strengthen and unify data protection for individuals within the EU. The new law replaced the previous Data Protection Directive 95/46/EC, which had been in place since 1995 (European Commission, 2016). The GDPR was drafted in response to the challenges of digital technology innovation, online services expansion, and unprecedented volumes of personal data being transferred internationally, which had necessitated the need for a modernized and more robust legal framework (Wachter et al., 2017).

GDPR was put into practice to ensure the privacy rights by allowing people to control their personal data. The legislation standardizes how organizations should process, store, and manage information. Privacy and security principles

are paramount; they include transparency, accountability, and data security. GDPR aims to achieve data privacy by maintaining consumer trust. Transparency, accountability, and security are very essential for the use of cloud technologies and MLaaS (Goodman & Flaxman, 2017).

According to GDPR, personal data is any information that can be associated with a natural person. The information can be such that identifies the data directly or through reference. The following are types of personal information; basic information that includes names, addresses, and contact information, online information, such as Internet Protocol (IP) addresses, cookies, and device identities, sensitive personal data, for example, health records, biometrics, and racial or ethnic origin, and behavioural information, such as browsing history, geolocation data, and purchasing habits (European Commission, 2016).

The regulation is based on core data protection principles of lawfulness, fairness, transparency, purpose limitation, data minimization, accuracy, storage limitation, and confidentiality (ICO, 2018). The GDPR also contains provisions that specifically impact the application of ML and XAI in data processing. Relevant articles of the GDPR include:

- **Article 5** emphasizes principles such as transparency, purpose limitation, and data minimization, which directly pertain to the use of machine learning models that rely on large datasets (European Commission, 2016),
- **Articles 13 and 14** outline the requirements to inform data subjects about the collection and processing of personal data, including automated decision-making. This aligns with the need for XAI techniques to explain ML model outcomes (Goodman & Flaxman, 2017),
- **Article 15** grants individuals the right to access information about the data processed about them, including the logic of the processing related to automated decision-making. This necessitates the need for interpretability in ML models (Wachter et al., 2017),
- **Article 22** gives individuals the right not to be subject to decisions based solely on automated processing, including profiling, unless certain

conditions are met. This has significant implications for ML systems that are based on automated decision-making (Wachter et al., 2017).

Ensuring GDPR compliance in MLaaS and XAI is challenging due to the complexity and opacity of many ML models, especially deep learning (Goodman & Flaxman, 2017). A comprehensive approach involving technical, procedural, and regulatory measures is essential.

Firstly, transparency can be achieved by adopting XAI techniques that provide clear and understandable explanations of model decisions (Wachter et al., 2017). Secondly, privacy protection can be enhanced through data anonymization and pseudonymization, which mitigate the risks associated with processing sensitive data on cloud platforms and minimize the impact of potential data breaches (ICO, 2018). Thirdly, PPML methods such as HE and DP enable secure training and inference, safeguarding sensitive information even when models are trained or deployed on third-party cloud platforms (Goodman & Flaxman, 2017). Finally, compliance can be assured by conducting regular audits and maintaining comprehensive documentation of all data processing activities. These practices not only help demonstrate compliance with GDPR requirements but also embed privacy-by-design principles into ML solutions (Solove & Schwartz, 2018).

3.6.2. HIPAA

The United States (US) has not taken up GDPR but instead has a variety of its own data protection laws in place. There are some that have a significant overlap with the GDPR for various industries, such as HIPAA. The Health Insurance Portability and Accountability Act of 1996, more commonly referred to as HIPPA, was enacted to secure sensitive patient data and medical records (which are called Protected Health Information (PHI)) at a time when healthcare documentation was starting to be digitalized. HIPAA specifically targets healthcare industry data, while GDPR has a universal approach designed to protect data across different sectors.

HIPAA was passed in the US to serve two functions: portability of

insurance and standardization of electronic health care transactions. The adoption of electronic records has led to more concerns over the privacy of patient data. HIPAA's Privacy Rule and Security Rule established standards for the protection of certain health information in paper and electronic form. (HHS, 2003).

HIPAA is even more applicable in the current day, where EHR vendors, health care providers, insurance companies, etc., are all looking to use cloud and machine learning models to process and analyse patients' data, especially when using MLaaS platforms (HHS, 2013).

HIPAA is designed to ensure the privacy of PHI, covering any personally identifiable information related to an individual's health in the past, present, or future. The types of data protected under HIPAA include:

- Personal identifiers such as names, addresses, Social Security Numbers (SSNs), and phone numbers,
- Medical records including diagnosis, treatment history, and medication details,
- Financial information related to healthcare billing and payments,
- EHRs containing comprehensive patient health data.

HIPAA also states that covered organizations must take measures to ensure that PHI is secured at all stages of its collection, storage, and processing. The processing and storage of data on third-party servers, as is the case with cloud-based deployments, needs special attention (HHS, 2003).

There are many components of HIPAA that relate directly to the use of machine learning in the healthcare domain. For example, the Privacy Rule regulates how PHI is handled by healthcare entities. This includes a patient's right to access their health records and the right to know who can access their PHI, as well as to make choices about what information can be shared and how it can be used. In terms of machine learning, these rights have a direct implication on the use of health data as training sets for machine learning models. If training data is used, it must be de-identified unless consent is granted (HHS, 2013). The Security Rule can also be directly mapped to HIPAA

compliance in MLaaS. The Security Rule establishes that covered entities must implement policies and procedures to protect electronic PHI using administrative, physical, and technical security safeguards. This is particularly relevant for cloud-based MLaaS solutions, as HIPAA compliance would necessitate the use of strong encryption and access control policies to be deployed to ensure data security.

The Breach Notification Rule also states that the Department of Health and Human Services (HHS) and individuals must be notified of the breach in cases of compromising data confidentiality. This points out the need to use privacy-preserving methods such as homomorphic encryption and DP to avoid these breaches. Lastly, the Enforcement Rule states that there may be a penalty for violations of any of the prior rules. The penalty that a health care organization is likely to face may include monetary sanctions or criminal charges, depending on the severity of the violation (HHS, 2013).

Implementing machine learning solutions in the healthcare sector while adhering to HIPAA regulations presents significant challenges, particularly when dealing with complex models that may lack inherent interpretability (Goodman & Flaxman, 2017). To achieve compliance, machine learning systems in this domain must integrate several critical elements: XAI, data anonymization and de-identification, PPML techniques, robust security measures, and comprehensive documentation and compliance audits.

In healthcare settings where the outcome of automated decision-making may have significant implications on a patient's life, XAI techniques are needed to explain the reasoning behind a model's predictions in a way that is human-understandable and actionable. This is to provide clarity to our practices in line with the Privacy Rule's requirements of transparency and accountability and also provide further assurance to individuals whose information is being processed by these models that their rights would not be infringed upon as a result of these automated decisions. One way we can uphold this principle is by anonymizing and de-identifying our training data so that we can still use data for model training purposes while mitigating the risk of information exposure to the

greatest extent possible. De-identification would involve removing all personally identifiable information from data sets so that it is impossible to link the information to a specific patient (HHS, 2003).

In addition, healthcare providers can utilize PPML techniques such as homomorphic encryption that would allow computation on encrypted data, thereby preserving the confidentiality of sensitive health data for the models' users when engaging in MLaaS solutions. Implementation of these controls is vital for ensuring adherence to the Security Rule's standards. The Security Rule mandates stringent safeguards for electronic PHI, which go beyond encryption alone. Layered access controls and audit trails are additional components that ensure robust data security.

Moreover, it is essential for healthcare providers to have proper documentation of all data processing activities and regularly conduct compliance audits. This not only helps in demonstrating HIPAA compliance but also in identifying and rectifying potential vulnerabilities proactively, thereby mitigating the risk of non-compliance penalties (Solove & Schwartz, 2018). By adopting these practices, healthcare systems can harness the benefits of machine learning to enhance patient care while ensuring comprehensive protection of sensitive health data.

3.6.3. KVKK

While GDPR is the EU standard for data protection, the Turkish Grand National Assembly had passed a similar and also comprehensive data protection law in Türkiye for data protection through the KVKK (the long form of this term in English is the Law on the Protection of Personal Data) (Law No. 6698) and the law had entered into effect in April 2016. This law, which was inspired by the GDPR, is adapted to global data protection norms and practices as much as possible to the domestic Turkish legal and cultural environment. The purpose of this law is to provide personal data protection through privacy, transparency, and accountability, regarding the way the organizations keep the individuals' personal data.

The previous absence of a comprehensive regulatory regime on data protection created obstacles for Türkiye to effectively address privacy concerns related to the use of cloud computing and machine learning. The goal of the KVKK is to conform to international data protection standards while maintaining regulatory sovereignty over data processing activities within the national borders (KVKK, 2016).

KVKK is similar to GDPR in many ways, such as in the principles of data minimization, lawfulness, fairness, and transparency. However, there are some differences between the two regulations. KVKK has stricter data localization requirements and limits the transfer of personal data to other countries unless the receiving country has an adequate level of protection or explicit consent is given (KVKK, 2016). Enforcement in Türkiye is carried out by the Turkish Personal Data Protection Authority, which has the power to issue fines, suspend activities, and ensure compliance with the KVKK. Penalties under KVKK are generally lower than under GDPR, which can impose fines of up to 4% of global turnover. However, fines under KVKK can still be substantial, particularly for violations involving sensitive personal data (KVKK, 2016).

KVKK also includes provisions that are relevant to ML models and XAI, although there are some differences compared to GDPR. The differences are:

- **Article 5** (Legal Grounds for Data Processing): KVKK places more emphasis on explicit consent than GDPR, particularly in cases involving sensitive data. Consent must be explicit and clear. For ML models, this means if personal data is being used for model training, organizations must obtain consent,
- **Article 6** (Sensitive Data Protection): While similar to GDPR in requiring extra protection for sensitive data, KVKK places more emphasis on explicit consent and generally prohibits processing without consent, even in legitimate interest cases,
- Data Subject Rights (**Article 11**): KVKK also grants rights similar to GDPR, such as access, correction, and deletion of personal data. The key difference is the response time: organizations must reply to data subject requests within

30 days, which is a shorter timeframe compared to GDPR's one-month flexibility.

Organizations must consider several key steps to ensure KVKK compliance when implementing machine learning solutions, especially when using cloud-based platforms. Firstly, because of the KVKK's strict data localization rules, companies using MLaaS platforms hosted outside of Türkiye will need to either obtain explicit consent from data subjects or have in place robust data processing agreements to ensure compliance with KVKK's data localization requirements. Additionally, the strict requirement for explicit consent means that organizations will need to have a robust consent management system in place, especially when processing special data for machine learning purposes. Moreover, given the increasing emphasis of the Turkish Personal Data Protection Authority on enforcement of KVKK compliance, organizations are advised to conduct regular compliance audits to evidence their compliance efforts and to pre-emptively identify potential legal risks.

3.6.4. Ensuring Regulatory Compliance with Homomorphic Encryption and Explainable AI

GDPR, HIPAA, and KVKK regulations should be considered when applying Machine Learning, and the use of such technologies needs to adhere to data protection principles and ensure transparency. A potential solution to these problems would be to utilize HE to enable secure model inference combined with XAI approaches to enhance transparency. HE ensures that sensitive data remains protected throughout the ML pipeline in the inference phase. The proposed approach aligns with the requirements of GDPR and KVKK regarding the confidentiality and security of personal data processing, especially in cloud environments, where data breaches are a risk.

In addition to protecting data during inference, XAI techniques can provide explanations for automated decisions, thereby meeting transparency requirements outlined in GDPR's Article 22 and KVKK's Article 11. These articles address the issue of human interpretability of automated decision-

making, which becomes significant in fields such as healthcare and finance, where ML models' predictions and classifications have significant impacts. By adopting XAI, organizations can ensure that their ML models provide insights into how and why a specific decision was reached and thus meet the underlying GDPR and KVKK principles, thus earning data subjects' and regulators' trust.

Another aspect of compliance is to address the issue of data consent. If the data is to be collected and used in plaintext (in the training phase), explicit consent must be received from the subjects to which the data belongs, especially when the data are sensitive. Furthermore, anonymization of the data must be performed before using such data on the developed systems in the cloud for training or inference. This removes all identification marks, thus making re-identification challenging and aligning with the data minimization requirements of GDPR, HIPAA, and KVKK.

CHAPTER 4

4. METHODOLOGY

4.1. MOTIVATION FOR ANN-BASED ESTIMATORS

The motivation behind adopting ANN-based estimators for approximating non-linear activation functions lies in achieving a flexible, uniform framework for handling various activation functions, such as Sigmoid and Tanh, without altering the underlying model structure. Traditional approaches often rely on separate, function-specific techniques, such as polynomial or piecewise linear approximations, each requiring distinct setups and having limited ability to accurately capture the complex behaviour of non-linear functions.

A key advantage of our ANN-based estimator is its simplicity. Switching to a different activation function only requires retraining, with no need to modify the network architecture. Specifically, to approximate a new non-linear activation (e.g., ReLU or SoftMax instead of Sigmoid/Tanh), we simply recollect the pre-activation outputs (from the main ANN's hidden layer) and pair them with the corresponding target post-activation values for the new function. This effectively yields a new "training set" for the single-layer ANN estimator, reflecting how the activation function transforms inputs over its valid range. Because the estimator learns a general mapping from pre- to post-activation values, no special polynomial expansions, breakpoints, or manually tuned approximations are required. Only the reference dataset—i.e., pairs of hidden-layer outputs and their correct activation values—must be regenerated, making the process both flexible and efficient. This design also shortens the development cycle: adjusting the estimator for another activation function requires only standard regression training on the updated dataset, without complex modifications to the neural network architecture or encryption parameters.

ANN-based estimators can additionally capture subtle, nuanced behaviours of non-linear activation functions by learning adaptively from training samples over the full function range. This allows them to model smooth transitions and inflection points more precisely than simpler polynomial or piecewise linear methods might miss. Unlike low-degree polynomial approximations—which often struggle with accuracy—or piecewise linear approximations—whose precision can vary across breakpoints—the proposed estimator can learn closer to the actual shape and characteristics of the function. This adaptability is especially beneficial in HE settings, where consistent high-fidelity approximations with minimal noise accumulation are crucial.

4.2. OVERVIEW OF THE PROPOSED SOLUTION

To provide a clear understanding of the proposed solution, Figure 4.1 illustrates the dataflow diagram. The diagram illustrates four preparatory steps required to create the ANN-based estimator. First, the range of input values for which activation functions need to be approximated is determined. Next, the training dataset is prepared by calculating the activation function for sample points within the defined range. This dataset forms the basis for training the ANN estimator, capturing the activation function's behaviour over the relevant range. The ANN is trained in plaintext mode, thereby avoiding the noise buildup that would occur with homomorphically encrypted calculations. Once trained, the ANN estimator is utilized during encrypted inference, effectively addressing the core challenge that this approach seeks to resolve.

To give an example of a use case for the discussed solution in a real environment, Figure 4.2 shows the communication diagram of a healthcare use case for dyslexia detection. The main actors for the presented communication diagram are four: patient, healthcare centre, CSP, and Key Management Service (KMS). The patient is the subject whose private data will be used by the system. They are responsible for their private data and holds the private key needed to decrypt their data. In other words, the patient will maintain full control over their

data, which grants privacy protection and also data protection law compliance. The healthcare centre is the medical centre providing the necessary medical services for the patient in question. The healthcare centre validates system predictions through additional medical testing and statistical evaluations. For this reason, the health care centre will be able to check if the predictions made by the system are medically relevant and ethical. The CSP is the cloud server that hosts encrypted machine-learning models and performs calculations on encrypted data. The CSP is assumed to only process encrypted inputs and output encrypted results with no possibility to access the original input data or the private key. The Key Management Service (KMS) is a service responsible for the creation and management of the key pair used to encrypt and decrypt data.

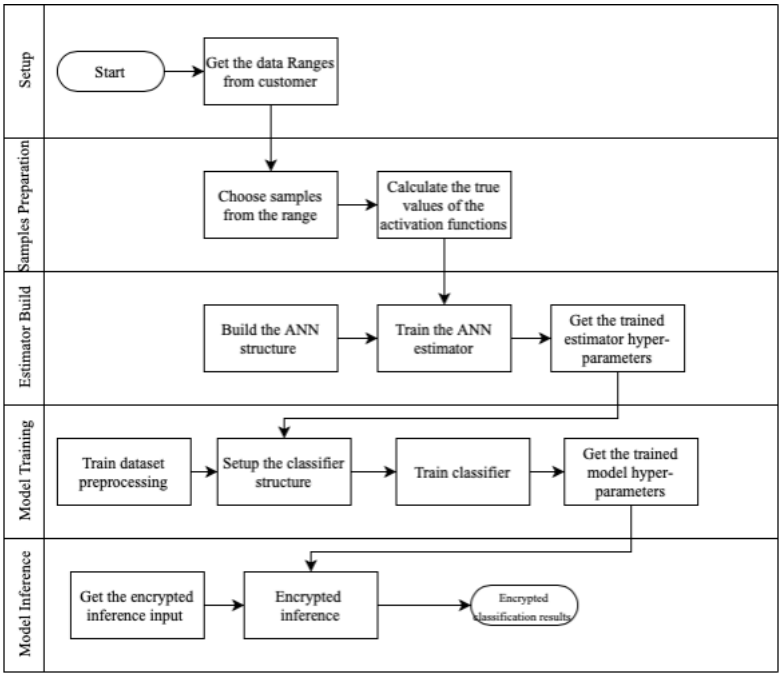

Figure 4.1 The Dataflow Diagram of the Proposed Work.

During the system preparation phase, the healthcare centre preprocesses the training dataset and securely shares it with the CSP. The CSP uses this dataset to train the ANN model. To address the challenge of non-linear activation functions in the HE inferences, the CSP also trains an ANN-based activation

function estimator, which will play a crucial role during the homomorphic inference phase.

The patient retrieves key pairs from the KMS before initiating the HE inference process. The patient gathers and preprocesses the QEEG data after acquiring the keys, then encrypts the data using the public key (Pk) and sends it to the CSP. The patient receives encrypted categorization results from the CSP's execution of the HE inference. The patient obtains the categorization outcomes by decrypting the results using their private key (Pr).

To enable the explainability option, the patient requests an explanation of the results from the CSP. The CSP calculates the SHAP mean values in encrypted mode. The patient then receives these encrypted SHAP values back and uses their Pr to decode them. The patient can safely send the decrypted SHAP values to the medical facility, which can provide a medical opinion based on the findings if they need expert interpretation.

To validate the practical viability and scalability of the client-server communication model, a proof-of-concept serverless architecture was deployed. This implementation demonstrates how MLaaS providers can host the trained ANN estimators to securely perform activation function approximation in an encrypted environment. This architecture, which ensures the server operates stateless and never accesses plaintext data, is detailed in Appendix A.

4.3. DESIGN OF THE MAIN ANN

The Main ANN serves as the primary classification model in the proposed solution, designed to perform predictive tasks while maintaining efficiency and compatibility with HE. Given the computational constraints associated with HE, the architecture is intentionally shallow to enhance stability, reduce processing overhead, and mitigate noise introduced during encrypted inference. From an architecture point of view, Glorot & Bengio (2010) highlighted the difficulties associated with training deep feedforward networks, especially when activation functions like Sigmoid are employed. Their findings emphasize that shallower

architectures are often better suited for tasks where computational simplicity and stability are critical. The following sections describe the design of the Main ANN for two specific tasks: MNIST digit classification and dyslexia detection using QEEG data. It is important to mention that the designs of the Main ANN for both classification tasks were established through extensive experimentation, where various architectural configurations, optimization strategies, and regularization techniques were systematically evaluated to achieve the highest possible performance while maintaining efficiency and robustness in both plaintext and encrypted inference settings.

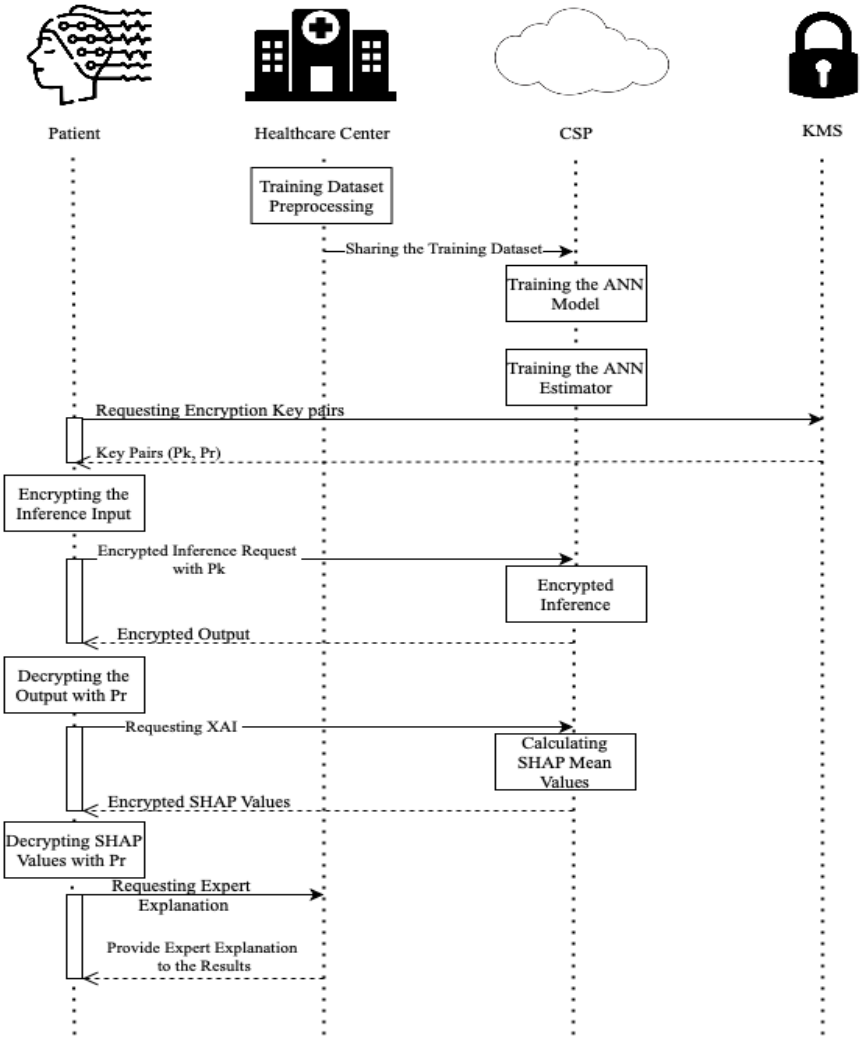

Figure 4.2 Communication Framework for the Proposed Solution.

4.3.1. Main ANN for MNIST Classification

The Main ANN for MNIST is a feedforward neural network that's specifically structured for efficient handwritten digit recognition while keeping computational demands relatively low. It has three layers: the input layer, a single hidden layer, and the output layer. The input layer deals with flattened 28×28 grayscale images, and it has 784 features, providing a structured representation of image data. The hidden layer, with 64 neurons, acts as a feature extractor that transforms the input data into a more meaningful representation. Non-linearity is introduced in the hidden layer via Sigmoid or Tanh activation functions, which helps the model capture more complex relationships within the data. To ensure stable training and avoid gradient issues, this layer also uses Xavier initialization for its weights (Glorot & Bengio, 2010). Xavier initialization sets the weights to be drawn from a zero-mean distribution with a variance that is the inverse of the number of input and output units.

The output layer has 10 neurons, one for each of the 10 classes (digits) of the MNIST dataset. As no explicit activation function is mentioned for the output layer, and since the CrossEntropyLoss function is used during training, it is equivalent to applying the SoftMax function to obtain the class probabilities.

The model is trained with the Adam optimizer with a learning rate of 0.001 and L2 regularization with weight decay of 1e-4 to avoid overfitting. The model is trained in plaintext to learn the optimal parameters for the model before performing encrypted inference. A shallow model was chosen as deeper neural networks require more computation time and are more noise sensitive in homomorphic encryption.

4.3.2. Main ANN for Dyslexia Detection

For dyslexia detection using QEEG data, the Main ANN is structured to classify individuals based on neural activity patterns. Unlike the MNIST classification task, which involves image-based data, this model operates on structured numerical features extracted from QEEG signals. The input layer

consists of 70 neurons, corresponding to the extracted QEEG features. To ensure consistency in feature distribution, the input data is normalized using StandardScaler, which standardizes the input values, aligning them to zero mean and unit variance.

The hidden layer comprises 128 neurons and serves as a fully connected feature transformation layer. The Sigmoid activation function is applied at this stage to capture non-linear relationships in the data, enhancing the model's ability to distinguish between dyslexic and non-dyslexic individuals. Similar to the MNIST model, Xavier initialization is employed to stabilize training, ensuring an appropriate weight scale.

The last output layer has only one neuron, which is used for a binary classification task. There is no explicit activation function in this layer, as the CrossEntropyLoss function applied on the output automatically computes the probabilities while training the model.

The model is trained using the Adam optimizer with a learning rate of 0.001. L2 regularization with weight decay of 1e-4 is also added to the model to avoid overfitting, and a learning rate scheduler (StepLR) is added, which decreases the learning rate by a factor of 0.1 every 20 epochs to help in stable convergence.

4.4. DESIGNS OF THE ANN ESTIMATORS

The design of the ANN estimator is tailored to support the distinct requirements of the two datasets used in our experiments: MNIST and QEEG-based dyslexia detection. Both designs focus on approximating non-linear activation functions within encrypted domains while ensuring computational efficiency and stability.

To comprehensively assess the performance of the estimator, three distinct experiments were carried out. The first experiment focused exclusively on evaluating the ANN-based estimator in isolation, without embedding it within a classification task. In this independent setting, the estimator was implemented

with a deliberately simple yet effective architecture, carefully designed to approximate the Sigmoid and Tanh activation functions with efficiency. The network architecture begins with a single input neuron, followed by a hidden layer composed of 16 neurons employing the ReLU activation function to introduce the required non-linearity. To enhance generalization and mitigate the risk of overfitting, a dropout layer with a dropout rate of 0.1 was incorporated into the design. The final output layer adopts a linear activation function, thereby ensuring a smooth and continuous approximation of the target functions. For training, the Adam optimizer was utilized with a learning rate fixed at 0.001, while the MSE served as the loss function. The training process was executed over 50 epochs using a batch size of 32, which was selected to strike a balance between computational efficiency and convergence stability. This lightweight yet effective configuration enables the estimator to approximate non-linear activation functions with accuracy, while at the same time maintaining efficiency and practicality for homomorphic encryption applications.

For the MNIST dataset, the ANN estimator is implemented as a single-layer regression model designed to approximate activation functions efficiently. It processes the 64-dimensional pre-activation outputs from the primary ANN's hidden layer, converting them into post-activation values using a fully connected layer. To enhance stability and prevent gradient vanishing or exploding, weights are initialized using the Xavier method. By adopting a single-layer structure, the estimator minimizes computational overhead and mitigates noise accumulation in the HE environment, aligning well with the resource constraints of encrypted inference.

For the dyslexia detection task, the ANN estimator consists of a single fully connected layer with 128 neurons, matching the size of the main ANN's hidden layer. This structure is specifically optimized to approximate the non-linear Sigmoid activation function in an encrypted inference setting. During inference, forward hooks are used to capture pre-activation outputs from the main ANN's forward pass. These pre-activation values are then normalized using MinMaxScaler to align with the appropriate input range for training, while

the corresponding post-activation outputs serve as ground truth labels. Through this approach, the estimator is trained with data aligned to practical deployment contexts, ensuring that the variability embedded in the computations of the main ANN is effectively preserved.

To further improve the precision and stability of the training process, the Adam optimizer is used with a learning rate of 0.0001. Weight clipping is also used to clip the estimator's weights to a maximum value of ±5.0, to ensure numeric stability and HE compatibility. After some iterations, the number of layers, learning rate, and weight clipping thresholds of the estimator were adjusted to achieve a precise and computationally efficient approximation of the activation functions.

As shown in Figures 4.3 and 4.4, these custom ANN estimator designs prioritize capturing the relationship between pre-activation and post-activation values for Sigmoid and Tanh activation functions respectively, which allows for accurate approximations in an encrypted inference pipeline.

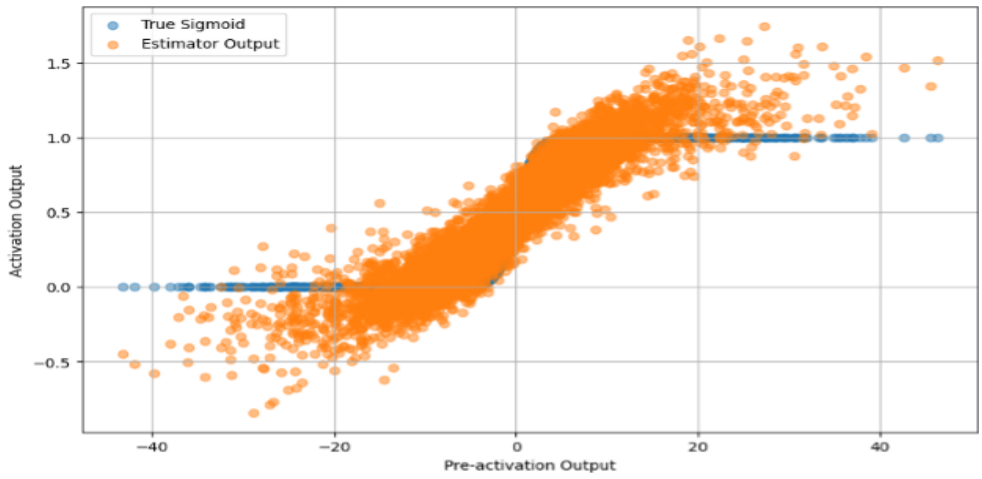

Figure 4.3 Sigmoid ANN Estimator and the Original Sigmoid.

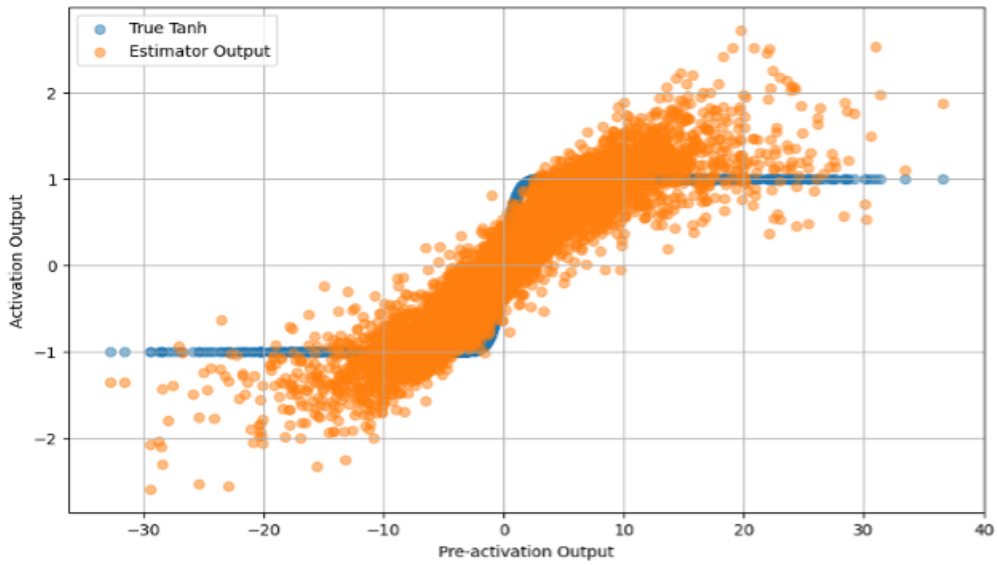

Figure 4.4 Tanh ANN Estimator and the Original Tanh.

4.5. HOMOMORPHICALLY ENCRYPTED INFERENCE

Algorithm 4.1 summarizes the steps related to the homomorphically encrypted inference phase.

Algorithm 4.1 Homomorphically Encrypted Inference

Input: Encrypted input data x, trained ANN and ANN estimator parameters $(W_1, b_1, W_2, b_2, W_e, b_e)$, encryption context.

Output: Encrypted inference result y and computation time.

1. **Step 1: Encode Model Parameters**
2. Extract trained weights and biases:
3. (W_1, b_1): First layer weights and biases of the main ANN.
4. (W_2, b_2): Second layer weights and biases of the main ANN.
5. (W_e, b_e): Weights and biases of the ANN estimator.
6. Encode each parameter as plaintext tensors compatible with the encryption context.
7. **Step 2: Compute First Layer Transformation**
8. Perform the first linear transformation:

9. $z_1 \leftarrow W_1 \,.\, x + \, b_1$ (element-wise operations on encrypted data).
10. z_1 represents the encrypted pre-activation values of the hidden layer.
11. Step 3: Apply Activation Function Using ANN Estimator
12. Use the ANN estimator to approximate the activation function:
13. $a_1 \leftarrow W_e \,.\, z_1 + \, b_e$ (encrypted approximation of Sigmoid or Tanh).
14. a_1 represents the encrypted post-activation values of the hidden layer.
15. Step 4: Compute Second Layer Transformation
16. Perform the final linear transformation:
17. $y \leftarrow W_2 \,.\, a_1 + \, b_2$ (encrypted transformation for prediction).
18. y represents the encrypted final output of the ANN.
19. Step 5: Return Results
20. **Return** encrypted output y and total computation time.

The encrypted inference process begins with the encrypted input data, represented as X_{enc}, which is derived by encoding and encrypting the original plaintext input X using a homomorphic encryption scheme. This encrypted input X_{enc}, allows computations to be carried out directly on encrypted data without decryption, preserving the privacy of sensitive information throughout the inference process.

Next, the trained model parameters (weights and biases) from both the main ANN and the ANN estimator are converted into plaintext tensors (indicated with a "pt" superscript), rendering them compatible with homomorphic encryption operations. Let n be the number of input features, h be the number of hidden units in the main ANN, and o be the number of output classes or continuous outputs. Weights and biases in the first layer of the main ANN are $W_{fc1} \in R^{h \times n}$ and $b_{fc1} \in R^h$, respectively. These parameters are applied to the encrypted input X_{enc} to compute the encrypted pre-activation values of the hidden layer using the equation:

$$h_{enc} = X_{enc} \,.\, W_{fc1}^{pt} + b_{fc1}^{pt} \qquad (4,1)$$

The encrypted pre-activation values h_{enc} are then processed through the ANN estimator, which approximates the activation function, which may be Sigmoid or Tanh based on the trained datasets. This involves applying the estimator's weights $W_{est} \in R^{h \times h}$ and biases $b_{est} \in R^{h}$, yielding the encrypted post-activation values:

$$h_{est_{enc}} = h_{enc} \,.\, W_{est}^{pt} + b_{est}^{pt} \qquad (4,2)$$

This representation effectively approximates the activation functions within the homomorphic encryption context, avoiding direct computation of the function and ensuring efficient encrypted operations.

Finally, the encrypted post-activation values h_{est_enc} are passed through the second layer of the main ANN, which uses weights $W_{fc2} \in R^{o \times h}$ and biases $b_{fc2} \in R^{o}$ to produce the encrypted final output:

$$y_{enc} = h_{est_{enc}} . W_{fc2}^{pt} + b_{fc2}^{pt} \qquad (4,3)$$

This output, y_{enc}, represents the main ANN results, which is concerned with the classification problem that the solution try to predict.

For a deeper understanding of the inference process and how it is influenced by the bias-variance trade-off, including its impact on model performance and generalization, please refer to (Glorot & Bengio, 2010).

4.6. EXPLAINABILITY IN PPML

Algorithms 4.2 and 4.3 outline the procedures for achieving explainability through the SHAP methodology for the proposed homomorphically encrypted inference.

Algorithm 4.2 Cloud-Side: Homomorphic Encryption Inference for SHAP-like Values

Input: Encrypted input enc_c, model M, activation estimator ε, number of

features d, perturbations per feature N.

Output: Encrypted SHAP-like vector $enc\_s \in C^d$

1. $enc\_y \leftarrow M(enc\_c)$
2. **For** $i = 1$ to N **do**
3. $list \leftarrow [\,]$
4. **For** $j = 1$ to N **do**
5. $enc\_x_{(i)}^{(j)} \leftarrow Perturb(enc\_x, i)$
6. $enc\_y_{(i)}^{(j)} \leftarrow M(enc\_x_{(i)}^{(j)})$
7. $\Delta_{(i)}^{(j)} \leftarrow enc\_y_{(i)}^{(j)} - enc\_y$
8. $Append\ (list, \Delta_{(i)}^{(j)})$
9. **End For**
10. $enc\_s[i] \leftarrow \frac{1}{N}\sum_{j=1}^{N} \quad list[j]$
11. **End For**
12. **Return** enc_s

Algorithm 4.3 Client-Side: Decryption, Aggregation, and Visualization

Input: Encrypted SHAP-like values enc_s, Private key Pr, Number of features d.

Output: Plaintext SHAP-like values and visualization.

1. $s \leftarrow Decrypt(enc\_s, Pr)$
2. **For** $i = 1$ to d **do**
3. $\underline{s_i} \leftarrow |s_i|$ (mean absolute attribution)
4. **End for**
5. $PlotBarChart(\underline{s_1}, \underline{s_2}, \dots, \underline{s_d})$
6. **Return**: $\underline{s}$ and the visualization

Algorithms 4.2 and 4.3 operate as follows. Let $x = (x_1, x_2, \dots, x_d)$ denote the plaintext feature vector and $\tilde{x}$ its CKKS ciphertext, with $d = 70$ in our study (the pre-processed QEEG features). The cloud first computes the baseline

ciphertext prediction $\tilde{y} = M(\tilde{x})$. For each feature i and perturbation j ($j = 1, \dots, N$), the i-th slot of $\tilde{x}$ is masked to form $\tilde{x}_{(i)}^{(j)}$ and the encrypted output $\tilde{y}_{(i)}^{(j)}$ is evaluated. The marginal contribution $\Delta\tilde{y}_{(i)}^{(j)} = \tilde{y}_{(i)}^{(j)} - \tilde{y}$ is averaged homomorphically to yield the encrypted SHAP-like value $\tilde{s}_i$.

The client decrypts $\tilde{s}_i$, forms $\underline{s_i} \leftarrow |s_i|$, and plots the vector $(\underline{s_1}, \dots, \underline{s_d})$ as a bar chart, obtaining an interpretable ranking of all 70 QEEG features without ever exposing raw data or intermediate predictions to the cloud.

CHAPTER 5

5. EXPERIMENTAL SETUP

5.1. EXPERIMENTS OVERVIEW

The efficiency of the proposed solution was evaluated through a series of controlled experiments. Figure 5.1 depicts a graphical representation of the workflow, showing the main steps of the experimentation process.

Figure 5.1 Overview of the Proceeded Experiments.

The first experiment conducted in the context of the work aimed to create an ANN-based estimator to approximate the Sigmoid and Tanh activation functions for privacy-preserving machine learning. The proposed method was compared with the two most popular approaches for activation function approximation (polynomial and piecewise linear estimator). Accuracy was the main metric to evaluate the performance of the estimator. Instead of a classification task, the first experiment was conducted only to create and test the estimators in the homomorphically encrypted environment. After the estimators were computed in encrypted mode, the results were decrypted, and the accuracy was evaluated using the MSE metric. No dataset was used in this experiment. However, to train the ANN estimators, 10,000 random floating-point numbers in the range of [-1,000, 1,000] were picked, and the Sigmoid and Tanh activation

functions were computed for these values and used as a training dataset. For the evaluation of the proposed solution and its comparison with other estimators, 100 random values were selected, and the experiment was repeated. Computation time and MSE metrics were derived for all the approaches, demonstrating the efficiency and competitiveness of the proposed solution compared to other methods.

The second experiment aimed to conduct a performance assessment of activation function estimators to determine their suitability in homomorphically encrypted inference using the standardized MNIST dataset. The dataflow diagram in Figure 5.2 illustrates the experimental setup. As depicted, six different inference tests were conducted to assess the effectiveness of polynomial, piecewise linear, and ANN-based estimators for both Sigmoid and Tanh activation functions. These tests were grouped into two categories—one for Sigmoid and the other for Tanh—each utilizing a dedicated ANN structure optimized for approximating the respective activation function. In addition, our proposed ANN estimator was also compared with similar approaches from the literature by integrating it within a CNN-based structure, enabling a broader performance comparison across different network architectures. This structured approach ensured a comprehensive evaluation of each estimator's performance, enabling a direct comparison of their accuracy and computational efficiency in an encrypted inference environment.

The third experiment focused on applying HE to a real-world healthcare scenario: dyslexia detection using QEEG data. This study aimed to evaluate the feasibility of encrypted inference for medical machine-learning applications while ensuring privacy preservation. The experiment leveraged an ANN trained on QEEG data in plaintext mode, followed by encrypted inference employing an ANN-based estimator for the Sigmoid activation function. Highlighting a key capability of the proposed framework, this experiment also integrated XAI techniques. Specifically, SHAP values were incorporated to provide insights into the model's decision-making process, demonstrating how to ensure interpretability within an encrypted environment.

The final experiment extended our evaluation by applying the proposed ANN-based activation function estimator to the UCI Heart Disease dataset as a case study. Unlike the QEEG dataset used in the dyslexia detection experiment—which consists of high-dimensional neurophysiological signals with temporal and frequency-based complexity—the UCI Heart Disease dataset provides structured, tabular clinical records with demographic and physiological attributes such as age, cholesterol, and blood pressure. This fundamental difference in data modality and statistical distribution allowed us to assess the estimator's adaptability beyond biomedical signal data. By testing on tabular clinical features rather than QEEG signals, the experiment demonstrated the robustness and flexibility of the estimator across heterogeneous data types. The UCI case study was therefore chosen not only because of its wide use as a healthcare benchmark but also to emphasize the generalizability of our approach, confirming that the proposed estimator is effective across diverse real-world problem settings and not confined to a single application domain.

5.2. DATASET DETAILS

In this study, two distinct datasets were utilized to evaluate the proposed privacy-preserving machine learning approach: the MNIST dataset and a QEEG dataset for dyslexia detection. These datasets encompass both standard benchmarks and real-world biomedical data, allowing for a comprehensive assessment of encrypted inference performance.

5.2.1. MNIST Dataset

For our second experiment, we used MNIST dataset, a standard dataset for benchmarking, which is used to validate machine learning models, in particular for recognizing handwritten decimal digits. MNIST is a set of greyscale images of handwritten digits 0-9 created by LeCun et al. (1998). It is one of the standard datasets for benchmarking the different approaches in machine learning. Sample data from MNIST dataset is presented in Figure 5.3 (Activeloop, n.d.).

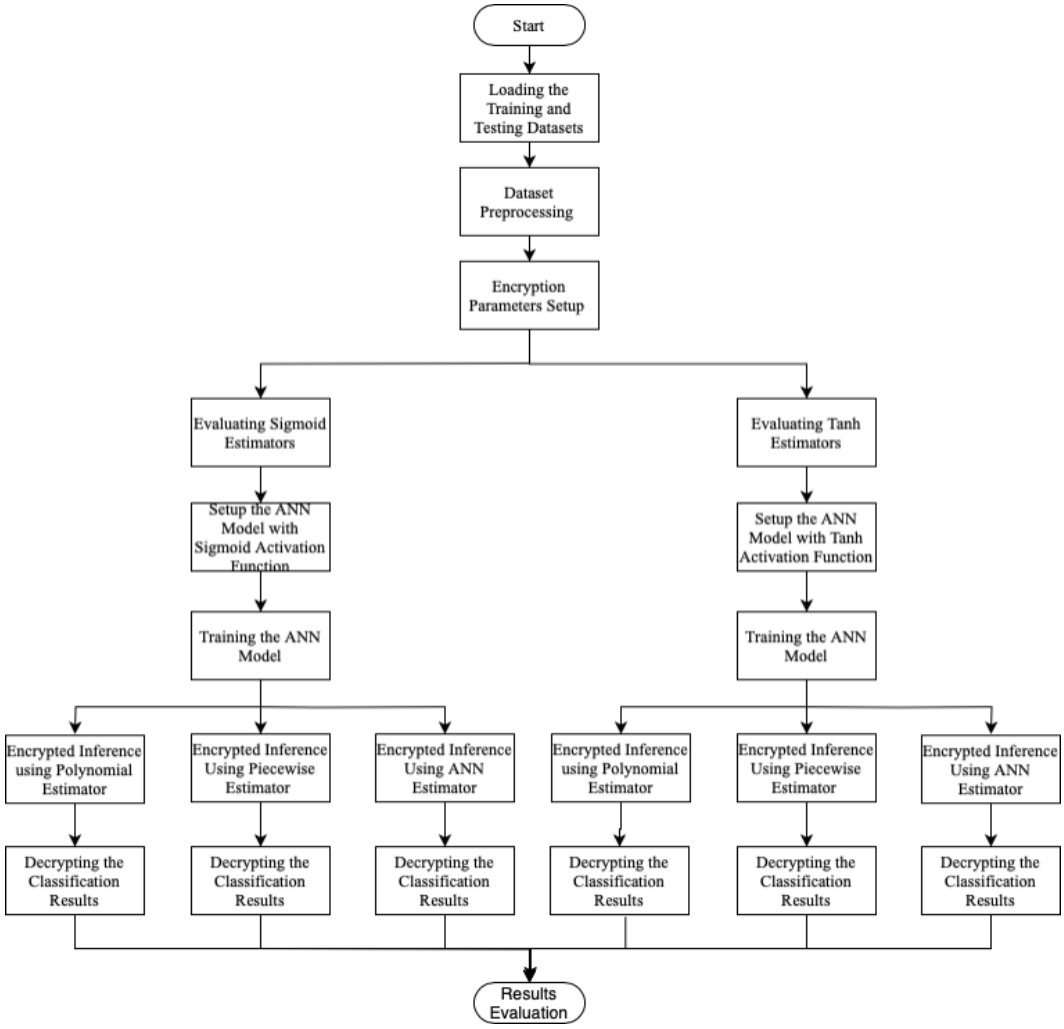

Figure 5.2 Tests Details of the HE Inferences Tests on MNIST Dataset.

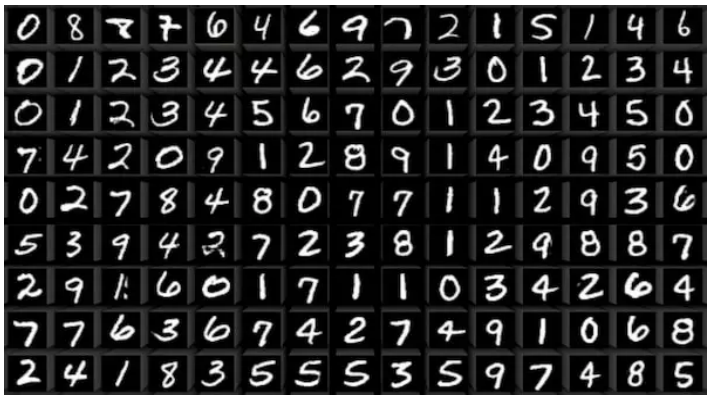

Figure 5.3 Examples of Handwritten Digits from the MNIST Dataset.

The MNIST dataset used for training and inference has 70,000 handwritten digit images in total: 60,000 training samples and 10,000 test samples. Each image is a 28×28-sized digit image which when flattened becomes a 784-dimensional vector. These flattened images are the inputs to the ANN models. In this work, we used all 60,000 training samples to train the main ANN and the ANN estimator.

For testing, we used the standard MNIST test set of 10,000 images. However, during encrypted inference, we sampled a fixed number of images from this dataset. Since the encrypted inference takes much longer than non-encrypted inference and testing, we capped the number of test samples for encrypted inference to 100 samples per class to strike a trade-off between coverage of the test set and overall computation time. Since there are 10 different classes (digits), this gives us a test set of 1000 samples. This number is enough to provide evaluation on all the digit classes and, at the same time, small enough for encrypted computation.

As with many image datasets, we pre-processed the MNIST data by reshaping it to form a tensor of data and normalizing it. We converted each image to a normalized image with a mean of 0.1307 and a standard deviation of 0.3081, which are standard MNIST dataset values. Normalization of data is an important step in stabilizing the training as it scales the input in a way that neural networks converge faster and more reliably. We also flattened each image to a 784-dimensional vector to match the dimension of the input layer of the main ANN.

5.2.2. QEEG Dataset for dyslexia detection

While the MNIST dataset provided a standardized benchmark for evaluating the performance of our ANN models in an image classification task, we further extended our analysis to a real-world application by leveraging a dataset focused on dyslexia classification using QEEG data. This dataset, derived from neurophysiological measurements, allowed us to explore the applicability of our approach in a privacy-sensitive domain, where homomorphic encryption plays a crucial role in preserving data confidentiality.

This database was acquired from 200 human subjects. There were 100 children with dyslexia and 100 typically developing children. The participants with dyslexia were confirmed by scores on the Test of Integrated Language and Literacy Skills (TILLS). All of the participants in the data gathering phase were acquired through online recruitment fliers. The participants were required to meet a number of eligibility criteria, such as being from a middle socioeconomic status, which was confirmed through parental questionnaires regarding their income, education, and job, not being on medication, and not having any other neurological or cognitive disorders except dyslexia. The average age of the dyslexic group was 8.80, and the control group was 8.83 years old.

The QEEG dataset employed in this study was obtained under an approved protocol by the Yeditepe University Ethics Committee, with the clinical trial registered at the Türkiye Pharmaceuticals and Medical Devices Agency (number: 71146310-511.06, 2.11.2018). In addition, before enrolling, the participants and their legal guardians were informed about the study procedures and gave written informed consent for participation. The authors do not have any direct relationship with data acquisition. It should be noted that the data acquisition process itself was outside the scope of this research; our work focused solely on utilizing this ethically approved dataset for developing and evaluating the proposed privacy-preserving framework.

The EEG signals of the dataset were collected for a period of 3–4 months as the subjects attended daily neurofeedback sessions. The recorded EEG signals were collected with the EMOTIV EPOC-X headsets with an internal sampling rate of 2048 Hz per channel. The headset was calibrated before collecting the data to ensure clean recordings. After the aliasing and artifacts were removed from the recordings, the sampling rate was down sampled to 128 Hz per channel to make the signal processing more efficient. The Fast Fourier Transform (FFT) was used to decompose the signals into frequency bands. The used EEG frequency bands were theta (4–8 Hz), alpha (8–12 Hz), beta-1 (12–16 Hz), beta-2 (16–25 Hz), and gamma (25–45 Hz). It was not possible to access the delta band (0–4 Hz) as a result of the hardware limitations. Badcock et al. (2013)

assessed EMOTIV devices and provided empirical proof of their efficacy in the collection of QEEG data.

The dataset comprised 70 extracted features derived from 14 EEG channels: AF3, F3, F7, FC5, T7, P7, O1, O2, P8, T8, FC6, F8, F4, and AF4. Figure 5.4 illustrates the scalp distribution of the electrodes, as described in (Pachì et al., 2023). For each channel, the power spectral density was computed across the five EEG frequency bands, forming the final feature set (Eroğlu et al., 2022). EEG data were collected under two conditions: a resting-state condition, where participants maintained open eyes for two minutes to record baseline brain activity, and a neurofeedback session, which consisted of a 30-minute training protocol designed to improve neural regulation. To ensure consistency in biomarker analysis, the study focused exclusively on resting-state EEG data, to prevent variations introduced by neurofeedback sessions from influencing the classification process.

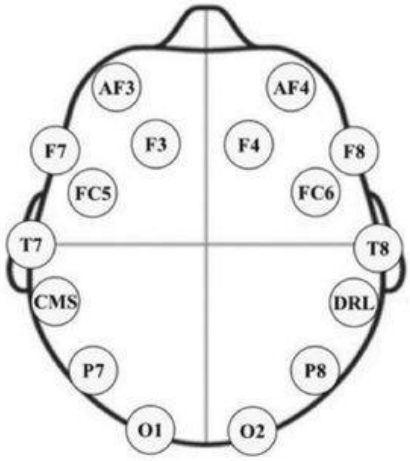

Figure 5.4 Scalp Distribution of Electrodes Used in the Study.

To ensure data quality, a strict preprocessing pipeline was applied. Artifact removal techniques were applied to remove noise caused by eye blinks and head movements. A statistical filtering was also performed by computing Z-scores and removing outliers more than 3 standard deviations away from the mean.

Each 2-minute resting-state recording had 2048 samples per channel, which were used to calculate mean and standard deviation features. The data were split into five-second windows to extract temporal features of EEG data. Overall, high-quality EEG recordings, strict preprocessing criteria, and repeated measures provided a very clean and reliable dataset to perform dyslexia classification based on QEEG biomarkers.

An 80:20 ratio was used to split the dataset into training and testing sets. This ratio allowed enough data for model training while retaining a meaningful test set for validating the inference performance. In addition to the 80:20 train-test split, 5-Fold Cross-Validation (CV) was performed to further assess the robustness and generalizability of the proposed framework. In particular, the 5-Fold CV involved splitting the dataset into 5 subsets, where each fold used 4 subsets for training and 1 subset for testing, and the final metrics were averaged across 5 folds.

5.3. HARDWARE AND SOFTWARE ENVIRONMENT

The experiments in this work are carried out on Google Colab. It operates as an online interactive platform to run Jupyter Notebook files. Users can access resources such as Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) for high-performance computation on Google Colab. It has been used for research and development of many machine learning models and for tasks in cryptography. This environment enabled efficient execution of the computationally demanding tasks required for this study.

A TPU was employed for this research, specifically the V2-8 TPU. As defined in (Jouppi et al., 2020), TPU is an Application-Specific Integrated Circuit (ASIC) developed by Google to accelerate machine learning workloads, especially deep neural networks. Optimized for vector and matrix processing, TPUs deliver high efficiency in executing training and inference workloads. The V2-8 TPU configuration, comprising eight cores, provided the necessary computational power to handle the resource-intensive nature of HE inferences,

significantly reducing computation time.

The use of the TPU was essential to perform HE computations efficiently. Homomorphic encryption operations, particularly those involved in encrypted inference, are inherently computationally expensive. Leveraging the TPU allowed the experiments to be conducted within a feasible time frame. The adoption of this experimental design allowed the study to highlight the effectiveness of the proposed framework in supporting secure and privacy-conscious data processing in cloud-based systems.

Python was used in all of our experiments due to its rich machine-learning libraries, mature support for homomorphic encryption schemes, and flexibility. It had a rich ecosystem of tools that made seamless implementation of both plaintext and homomorphic encryption-based inference easy. Readability and simplicity made prototyping and debugging fast, while widespread use in the research community ensured that Python was already compatible with the state-of-the-art privacy-preserving machine learning algorithms. Table 5.1 summarizes the libraries leveraged in the implementation.

Table 5.1 Used Python Libraries.

library	Usage
Pandas	Data analysis and manipulation
Torch	Deep learning framework for neural networks
Torch.nn	Module for defining ANN layers
Torch.optim	Optimization algorithms for training ANNs
Numpy	Numerical computations and array handling
Tenseal	Homomorphic encryption library for secure computations
Time	Measuring the execution time of functions
Sklearn.metrics	Evaluation metrics for model performance
Sklearn.preprocessing	Data preprocessing (scaling and normalization)
Sklearn.model_selection	Dataset splitting for training and testing
Matplotlib.pyplot	Visualization of results and performance metrics
Torch.utils.data	Handling dataset batching and loading in PyTorch

5.4. HOMOMORPHIC ENCRYPTION SCHEME SETTINGS

In our experiments, we employed the CKKS homomorphic encryption scheme to perform privacy-preserving computations on encrypted data. As mentioned in Section 3.2, CKKS is particularly suited for approximate arithmetic, making it ideal for machine learning tasks where minor numerical imprecision is acceptable. The scheme enables secure computation on encrypted floating-point values while maintaining efficiency in inference tasks.

To accommodate different experimental settings, we designed and implemented three distinct CKKS configurations. Each setup was tailored to balance computational efficiency, security, and numerical precision, depending on the nature of the dataset and the complexity of the inference process.

In the CKKS encryption scheme, the polynomial modulus degree and coefficient modulus bit sizes are fundamental parameters that directly impact security, computational efficiency, and precision. The polynomial modulus degree determines the ciphertext size and influences the security level of the encryption. A higher modulus degree provides greater security but also increases computational complexity and memory usage. In contrast, the coefficient modulus bit sizes define the available precision for encrypted computations. A larger coefficient modulus enables higher numerical accuracy but increases ciphertext size, requiring more computational resources for encryption, decryption, and homomorphic operations. The global scale parameter plays a crucial role in maintaining precision during encrypted computations. It acts as a scaling factor, mitigating precision loss due to the approximate nature of CKKS arithmetic. A large global scale will make the algorithm less susceptible to numerical error but requires a larger coefficient modulus to avoid overflow. As a result, proper setting of these two parameters is important for balancing precision, efficiency, and security to allow for well-performing encrypted machine learning models without too much overhead.

Table 5.2 summarizes the encryption parameters that were used for each of the proceeded experiments.

Table 5.2 CKKS Parameters for the Proceeded Experiments.

Experiment	Polynomial Modulus Degree	Coeff. Modulus Bit Sizes	Global Scale	Rationale for Differences
Activation Function Estimator standalone test	16,384	[60, 40, 40, 60]	2^{30}	A smaller modulus degree was chosen to minimize computational overhead since this experiment only required evaluating individual activation functions on encrypted inputs. The lower global scale ensures a manageable ciphertext size while maintaining adequate precision.
MNIST Classification	32,768	[60, 50, 50, 50, 50, 50, 60]	2^{50}	The higher polynomial modulus degree was necessary to support the complexity of neural network inference on encrypted MNIST data. A larger coefficient modulus allowed for a higher global scale, which improved numerical stability during deeper inference computations.
Dyslexia classification	32,768	[60, 50, 50, 50, 50, 50, 60]	2^{50}	Similar to MNIST, this setup required high precision and numerical stability for encrypted QEEG signal processing. The larger modulus and global scale reduced precision loss across inference operations.

The rationale for parameter selection was primarily based on computational trade-offs and the nature of the respective tasks. The activation function estimator test required minimal encryption overhead, favouring a lower polynomial modulus degree and a moderate global scale. Conversely, the MNIST and dyslexia classification implementations required extensive encrypted matrix operations, necessitating a higher modulus degree and increased coefficient bit sizes to prevent precision degradation over multiple encrypted operations.

5.5. CASE STUDY: PREDICTING HEART DISEASE

We performed a case study utilizing the UCI Heart Disease dataset to show the generalisability and practical applicability of our suggested ANN-based activation function estimators. This experiment investigates the performance of activation estimators in a real-world clinical dataset under plaintext inference conditions, but the main focus of this study has been on assessing them in the context of encrypted inference. By using a tabular medical dataset, we hope to demonstrate that the suggested estimator framework is not just applicable to image data (like MNIST and EEG) but can also be effectively applied in privacy-sensitive fields like healthcare, where decision support systems are increasingly using predictive modelling.

The UCI Heart Disease dataset is a widely used benchmark in medical machine learning studies. It contains 303 patient records with 13 input features that include clinical and demographic indicators such as age, sex, chest pain type, resting blood pressure, cholesterol level, and electrocardiographic results (Janosi et al., 1989). The purpose is to predict the presence of cardiac disease in a patient, which is expressed as a binary classification task. This dataset is especially useful for testing the resilience of machine learning models in healthcare because of its real-world properties, class imbalance, and small sample size.

While the UCI Heart Disease data set does not have direct identifiers (e.g., name, SSN), it contains the quasi-identifiers like age, sex, resting blood pressure,

and cholesterol levels that are arguably well-known to the privacy community as potential indirect identifiers, whose combinations could be leveraged to perform re-identification attacks through record linkage. A basic check of the uniqueness of the records on the four attributes of age, sex, cholesterol, and resting blood pressure yields that 10 out of 303 records (3.3%) are unique, indicating a tangible potential risk to the privacy of de-identified medical records in the real world. This is consistent with previous observations that (i) up to 28% of patients in hospital discharge data were re-identified through newspaper linkage (Sweeney et al., 2027) when the data was HIPAA compliant, and (ii) up to 99.98% of US residents are uniquely identifiable from knowledge of only 15 demographic fields (Rocher et al., 2019). Copula-based risk models on similar clinical data have shown that even a small number of quasi-identifiers can carry significant re-identification risk (Jiang et al., 2022).

The main objective of this work is not to study the risk of direct identifier disclosure but rather to verify the secure inference capability of our encrypted ANN model through ciphertext-only computations, even when quasi-identifiers are present in the input data. Such design decisions have been made to stress the importance of having privacy-preserving inference capabilities over sanitized quasi-identifier data rather than being entirely dependent on data de-identification or sanitization.

The UCI Heart Disease dataset was pre-processed for maximum data quality and compatibility with the neural network training approach. In order to retain all records, the missing values of feature variables were filled with the column-wise mean. The feature attributes were encoded into a consistent numerical form, and the class labels were encoded in binary to simplify the classification into a binary task by using label 1 for the presence of heart disease and 0 otherwise. To ensure uniformity of feature contribution, all features were standardised (zero-mean and unit-variance). The data was then split into a training and testing set with an 80:20 ratio to ensure uniform class distributions. Finally, the data was batched for further processing during training and evaluation.

The main ANN designed for this case study consists of a fully connected input layer, a single hidden layer with 512 neurons, and an output layer configured for binary classification. A dropout layer with a rate of 0.3 was incorporated after the activation function to improve generalization. Three activation functions—Sigmoid, ReLU, and Tanh—were tested in the hidden layer to assess their effect on classification performance. To facilitate encrypted inference, we trained an ANN-based estimator designed to approximate the chosen activation function using a simple two-layer fully connected architecture. During training, we first trained the main ANN using the cross-entropy loss function with an adaptive optimizer and a learning-rate scheduler over 300 epochs. Following the main model training, the estimator was trained using a regression task. Its goal was to learn the activation function's behaviour by mapping pre-activation inputs to their correct post-activation outputs. This entire experiment was conducted separately for each activation function. The results were then compared against a standard plaintext inference to measure accuracy, loss, and generalization.

To maintain consistency and comparability, the homomorphic encryption configuration in this experiment was kept identical to that used in the MNIST experiment, with a polynomial modulus degree of 32,768, coefficient modulus bit sizes of [60, 50, 50, 50, 50, 50, 60], and a scale of 2^{50}.

5.6. EVALUATION MEASUREMENT

5.6.1. Metrics and Procedures for Estimator Assessment

The evaluation of the activation function estimators in the first experiment was carried out using two key performance dimensions: computational efficiency and prediction accuracy under homomorphic encryption.

Evaluation began by measuring the execution time for homomorphic estimations. This was achieved by recording the timestamps immediately before and after the execution of the estimator functions. By measuring the difference

between timestamps, the elapsed time was derived and employed as a key measure of the homomorphic encryption scheme's computational efficiency.

After obtaining the timing data, the encrypted output tensors from the estimators were decrypted to get the plaintext predictions. The individual tensors were decrypted, and the plaintext values were concatenated into a single array for later analysis.

We measured prediction accuracy in terms of MSE between the output of the decrypted predictions and the true activation function. To calculate MSE, first compute the errors of predicted outputs with respect to the ground-truth and then square each error. After that, it calculates the mean of all the errors.

MSE enables us to see how close their estimation is to the actual data because if the MSE is low, the mean of the squared difference is also low. By also considering the computation time of our estimator along with the MSE loss, we get an overview of the performance of our estimator.

5.6.2. Evaluation Metrics for Classification Models

Different evaluation metrics were used to examine the performance of the proposed framework for different tasks. As the scope of this study covers different classification problems, such as multi-class classification for MNIST and binary classification for dyslexia detection, the evaluation metrics used were selected based on the nature of these tasks. The results of encrypted inference were first decrypted for the classification tasks and were compared with the ground truth of the test set.

Table 5.3 provides a summary of the evaluation metrics used, descriptions and their formulas. The evaluation metrics are described in more detail in (Powers, 2020). In Table 5.3, True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) represent classification outcomes used to calculate evaluation metrics for classification tasks.

Table 5.3 Evaluation Measurements.

Metric	Description	Formula	Applicability
Accuracy	Expresses the share of correctly identified instances relative to the total number assessed.	$\frac{(TP + TN)}{(TP + TN + FP + FN)}$	MNIST, Dyslexia
MSE (loss)	Measures the mean squared gap between predicted and actual values, commonly used to evaluate errors in approximation tasks.	$\frac{1}{n}\sum_{i=1}^{n}(y_i - y'_i)^2$	Activation Function Estimators test, MNIST, Dyslexia
F1-Score	The harmonic mean of Precision and Sensitivity, balancing false positives and false negatives.	$2 \times \frac{(Precision \times Sensitivity)}{(Precision + Sensitivity)}$	MNIST, Dyslexia
Sensitivity (Recall)	Indicates how many of the actual positive cases are correctly identified by the model, a metric widely called Recall.	$\frac{TP}{(TP + FN)}$	MNIST, Dyslexia
Specificity (True Negative Rate)	Measures the proportion of actual negative cases correctly identified by the model, highlighting its ability to avoid false positives.	$\frac{TN}{(TN + FP)}$	MNIST, Dyslexia
Precision (Positive Predictive Value)	The proportion of predicted positive cases that are actually positive, indicates the accuracy of positive classifications.	$\frac{TP}{(TP + FP)}$	MNIST, Dyslexia
Receiver Operating Characteristic	A graphical representation of the trade-off between True Positive Rate and	-	MNIST, Dyslexia

Table 5.3 (Next) Evaluation Measurements**.**

Metric	Description	Formula	Applicability
(ROC) Curve	False Positive Rate as the decision threshold is varied.		
Area Under Curve (AUC)	The area beneath the ROC curve serves as a measure of the model's skill in distinguishing positives from negatives.	-	MNIST, Dyslexia

Since encryption and decryption processes have consistent computational times across all experiments, these were excluded from computational efficiency measurements. The primary focus remains on evaluating the efficiency of encrypted inference within the homomorphic encryption framework. To ensure practical applicability, inference times were averaged over multiple runs to account for variability.

5.6.3. Evaluation Metrics for Explainable AI under Homomorphic Encryption

To verify that the explanations generated under HE remain faithful to those obtained in plaintext, SHAP values were computed in two modes—once using the standard plaintext workflow and once within the encrypted workflow.

In the plaintext setting, the unencrypted feature vector was analysed with the standard SHAP library, producing the reference importance scores:

$$s^{plain} = \left(s_1^{plain}, s_2^{plain}, \dots, s_d^{plain}\right) \qquad (5,1)$$

For the encrypted workflow, the same feature vector was encrypted using the CKKS scheme, processed by the cloud-side inference algorithm, and then decrypted on the client side to obtain:

$$s^{enc} = (s_1^{enc}, s_2^{enc}, \dots, s_d^{enc}) \quad (5,2)$$

The similarity between the plaintext and encrypted attribution vectors was quantified using two complementary measures.

The first is the MAE, which can be calculated through the equation:

$$MAE = \frac{1}{d}\sum_{i=1}^{d}\left|s_i^{enc} - s_i^{plain}\right| \quad (5,3)$$

which captures the average numerical discrepancy per feature between the two SHAP value sets.

Another measure used is Spearman's rank correlation coefficient, which captures the strength and direction of monotonic relationships in ranked data. For feature importance vectors s^{enc} and s^{plain}, Spearman's coefficient ρs is calculated as:

$$\rho s = 1 - \frac{6\sum_{i=1}^{d} d_i^2}{d(d^2-1)} \quad (5,4)$$

Where, d_i is the difference between the ranks of the $i-th$ feature in the two vectors. Values of ρs close to 1 indicate strong agreement in feature ranking, while values near 0 indicate little to no monotonic relationship.

While SHAP values offer valuable interpretability, perturbation-based queries in encrypted settings could theoretically introduce access-pattern leakage. To mitigate this risk, the proposed design was structured to ensure that only encrypted perturbation results are returned to the client, with all SHAP aggregation and visualization performed locally after decryption. This approach minimizes potential side-channel threats while preserving the fidelity of model explanations. For further details on Spearman's rank correlation, please refer to (Zar, 2025).

CHAPTER 6

6. RESULTS

6.1. PERFORMANCE OF STANDALONE ACTIVATION FUNCTION ESTIMATORS

Table 6.1 summarizes the performance metrics for the Sigmoid activation function, while Table 6.2 presents the results for the Tanh activation function. During testing, the activation function was recalculated using HE estimators in encrypted mode, and the outputs were decrypted for comparison with the true values. In the validation phase, new randomly generated points were used to assess the estimators' generalization.

Table 6.1 Performance Metrics for Sigmoid Estimator.

Estimator	Training		Validation	
type	Time	MSE	Time	MSE
ANN	261.3260s	0.2551	2.3612s	0.2511
Polynomial	027.7068s	0.4799	0.2988s	0.4516
Piecewise Linear	167.7890s	0.4995	1.8562s	0.4700

Table 6.2 Performance Metrics for Tanh Estimator.

Estimator	Training		Validation	
type	Time	MSE	Time	MSE
ANN	216.3279s	0.9994	2.0229s	0.9888
Polynomial	020.1660s	0.9990	0.2378s	0.9951
Piecewise Linear	156.4878s	0.9990	1.6055s	0.9951

6.2. HOMOMORPHIC ENCRYPTION INFERENCE ON THE MNIST DATASET

6.2.1. Results Using Sigmoid Activation Function Estimators

Table 6.3 presents the results of the homomorphic inferences using the Sigmoid activation function and its respective estimators within the homomorphic encryption framework. Figures 6.1, 6.2, and 6.3 display the ROC curves along with the AUC values for the homomorphic inferences conducted with the polynomial, piecewise linear, and ANN-based Sigmoid estimators, respectively.

Table 6.3 Homomorphic Inferences Using Different Estimators for Sigmoid Activation Function on MNIST Dataset.

Estimator	Accuracy	Sensitivity	Specificity	F1-score	Avg. Loss	AUC
Polynomial	0.8540	0.8540	0.9838	0.8538	0.8392	0.9863
Piecewise Linear	0.1000	0.1000	0.9000	0.0182	2.3130	0.4927
ANN	0.8710	0.8710	0.9857	0.8703	0.4238	0.9887

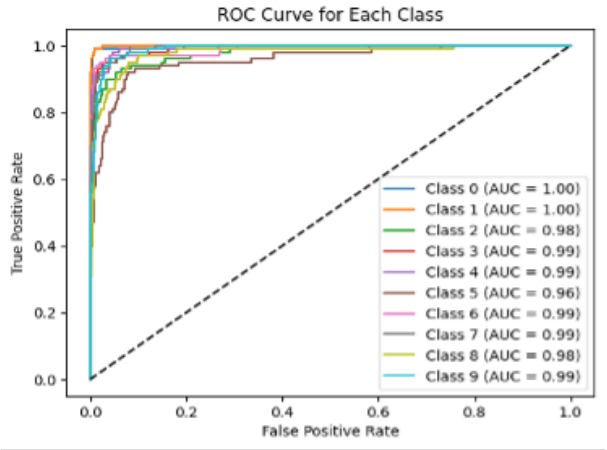

Figure 6.1 ROC Curves and AUC Values for the Homomorphic Inference Using Sigmoid Polynomial Estimator on MNIST Dataset.

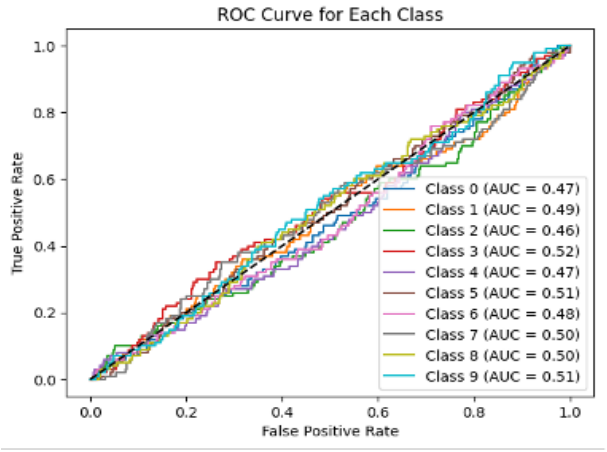

Figure 6.2 ROC Curves and AUC Values for the Homomorphic Inference Using Sigmoid Piecewise Linear Approximation on MNIST Dataset.

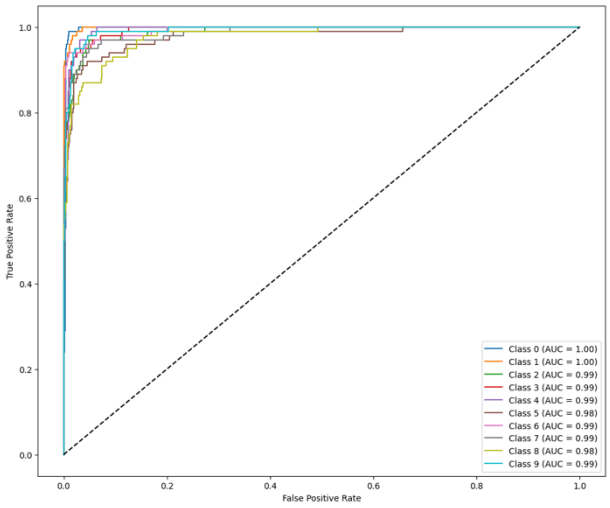

Figure 6.3 ROC Curves and AUC Values for the Homomorphic Inference Using Sigmoid ANN Estimator on MNIST Dataset.

6.2.2. Results Using Tanh Activation Function Estimators

Table 6.4 presents the homomorphic inference results using different Tanh activation function estimators. Figures 6.4, 6.5, and 6.6 display the corresponding ROC curves and AUC values for the polynomial, piecewise linear, and ANN-based estimators, respectively.

Table 6.4 Homomorphic Inferences Using Different Estimators for Tanh Activation Function on MNIST Dataset.

Estimator	Accuracy	Sensitivity	Specificity	F1-score	Avg. Loss	AUC
Polynomial	0.4950	0.4950	0.9439	0.4548	11.2686	0.9395
Piecewise Linear	0.1000	0.1000	0.9000	0.0182	2.3124	0.4938
ANN	0.8560	0.8560	0.9840	0.8549	0.4748	0.9876

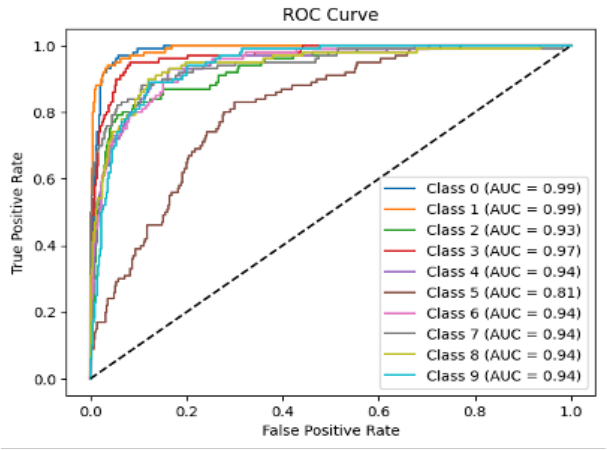

Figure 6.4 ROC Curves and AUC Values for the Homomorphic Inference Using Tanh Polynomial Estimator on MNIST Dataset.

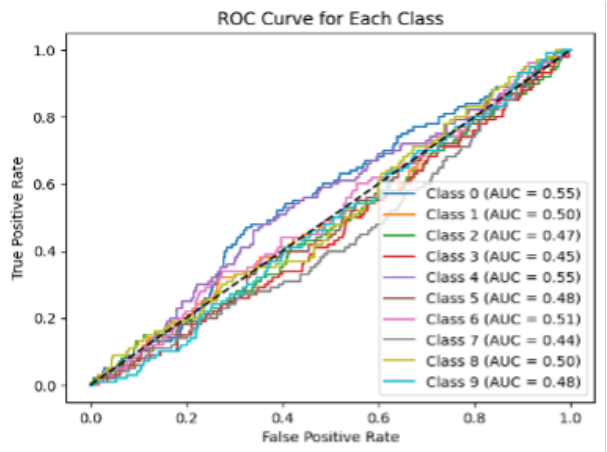

Figure 6.5 ROC Curves and AUC Values for the Homomorphic Inference Using Tanh Piecewise Linear Approximation on MNIST Dataset.

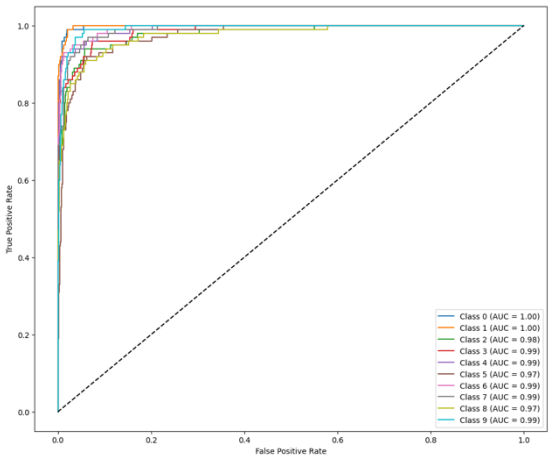

Figure 6.6 ROC Curves and AUC Values for the Homomorphic Inference Using Tanh ANN Estimator on MNIST Dataset.

6.2.3. Encrypted Inference Time

Since the homomorphically encrypted inference was tested on a dataset containing 1,000 points (100 from each class), the average computation time was calculated for each method. The encryption and decryption times were excluded, as they remain consistent across all tests. Our focus was to compare the computational differences between non-linear estimators of activation functions. Implementation time details are provided in Table 6.5.

Table 6.5 Average Implementation Time for the Encrypted Inferences Based on the Type of Estimators to Sigmoid and Tanh on MNIST Dataset.

Estimator	Sigmoid (s)	Tanh (s)
Polynomial	1.9145	1.9388
Piecewise Linear	2.8261	2.9069
ANN	5.6938	5.6947

6.2.4. Comparative Analysis

Since the majority of prior works (see Section 2.2) apply CNNs as the underlying network to perform the inference on the HE data, and in order to facilitate a more fair and comprehensive comparison, we evaluate the extension of the proposed ANN-based estimator in the CNN settings. We present a comparison on the CNN network architecture, HE security level (bits), classification accuracy, and implementation time in this subsection. With a comparison to the existing polynomial/piecewise-based methods, we show that the proposed ANN estimator can still be effective and flexible when applied to deeper and more complex networks. Meanwhile, this comparison also reveals the trade-off among the accuracy, security, and computational overhead. We summarize this comparison in Table 6.6.

The security level for CKKS and BFV HE schemes is primarily determined by two critical parameters: the polynomial modulus degree (N) and the coefficient modulus bit sizes. As outlined by Furka et al. (2019), the polynomial modulus degree N significantly influences both security and computational complexity, where larger N values generally provide higher security but increased computational demands. The total sum of bits used for the coefficient modulus (denoted as Q) is then compared against standardized security bounds provided by the Homomorphic Encryption Security Standard. In particular, the maximum allowed coefficient modulus bit-length (Q_{max}) to reach a target security level, e.g., 128 bits, is given for each polynomial degree, e.g. for $N = 8{,}192$ the maximum allowed bit-size sum for 128-bit security is 218 bits. Therefore, making sure that the total bits in coefficient modulus are less than this, will ensure the target security level in bits.

6.3. REAL-WORLD APPLICATION: HE INFERENCE AND EXPLAINABLE AI FOR DYSLEXIA DETECTION

As previously mentioned, the third experiment focuses on dyslexia

detection using the QEEG dataset. To assess the performance of the proposed ANN model, Table 6.7 presents a detailed comparison of evaluation metrics in both plaintext and HE modes. Additionally, Figure 6.7 and Figure 6.8 illustrate the ROC curves and AUC values for plaintext and HE modes, respectively, offering insights into the model's classification performance. The average inference time for HE inference is recorded at 2.0969 seconds.

Table 6.6 Summary of CNN Architectures and Performance Metrics in Homomorphic Encrypted Inference for MNIST Dataset.

Ref.	Model Design	Security Level	Performance	Implementation Time
Zhang et al. (2024)	Conv Layer: (Kernel=5×5, Stride=2, Padding=2), Channels: 1→5, polynomial activation (degree=2); Fully Connected: 980→100, polynomial activation (degree=2); Fully Connected: 100→10	128-bit (Microsoft SEAL 3.7)	Accuracy: 0.9870	6.7000 s
Shi and Zhao (2023)	Input Layer: 784 neurons; 6 Hidden Layers: 512→256→128→64→ 32→16 (ReLU activation); Output Layer: 10 neurons (SoftMax activation)	Efficient integer vector based HE (security bits not explicitly stated)	Accuracy: 0.8639 – 0.8879	Not explicitly stated (reported 58× faster than traditional CNN)
Khan and Michalas (2023)	Conv: Input 28×28, kernel 5×5, stride=1, Channels: 1→5, ReLU; Mean Pooling: 2×2, stride=1; Conv: kernel 5×5, stride=1, Channels: 5→10, ReLU; Mean Pooling: 2×2, stride=1; FC: 490→128, ReLU; FC: 128→10, SoftMax	Fan-Vercauteren Somewhat HE (security bits not explicitly stated)	Accuracy: 0.9850	153.3172 s
Nguyen et al. (2023)	Conv: Kernel=5×5, stride=2, padding=2, Channels: 1→5, ReLU; FC: 980→100, ReLU; FC: 100→10	HE-based (security bits not explicitly stated)	Accuracy: 0.9831 (HeFUNs), 0.9916 (HeFUNl)	1.3740 s (HeFUNs), 1.5010 s (HeFUNl)
Xiong et al. (2020)	LeNet-style: Conv → Pool → FC (for MNIST/CIFAR-10)	BFV scheme (based on RLWE), 80-bit	Accuracy: 0.9600 – 0.9700 (Encrypted Ensemble)	< 7 min for 50,000 images; Layer-wise: Conv ~2.9000 s, Dense ~6.8000 s
Hesamifard et al. (2018)	Fully connected neural network with 1–5 hidden layers, using low-degree	80-bit security	Accuracy (Polynomial Sigmoid): 0.9915	~320.0000 s

Table 6.6 (Next) Summary of CNN Architectures and Performance Metrics in Homomorphic Encrypted Inference for MNIST Dataset.

Ref.	Model Design	Security Level	Performance	Implementation Time
	polynomial approximations (Chebyshev or custom) to replace standard activation functions, implemented for both training and inference on encrypted data via levelled HE			
This work	Conv (3×3, 32→64) → MaxPool ×2 → FC (3136→64→10), Sigmoid ANN Estimator	CKKS (poly_mod_degree=32768, 128-bit equiv.)	Accuracy: 0.9770, F1: 0.9770, AUC: 0.9997, Spec: 0.9974	CNN Train: 162.6100 s; Estimator Train: 137.5700 s; Encrypted Inference: 21.8737 s

Table 6.7 Evaluation Metrics Results for Dyslexia Detection.

Inference Type	Accuracy	Sensitivity	Specificity	F1-score	AUC	5-Fold CV
Plaintext	0.9269	0.6991	0.6991	0.7631	0.8604	0.9246
HE	0.9003	0.6134	0.6134	0.6536	0.8218	0.8953

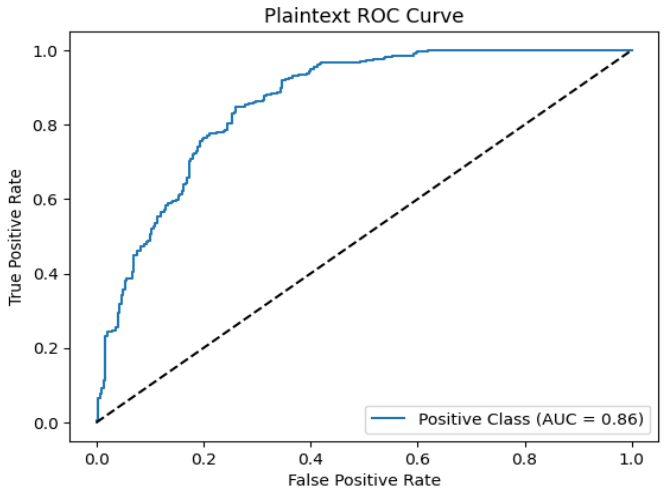

Figure 6.7 ROC Curve and AUC Value for the Plaintext Inference of Dyslexia Detection.

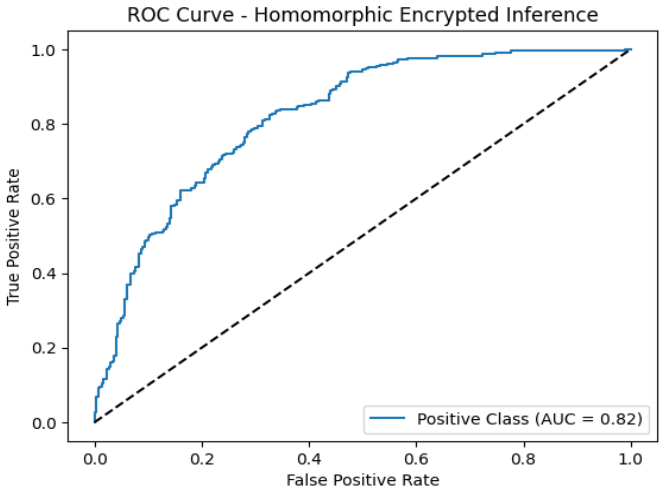

Figure 6.8 ROC Curve and AUC Value for the HE Inference of Dyslexia Detection.

A critical parameter in the SHAP algorithm under the HE mode (Algorithm 4.2 in Chapter 4) is the number of perturbations N used to sample the feature space. In a plaintext environment, increasing the number of perturbations typically converges towards a more accurate Shapley value. However, in a homomorphically encrypted environments, increasing N imposes a linear penalty on computational time and introduces potential noise accumulation.

To determine the optimal N for the HE inference mode, we conducted a sensitivity analysis comparing the fidelity of SHAP values generated under encryption against those generated in plaintext. In the conducted analysis, four distinct perturbation levels were tested, which are $N \in \{10, 25, 50, 75\}$.

To ensure the reliability of these measurements and mitigate the influence of input-specific variability, the experiment was conducted using 3 randomly selected data points from the test dataset. For each perturbation level N, SHAP values were computed for these three distinct samples in both plaintext and encrypted modes. The results reported in Figure 6.9, including both the Spearman Rank Correlation coefficients and the total computation times, represent the average values derived from these three instances. The blue line

represents the Spearman Correlation (fidelity), and the red dashed line represents the execution time.

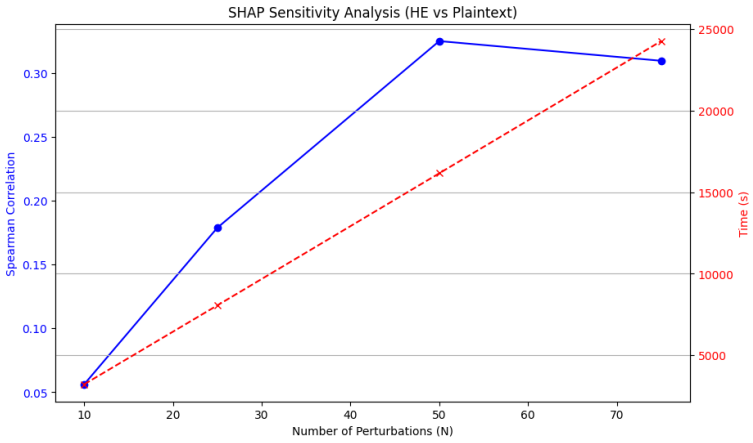

Figure 6.9 SHAP Sensitivity Analysis (HE vs Plaintext).

Based on the empirical results from Figure 6.9, and since we need to determine the perturbation number to proceed with the following steps, $N = 50$ was selected as the optimal hyperparameter for the subsequent experiments, as it provided the maximum interpretability fidelity before the onset of noise-induced degradation.

Figure 6.10 illustrates the interpretability analysis using mean absolute SHAP-like values, highlighting the relative importance of features during HE inference.

To provide a more intuitive spatial understanding of the SHAP analysis, Figure 6.11 visualizes the feature importance for each EEG channel as a heatmap, identifying critical brain regions.

For comparative purposes, Figure 6.12 shows the SHAP-like feature attributions computed in plaintext inference mode, serving as a reference to evaluate the consistency and reliability of encrypted explainability results.

Table 6.8 quantitatively compares the encrypted and plaintext SHAP-like outputs. The results show a moderate MAE and moderate positive Spearman rank correlation, indicating that encrypted inference preserves a significant portion of the interpretability structure observed in plaintext.

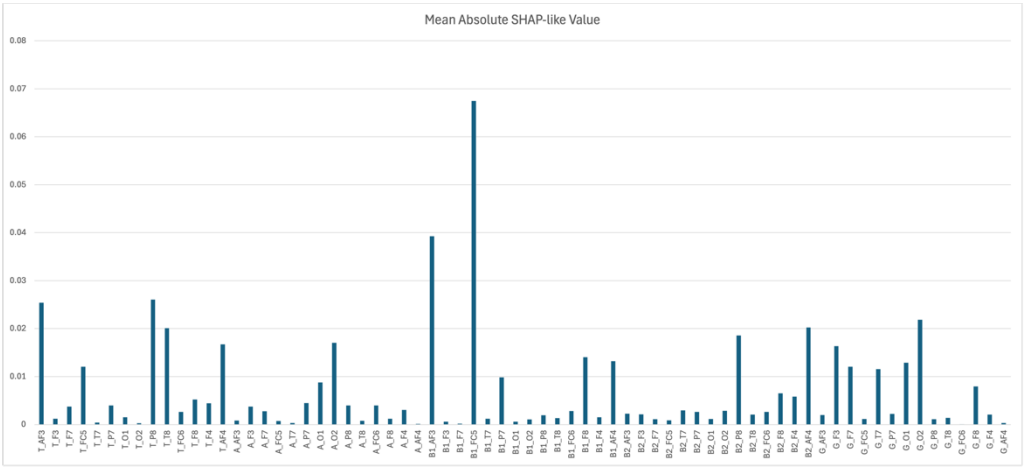

Figure 6.10 Interpretability Analysis: Mean Absolute SHAP-Like Values for Features in Encrypted Inference.

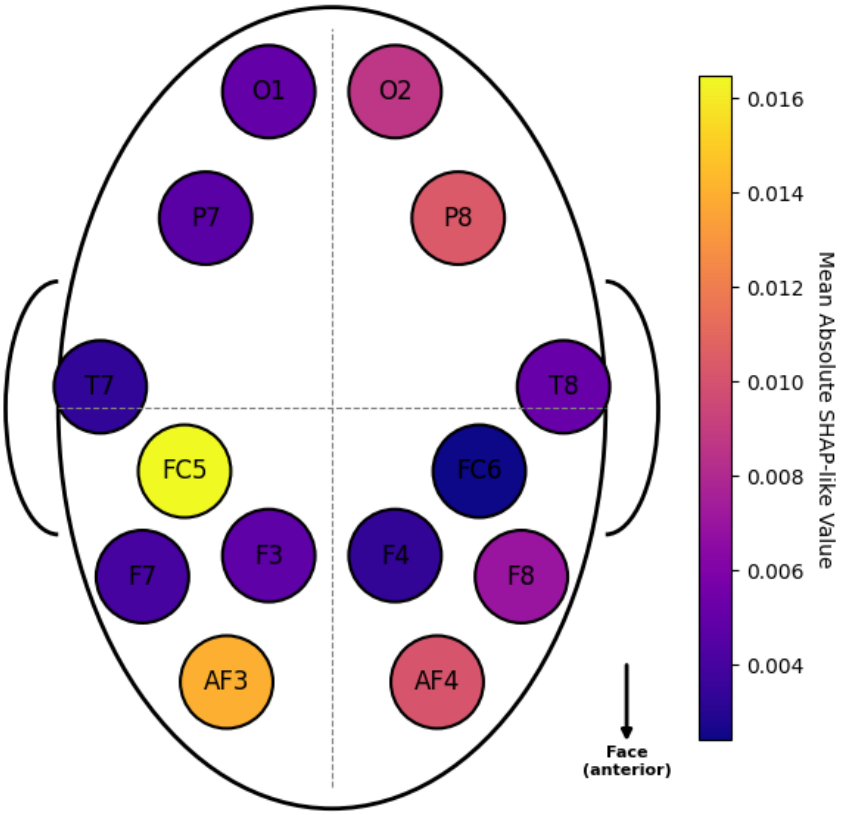

Figure 6.11 EEG channel-wise importance during HE inference.

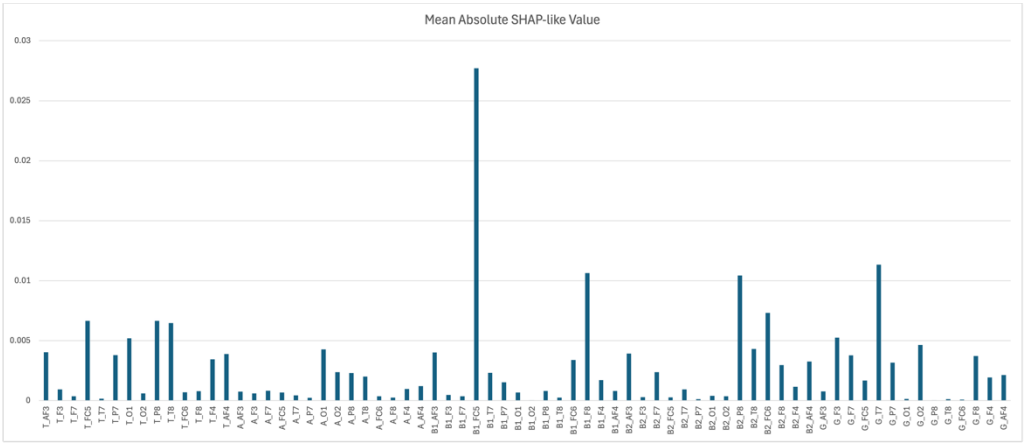

Figure 6.12 Interpretability Analysis: Mean Absolute SHAP-Like Values for Features in Plaintext Inference.

Table 6.8 Comparison between Encrypted and Plaintext SHAP-like Values.

Metric	Value	Interpretation
MAE	4.95×10^{-3}	Moderate deviation in magnitude
Spearman Rank Correlation (ρ)	0.5885	Moderate positive correlation
p-value (for ρ)	8.41×10^{-8}	Highly statistically significant ($p \ll 0.01$)
Max Absolute Plaintext SHAP	≈ 0.0114	Context for interpreting the MAE scale

6.4. CASE STUDY RESULTS: HEART DISEASE PREDICTION

The performance results of the ANN classifier and activation function estimators on the UCI Heart Disease dataset are presented in Table 6.9. The table compares three activation functions—Sigmoid, Tanh, and ReLU—under both plaintext and ciphertext inference.

Table 6.9 Performance Comparison of Activation Function Estimators on the UCI Heart Disease Dataset.

Activ-ation Funct-ion	Inference Type	Time (s)	Accuracy	Sensitivity	Specificity	F1-score	Loss	AUC
Sig.	Plaintext	0.0200	0.8525	0.8582	0.8582	0.8525	0.3272	0.9502
	Ciphertext	5.7030	0.8525	0.8582	0.8582	0.8525	0.3500	0.9459
Tanh	Plaintext	0.0200	0.8033	0.8155	0.8155	0.8019	0.4238	0.9437
	Ciphertext	5.7164	0.7705	0.7798	0.7798	0.7699	0.7048	0.9123
ReLU	Plaintext	0.0200	0.8033	0.8101	0.8101	0.8032	0.7537	0.9297
	Ciphertext	5.7208	0.8525	0.8582	0.8582	0.8525	1.1677	0.9343

CHAPTER 7

7. DISCUSSION

7.1. SECURITY THREAT MODEL AND GAME-THEORETIC ANALYSIS

This section formalizes the adversarial setting, identifies the primary attack surfaces illustrated in Figure 7.1, and explains how each threat is mitigated without degrading the diagnostic performance.

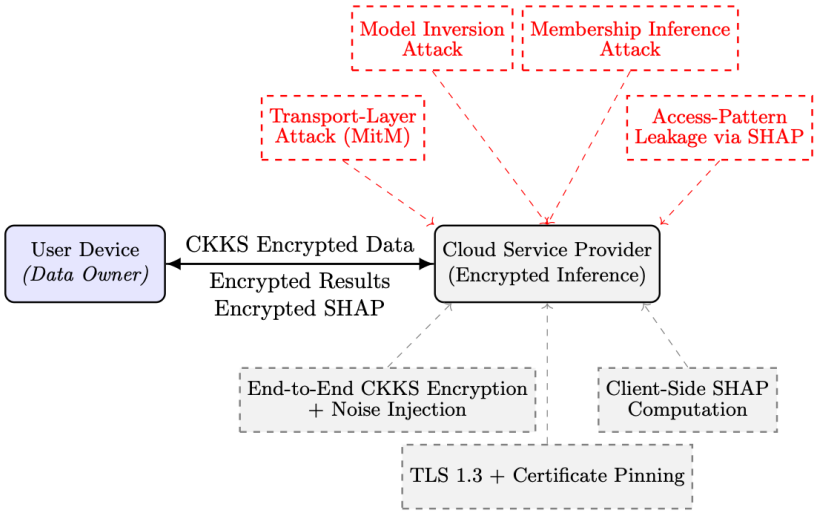

Figure 7.1 Game-Theoretic Adversarial Model: Visualization of threats, attack vectors, and corresponding mitigation strategies in privacy-preserving encrypted dyslexia detection system.

We adopt an honest-but-curious threat model, where the CSP strictly follows the prescribed protocol but may attempt to extract sensitive information from ciphertexts, intermediate computation values, or explainability outputs. Communication channels are also assumed to be susceptible to man-in-the-

middle (MitM) adversaries capable of intercepting, replaying, or modifying messages. Fully malicious CSP behavior—such as deviating from the protocol or deliberately providing incorrect inference results—is considered out of scope for this study and is left for future investigation.

The main attack vectors to be considered are the following:

1. Model inversion: Reconstructing prototypical QEEG signals, given access to the encrypted outputs of a black-box model or its gradients, i.e., an inverse attack (Fredrikson et al., 2015).
2. Membership inference: Inferring whether a targeted patient record was or was not part of the training data (Shokri et al., 2017).
3. Access-pattern leakage via SHAP: Leaking information about features or the training dataset, using perturbation queries in order to infer saliency scores (Yan et al., 2023).
4. Transport-layer MitM: Intercepting or tampering with the encrypted messages and content, including public keys, ciphertexts, and encrypted SHAP values.

7.1.1. Game-Theoretic Security Framework

We model the defender–adversary interaction as a two-player, non-cooperative strategic game:

- Player 1 (Defender): Data owner and system designer.
- Player 2 (Adversary): Honest-but-curious CSP or MitM attacker.

Defender strategies (S_D):

- End-to-end CKKS encryption for all computations.
- DP noise injection.
- Client-side SHAP computation.
- TLS 1.3 with certificate pinning.

Adversary strategies (S_A):

- Model inversion.
- Membership inference.

- Access-pattern analysis from SHAP queries.
- Transport-layer interception or tampering.

The defender's objective is to maximize data confidentiality and minimize leakage while maintaining acceptable model performance. The adversary's goal is to maximize information gain or attack success probability. The optimal defensive strategy minimizes the adversary's expected payoff and stabilizes the system at a point where the adversary cannot improve their outcome by changing strategies.

7.1.2. Mitigation Strategies

We utilize a layered security architecture to defend against the aforementioned threats. This architecture includes the following mechanisms:

1. End-to-End CKKS Encryption: All matrix multiplications and activation function estimations are carried out directly on ciphertexts. This ensures that the plaintext data, intermediate results, and SHAP values are all inaccessible to the CSP.
2. DP Noise Injection: Gaussian noise with a variance of $\sigma^2 = 0.25$ is added to the hidden-layer activations. This provides $(\varepsilon = 1.0, \delta = 10^{-5})$ differential privacy guarantees for the NN, preventing inference attacks.
3. Client-Side SHAP Computation: The perturbation results are encrypted and sent to the CSP and decrypted only on the client side. This prevents the CSP from obtaining access to feature importance scores.
4. TLS 1.3 with Certificate Pinning: This ensures the integrity and confidentiality of communication, protecting against MitM and replay attacks.

Table 7.1 provides a mapping of threats and mitigations in the system.

The combination of encryption, DP, secure explainability, and robust communication protocols diminishes the system's susceptibility to model inversion, membership inference, access-pattern leakage, and MitM attacks.

These security guarantees are based on the Ring-LWE hardness assumption for CKKS and the formal privacy bounds of DP. Any remaining risk, such as cross-session linkage, will be a topic for future work.

Table 7.1 Threats and Corresponding Mitigation Strategies.

Threat	Attack Strategy	Defence Mechanism
Model Inversion Attack	Reconstruct QEEG features from encrypted outputs/logits	Apply End-to-End CKKS Homomorphic Encryption; Inject DP Noise ($\varepsilon = 1.0$, $\delta = 10^{-5}$)
Membership Inference Attack	Determine if a record is part of the training set	Apply End-to-End CKKS Homomorphic Encryption; Inject DP Noise
Access-Pattern Leakage via SHAP	Analyse perturbations to infer feature importance	Perform Client-Side SHAP Computation; Encrypt Perturbation Responses
Man-in-the-Middle (MitM) Attack	Intercept or tamper with encrypted communications	Establish TLS 1.3 with Certificate Pinning

7.2 ANALYSIS OF STANDALONE ACTIVATION FUNCTION ESTIMATORS

For the Sigmoid function (Table 6.1), the ANN estimator clearly has the lowest MSE during training and validation. This makes sense, since the ANN estimator is allowed to best approximate the function in question, which is Sigmoid in this case. However, as previously discussed, this increased accuracy has a drawback in terms of increased inference times compared to both the Polynomial and Piecewise Linear estimators. The Polynomial estimator demonstrated quicker inference capabilities but incurred a higher MSE, which indicates it may not efficiently model the function. The Piecewise Linear estimator also had a higher MSE compared to the ANN estimator, similar to the Polynomial approach.

When we examine the Tanh function in Table 6.2, we see a similar trend. The ANN model was the best, with respect to MSE; however, this came at the expense of training time. The Polynomial estimator was the quickest; however,

this had the greatest MSE. The Piecewise Linear estimator achieves a satisfactory balance between the ANN and Polynomial estimator regarding training time and MSE performance. The MSE values obtained from all Tanh function estimators display minimal variances, with some even matching.

These results further highlight the trade-offs that must be made when selecting an estimator for activation functions. It is evident that ANNs provide the most accurate results. However, in certain situations, the resources available to run these estimators may be limited. For instance, ANN models may be impractical if the time required to compute the query or private function must be extremely short. Simpler estimators, such as the Polynomial or Piecewise Linear estimators will be faster, but less accurate. It is for this reason that the estimator should be made in accordance with the requirements of the application, e.g., what is considered a "real-time" response, or the error that may be tolerated.

7.3. EVALUATING HOMOMORPHIC ENCRYPTION INFERENCE ON THE MNIST DATASET

In this subsection, the ANN estimators for the Sigmoid and Tanh activation functions are assessed in the MNIST classification problem. By re-visiting the evaluation metrics enlisted in Tables 6.3 and 6.4, it is noticeable that ANN-based estimators work favourably towards the approximation of non-linear activation functions within PPMLs with an HE overhead. By cross-verifying the performance of the Sigmoid and Tanh activation function estimators across the polynomial, piecewise linear, and ANN-based approximations, it is clear that ANN estimators present the best-fit performance with respect to accuracy and sensitivity metrics. For example, the average AUC for ANN estimators is 0.9887 for Sigmoid (Figure 6.3) and 0.9876 for Tanh (Figure 6.6), while polynomial and piecewise linear approximation yield a sub-optimal performance. These observations could intuitively conclude that ANN-based approximation is a more precise mode of representing and working with complex non-linear activation functions for carrying out encrypted inference for

machine learning tasks.

One major benefit of ANN-based estimators is that they are not tied to any particular approximation method for modelling the non-linear activations. Various methods exist to approximate non-linear functions, but each comes with specific advantages and disadvantages, making the flexibility of ANN-based estimators beneficial because they can use any activation function without limitations to specific non-linear patterns. The Universal Approximation Theorem demonstrates that ANN estimators achieve generalization for different nonlinear functions by adjusting the training data without changing the model structure. By comparison, the polynomial approximation method would need a higher degree polynomial to achieve a similar level of accuracy, resulting in a larger number of multiplications. The piecewise linear methods, on the other hand, cannot exactly fit the curve of a smooth activation function, which results in a less accurate estimator. In summary, ANN estimators provide a well-balanced approach that can achieve both accuracy and computational efficiency. This generality is also especially important in the context of MLaaS, where different activation functions are needed in different encrypted models.

Examining the piecewise linear approximation method for estimating Sigmoid and Tanh in Tables 6.3 and 6.4, it is apparent that this approach resulted in significantly lower accuracy compared to other estimation techniques. This reduction in performance can be attributed to the inherent limitations of piecewise linear approximations in capturing the non-linear characteristics of complex activation functions. Since this method partitions the input space into discrete intervals and applies linear transformations within each segment, it fails to account for the smooth and continuous curvature of activation functions such as Sigmoid and Tanh. As a result, essential non-linear information is lost, impacting the model's predictive accuracy, particularly when dealing with datasets characterized by intricate non-linear relationships between features.

ANN-based estimators have better performance in accuracy compared to the previous estimators, but at the cost of a higher computation burden. The average inference time for the ANN estimator, which was encrypted, is 5.69

seconds for both Sigmoid and Tanh (refer to Table 6.5) compared to 1.91 seconds for the polynomial estimator. It is important to note that the computational overhead introduced by ANN-based estimators, while a significant factor, might be a worthwhile trade-off for the considerable performance gain achieved. This is especially true in applications where accuracy is paramount and computational resources are not a limiting constraint, such as in critical domains like healthcare and finance. In these fields, the need for precise and privacy-preserving inferences can make the additional computational time a justifiable expense for the sake of data security and model reliability.

If we see the comparative analysis in Table 6.6, it is evident that while some previous works, such as Zhang et al. (2024), achieve a high accuracy of 0.9870 using two-stage polynomial activations, and Khan and Michalas (2023) secure 0.9850 accuracy with an intricate CNN framework that incurs over 150 seconds of inference time, these approaches come with increased complexity and computational overhead. Similarly, Nguyen et al. (2023) and Xiong et al. (2020) deliver competitive performance with fast inference times yet often depend on complex network designs and advanced noise-management techniques—and some even operate at a lower 80-bit security level. In contrast, our approach leverages a lightweight, single-layer ANN-based estimator that not only achieves a commendable accuracy of 0.9770 but also registers an outstanding AUC of 0.9997. This high AUC value is particularly significant as it demonstrates the model's superior ability to distinguish between classes under encrypted conditions, which is critical in high-stakes applications where precision is essential. Furthermore, our solution offers 128-bit strong security (equivalent to most high-security settings in the literature), while reaching a fully encrypted inference time of around 21.87 seconds. This trade-off design further demonstrates the practical advantages of our method: reduced complexity, strong security, and state-of-the-art discriminative performance, making it an appealing choice for real-world PPML applications.

7.4. INSIGHTS FROM HE INFERENCE AND EXPLAINABLE AI IN DYSLEXIA DETECTION USING QEEG DATA

7.4.1. Dyslexia Classification Discussion

As can be observed from the results, the proposed HE inference solution is able to reach significant scores, which demonstrates its feasibility even when computing in an encrypted space. Although there is a minor decrease in performance due to the transition to encrypted computation, the results still show the strength and practicality of HE in terms of secure and private computations.

As can be observed from the ROC curves in Figures 6.7 and 6.8, both the plaintext and HE inferences are still able to provide significant class separation. In particular, the AUC for the HE inferences reached 0.8218, which is only marginally lower than the corresponding plaintext inference AUC of 0.8604. The close values in AUC demonstrate that the HE-based model is still able to provide similar modelling of the relationship between the variables and is still able to maintain significant classification power in the fully encrypted domain, which is a sign of robustness of the entire pipeline to noise and approximation error.

On reviewing the results in Table 6.7, we can find that the accuracy of HE inference is 0.9003, which is close to that of plaintext accuracy of 0.9269. The values of sensitivity, specificity, and F1-score also follow a similar trend. We can safely conclude that the HE inferences model has been able to achieve a balance between the true positive and true negative rates. The performance outcome is remarkable given the restrictions both in computational power and representation when working with encrypted data. Moreover, the results of 5-Fold Cross Validation give further proof of the robustness and generalizability of the proposed framework. We see that the average value of accuracy in the plaintext mode is 0.9246 and the average value in HE inference mode is 0.8953. We see that the 80: 20 split and 5-Fold CV average values follow similar trends, proving that the performance of the model is consistent with data partitions.

The decrease in sensitivity (0.6134 in HE vs. 0.6991 in plaintext) and

specificity is expected and not considered critical due to the noise and precision restrictions inherent to the encryption. The gap between the two methods was anticipated, as the limits on precision and the consequent noise accumulation effects of HE is well known to cause mild sensitivity degradation. This trade-off is generally accepted within the field; for instance, the High-Level Expert Group on AI of the European Commission underlines privacy and data governance as fundamental pillars of trustworthy AI systems and states that slight model performance compromises may be morally acceptable if weighed against more robust data protection measures (European Commission's High-Level Expert Group on AI, 2019). Similarly, biomedical ethics principles emphasize the critical importance of patient confidentiality and autonomy (Beauchamp & Childress, 2001; Munjal & Bhatia, 2023). This additional context also justifies the HE inference model's trustworthiness in real-world use cases where the privacy of the user's data needs to be prioritized.

The results demonstrate that the proposed HE inference solution for dyslexia detection achieves strong performance metrics, reaffirming its practicality for privacy-preserving computations. While a slight decline in accuracy compared to plaintext inference is observed, the overall robustness of the HE-based approach highlights its viability for secure machine learning applications.

7.4.2. Analysis of SHAP Perturbation Sensitivity in Encrypted Inference

The implementation of SHAP within an FHE scheme such as CKKS introduces unique challenges that are not present in standard plaintext deployments. Our sensitivity analysis (presented in Figure 6.9) highlighted a non-monotonic relationship between the number of perturbations N and the accuracy of the explanation.

As expected, lower perturbation counts (such as $N = 10$ and $N = 25$) resulted in significantly faster implementation times. However, the correlation between the encrypted and plaintext SHAP values was poor in these configurations. This is consistent with the theory of Shapley values, where

insufficient sampling of the feature coalitions leads to high variance and unstable attribution scores. In a privacy-preserving context, this instability renders the explanation untrustworthy, as it fails to accurately reflect the model's decision boundary.

Conversely, while increasing N typically improves accuracy in plaintext models, our results demonstrated a performance degradation at $N = 75$. Although the execution time increased linearly as expected—reflecting the computational burden of performing additional encrypted forward passes—the fidelity of the explanations dropped compared to $N = 50$. This phenomenon can be attributed to the cumulative noise inherent in the CKKS scheme.

As discussed in Section 3.2.1, every arithmetic operation introduces a small amount of noise to the ciphertext in CKKS. While the scheme allows for rescaling to manage this noise, the aggregation of a large number of encrypted perturbation results (as required by the SHAP kernel) accumulates this noise. At $N = 75$, the noise accumulation began to outweigh the benefits of additional sampling, distorting the final marginal contributions and reducing the correlation with the ground truth.

Therefore, $N = 50$ represents the optimal balance for this architecture. It reconciles the statistical requirement for sufficient sampling to achieve a meaningful correlation against the cryptographic constraints of noise budget and the practical constraints of implementation time. This finding, robustly supported by the multi-sample average, proofs that in HE-based XAI, simply increasing computational resources does not guarantee better results; rather, careful parameter tuning is required to manage the delicate trade-off between sampling precision and encryption noise.

7.4.3. SHAP Analysis and Neurophysiological Interpretability

The SHAP values from Figure 6.10 and visualized more clearly in the heatmap in Figure 6.11, can also give insight into which features and, by proxy, brain regions are most important for the SHAP prediction of dyslexia. The

QEEG features that have the highest SHAP values are largely overlap with the most affected regions in neurophysiological studies of dyslexia and other types of learning disorders. Dyslexia is a neurodevelopmental disorder, and as such, has a consistent pattern of brain activity disruption associated with it.

The temporal and frontal regions of the brain remain the most important with SHAP-like importance scores. Features like T_AF3, T_P8, and T_T8 have the highest SHAP values, and as such, may be the most predictive features for the model to make predictions on dyslexia. These features represent the temporal regions that have been shown to be significantly involved in phonological decoding, reading fluency, and even broader auditory processing. Frontal features T_F3 and T_F7 have high SHAP values as well, and this is largely due to their proximity to the temporal lobe but also to dyslexia's effects on working memory, attentional control, and executive function processing (functions known to be disrupted in dyslexics). The emphasis on these temporal and frontal regions also relates to their disruption on the left side of the brain in language-related tasks.

Occipital and posterior regions, represented by features like T_O1 and T_O2, contribute to the model's predictions with moderate SHAP values. These regions are associated with visual processing and visual word form recognition, challenges that are common in dyslexic individuals. Additionally, lateralization patterns remain evident, with left-hemisphere features (e.g. T_AF3, T_F3) generally showing higher SHAP values compared to their right-hemisphere counterparts, consistent with the left hemisphere's dominance in language processing.

Interestingly, some features related to the B1 and B2 channels were among the highest-ranking according to the SHAP score, namely B1_FC5, B1_AF3, and B2_AF4, which is indicative of the importance of frontal-temporal connectivity as well as parietal cortex in dyslexia prediction. The frontal-temporal regions have been previously associated with phonological processing and mapping of orthographic to phonological representations, two cognitive processes known to be affected in dyslexia. Two other features with the high

contributions were G_O2 and G_F3, which also point to the involvement of cortical regions across the brain.

On the other hand, other features like G_FC6 or B2_FC6 had almost null SHAP values, and therefore, the influence on the model prediction is considered to be low. Dyslexia-related brain activity appears confined to specific cortical areas instead of being diffusely spread throughout the entire cortex.

From a neurophysiological standpoint, the greater importance of frontal areas is consistent with their role in executive control and attention regulation, while the predominance of the temporal lobe highlights its essential role in auditory and phonological processing. The involvement of parietal and occipital lobes further underscores the integration of visual and auditory information that is crucial for reading and language comprehension. These results validate the biological plausibility of the high-importance features identified by the model.

The visualization of the SHAP values further supports these observations, clearly demonstrating the dominance of temporal, frontal, and selected parietal regions in the model. This visualization aligns closely with the neural mechanisms underlying dyslexia, particularly disruptions in phonological and orthographic processing networks, and provides a clear representation of how different brain regions contribute to the model's predictive performance.

We analysed the behaviour of our explainability under encryption as follows. We examined the SHAP-like feature attributions computed in the encrypted domain (Figure 6.10) and compared them with the plaintext ones (Figure 6.12) as shown in Table 6.8. The MAE between plaintext and encrypted feature importance scores is 4.95×10^{-3}, which is a small absolute error in the context of the overall range of values. In addition, the Spearman rank correlation coefficient was $\rho = 0.5885$ with a p-value of 8.41×10^{-8}. The p-value shows that the two feature attribution sets are moderately correlated (positive monotonic) and statistically significantly different from uncorrelated (uncorrelated implies the value of the rank correlation is 0). Thus, our encrypted SHAP-like explanation successfully captured semantically interpretable patterns of feature importance that are consistent with the plaintext model. The deviation

is expected since the explanation is an approximation due to the activation estimator and the approximate nature of the explanation under HE in a perturbation-based manner. This experiment provided evidence to support the use of interpretable explanations in the setting of privacy-preserving ML and a reasonable consistency with the baseline case (standard SHAP).

7.4.4. Implications for Research and Clinical Applications

One noteworthy aspect of these findings is the neurophysiological consistency they demonstrate. The identification of key QEEG features, such as sensor readings from specific electrodes, reflects underlying brain regions and neural activity patterns. This alignment between XAI insights and established dyslexia-related research suggests that the model is capturing biologically relevant information. The relevance of these features to dyslexia aligns with existing literature on the disorder, providing external validation for the model's predictions. In the context of dyslexia detection, these insights offer a potential bridge between machine learning models and our understanding of the neurobiological basis of dyslexia. By highlighting the specific brain regions and features associated with dyslexia, the model can provide valuable information for further research, early detection, and intervention strategies. Understanding the key brain regions and patterns associated with dyslexia can aid in the development of targeted interventions, such as neurofeedback training or personalized reading programs, aimed at strengthening the affected neural pathways.

In terms of advancing the model's performance, these insights can be used to inform feature engineering and model refinement. For example, additional features related to functional connectivity between high-importance regions can be incorporated into the model to capture more complex patterns of brain activity associated with dyslexia. Further research can also explore the impact of different feature selection methods on classification performance and investigate potential subgroup-specific differences in feature importance.

7.5. CASE STUDY ANALYSIS: UCI HEART DISEASE DATASET

Related to the case study results, the findings in Table 6.9 confirm that HE inferences performs closely to plaintext inference across all tested activation functions. Although HE inferences introduces a noticeable increase in implementation time due to encryption overhead, the model's classification performance remains consistent. Accuracy, sensitivity, specificity, and AUC values show minimal degradation, indicating that the proposed ANN estimator can effectively approximate activation functions in the encrypted domain.

Among the activation functions, ReLU and Sigmoid achieved the highest accuracy during HE inference, each reaching 0.8525, matching their plaintext counterparts. This demonstrates the estimator's ability to accurately replicate activation function behavior in encrypted inference settings. While Tanh exhibited slightly lower performance under encryption, it still produced reasonable results, affirming the robustness of the overall approach for medical data analysis in secure environments.

The performance of our ANN models on the UCI Heart Disease dataset aligns with previous studies using the same data. For example, Mohan et al. (2019) achieved 0.8901 accuracy with a hybrid ANN model, Srinivasan et al. (2023) reported 0.9478 using a Naïve Bayes network, and Narasimhan & Victor (2025) observed ANN accuracy of about 0.8361. These results support the validity of our approach, demonstrating that our models maintain competitive accuracy even under HE constraints.

7.6. LIMITATIONS AND FUTURE DIRECTIONS

Despite the promising outcomes demonstrated in this study, several limitations must be acknowledged.

For the MNIST dataset experiments, a key limitation lies in the performance of the piecewise linear approximation. While computationally efficient, this approach exhibited significantly lower accuracy and sensitivity,

making it less suitable for HE applications that demand high precision. Additionally, noise accumulation within HE computations remains a challenge, particularly in ANN-based estimators where multiple encrypted operations compound the noise, potentially affecting inference accuracy. This issue underscores the need for further optimization of both ANN architectures and HE parameters to balance computational feasibility with precision, especially when scaling to larger datasets or high-dimensional inputs.

Similarly, another point to take into consideration for the dyslexia detection experiment is the accuracy loss due to encrypted computation. As already mentioned previously, the calculations in HE are prone to noise, which might influence the final model in terms of its accuracy and generalisability in comparison to plaintext inference. This is seen in the marginal decrease in sensitivity and specificity when HE-based prediction was used. Moreover, QEEG data can be subject to high variability in data acquisition, as differences in collection protocols, electrode placement, and preprocessing techniques between studies might introduce inconsistencies that impact generalisability. Standardization is important for reproducibility and clinical applicability.

Another notable limitation is the computational overhead imposed by HE inferences. The recorded average inference time of 2.0969 seconds in dyslexia detection is significantly higher than that of plaintext computation, which may hinder real-time or large-scale implementations. While such a trade-off is acceptable in privacy-sensitive applications, future optimizations in encryption parameters and model structure could help reduce computational latency without compromising accuracy.

Moreover, the current study does not investigate subgroup-specific variations, such as the influence of age, gender, or dyslexia severity on predictive performance. Exploring these variations could improve the model's robustness and offer insights into how different demographic factors impact classification outcomes.

CONCLUSION AND FUTURE PERSPECTIVES

This dissertation explored the integration of HE with ANNs to enable PPML. The research addressed fundamental challenges in performing secure encrypted inference while maintaining high accuracy, computational efficiency, and model interpretability. The proposed framework introduced ANN-based estimators for approximating non-linear activation functions, particularly Sigmoid and Tanh, overcoming the limitations of traditional polynomial and piecewise linear approaches. Additionally, explainability techniques were incorporated to ensure transparency in HE-based inference.

The experimental evaluation across two case studies—MNIST digit classification and dyslexia detection using QEEG data—demonstrated the efficacy of the proposed approach. In the MNIST experiment, ANN-based estimators significantly improved accuracy, sensitivity, and AUC compared to polynomial and piecewise linear approximations, reinforcing their potential for enhancing encrypted inference. The dyslexia detection experiment further validated the practicality of HE-based models in real-world applications, achieving an AUC of 0.8218 while preserving data privacy. SHAP-based analysis also confirmed the biological relevance of the model's feature importance rankings, supporting the interpretability of encrypted inference outcomes.

Despite these advancements, computational cost remains a challenge. While ANN-based estimators achieve superior accuracy, their increased processing time may limit their applicability in latency-sensitive environments. Similarly, the overhead introduced by HE operations—averaging 2.0969 seconds per prediction in the dyslexia detection study—highlights the need for further optimization to support real-time inference scenarios.

For the future works, ANN-based HE estimators would require improvement to reduce their computational overhead while preserving their estimation accuracy. This can be achieved by optimizing their design and

exploring more efficient neural network architectures, model pruning techniques, and quantization methods. Extending the framework to support additional activation functions, such as SoftMax, would broaden its applicability to various deep learning models.

Addressing the high computational demands associated with HE-based inference would be an important future direction. Optimizing the encryption scheme, fine-tuning parameter selection, and leveraging hardware accelerations such as GPUs, TPUs, or FPGA-based implementations can help reduce the inference time. Exploring hybrid privacy-preserving techniques that combine HE with other approaches, such as SMPC or TEEs, could offer a more balanced trade-off between security and efficiency.

Improving the interpretability of encrypted machine learning models is another area for future exploration. While this work demonstrated explainability using SHAP, other feature attribution methods could be explored and benchmarked to enhance interpretability further. Developing explainability techniques tailored for deep learning architectures and encrypted settings would be essential for increasing transparency and trust in privacy-preserving AI systems, particularly in regulated industries.

Expanding the proposed approach to multi-modal machine learning and personalized models is an important direction. Integrating additional data sources beyond QEEG features, such as behavioural and eye-tracking data, could improve the dyslexia detection accuracy and robustness of the model.

Incorporating subgroup-specific variations, such as age, gender, or severity levels, would allow for personalized adaptations and increase the model's generalizability to different populations. Investigating the application of FL as a privacy-preserving mechanism for training models on distributed datasets without sharing raw data can also be considered in future work.

Finally, future research should address real-world deployment and scalability considerations. Assessing the practicality of integrating HE-based ML models into existing cloud-based MLaaS offerings and compliance with data protection regulations, such as GDPR and HIPAA, is essential. Developing

optimized batch processing approaches to handle larger datasets efficiently in encrypted domains and conducting longitudinal studies to evaluate the long-term stability and robustness of privacy-preserving models in real-world settings will be important considerations in future work.

REFERENCES

Abou Harb, M. R., & Celiktas, B. (2024, December). Efficient Estimation of Sigmoid and Tanh Activation Functions for Homomorphically Encrypted Data Using Artificial Neural Networks. In 2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS) (pp. 1-7). IEEE.

Abou Harb, M. R., & Celiktas, B. (2025). Privacy-Preserving Machine Learning: ANN Activation Function Estimators for Homomorphic Encrypted Inference. IEEE Access.

Abou Harb, M. R., Celiktas, B. & Eroğlu, G. (2025, October). Secure and Interpretable Dyslexia Detection Using Homomorphic Encryption and SHAP-Based Explanations. In TIPTEKNO'25 (pp. 1-4). IEEE.

Abou Harb, M. R., Celiktas, B. & Eroğlu, G. (2025). Privacy-Preserving Dyslexia Detection Using Homomorphic Encryption for QEEG Data with Explainable AI Insights via SHAP [Manuscript submitted for publication].

Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (Csur), 51(4), 1-35.

Activeloop. (n.d.). MNIST dataset documentation. Activeloop. Retrieved 1/2/2025 from https://datasets.activeloop.ai/docs/ml/datasets/mnist/

Allavarpu, V. D., Naresh, V. S., & Mohan, A. K. (2025). Neural network-driven privacy-preserving credit risk analysis: A homomorphic encryption approach. Contemporary Mathematics, 6, 1051–1075.

Almutairi, N., Coenen, F., & Dures, K. (2023). PPNNBP: A Third-Party Privacy-Preserving Neural Network With Back-Propagation Learning. IEEE Access, 11, 31657-31675.

Alpaydin, E. (2021). Introduction to machine learning (4th ed.). MIT Press.

Al-Rubaie, M., & Chang, J. M. (2019). Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy, 17(2), 49–58.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, 58, 82-115.

Amin, A., Hasan, K., Zein-Sabatto, S., Chimba, D., Ahmed, I., & Islam, T. (2023, December). An explainable ai framework for artificial intelligence of medical things. In 2023 IEEE Globecom Workshops (GC Wkshps) (pp. 2097-2102). IEEE.

Aremu, T., & Nandakumar, K. (2023, February). Polykervnets: Activation-free neural networks for efficient private inference. In 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML) (pp. 593-604). IEEE.

Badcock, N. A., Mousikou, P., Mahajan, Y., De Lissa, P., Thie, J., & McArthur, G. (2013). Validation of the Emotiv EPOC® EEG gaming system for measuring research quality auditory ERPs. PeerJ, 1, e38.

Baryalai, M., Jang-Jaccard, J., & Liu, D. (2016, December). Towards privacy-preserving classification in neural networks. In 2016 14th annual conference on privacy, security and trust (PST) (pp. 392-399). IEEE.

Beauchamp, T. L., & Childress, J. F. (2001). Principles of biomedical ethics (5th ed.). Oxford University Press.

Cao, X. K., Wang, C. D., Lai, J. H., Huang, Q., & Chen, C. P. (2023). Multiparty secure broad learning system for privacy preserving. IEEE Transactions on Cybernetics, 53(10), 6636–6648.

Chen, H., Gilad-Bachrach, R., Han, K., Huang, Z., Jalali, A., Laine, K., & Lauter, K. (2018). Logistic regression over encrypted data from fully homomorphic encryption. BMC medical genomics, 11(Suppl 4), 81.

Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23 (pp. 409-437). Springer International Publishing.

Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., ... & Ranjan, R. (2023). Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys, 55(9), 1-33.

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science.

Epelbaum, T. (2017). Deep learning: Technical introduction. arXiv preprint arXiv:1709.01412.

Eroğlu, G., Gürkan, M., Teber, S., Ertürk, K., Kırmızı, M., Ekici, B., ... & Çetin, M. (2022). Changes in EEG complexity with neurofeedback and multi-sensory

learning in children with dyslexia: A multiscale entropy analysis. Applied Neuropsychology: Child, 11(2), 133-144.

European Commission's High-Level Expert Group on AI. (2019). Ethics guidelines for trustworthy AI. Publications Office of the European Union. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Frasca, M., La Torre, D., Pravettoni, G., & Cutica, I. (2024). Explainable and interpretable artificial intelligence in medicine: a systematic bibliometric review. Discover Artificial Intelligence, 4(1), 15.

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 1322–1333).

Furka, M., Kalúz, M., Fikar, M., & Klaučo, M. (2023). Guidelines for secure process control: Harnessing the power of homomorphic encryption and state feedback control. IEEE Access, 11, 110328–110341.

European Commission. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation). Available at: https://eur-lex.europa.eu (Accessed: 16 November 2024).

Gallego-Molina, N. J., Ortiz, A., Arco, J. E., Martinez-Murcia, F. J., & Woo, W. L. (2024). Unraveling Brain Synchronisation Dynamics by Explainable Neural Networks using EEG Signals: Application to Dyslexia Diagnosis. Interdisciplinary Sciences: Computational Life Sciences, 1-14.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) (pp. 249–256). PMLR.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. https://www.deeplearningbook.org/

Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine, 38(3), 50-57.

Guo, Z., Wang, S., Jin, W., Gong, C., & Lin, N. (2019, October). A k-nearest neighbor algorithm based on homomorphic encryption. In 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC), Data Science and Computational Intelligence (DSCI), and Smart Computing, Networking and Services (SmartCNS) (pp. 15–20). IEEE.

Haykin, S. (2009). Neural Networks and Learning Machines (3rd ed.). Pearson.

Hesamifard, E., Takabi, H., Ghasemi, M., & Jones, C. (2017, November). Privacy-preserving machine learning in cloud. In Proceedings of the 2017 on cloud computing security workshop (pp. 39-43).

Hesamifard, E., Takabi, H., Ghasemi, M., & Wright, R. N. (2018). Privacy-preserving machine learning as a service. Proceedings on Privacy Enhancing Technologies.

HHS. (2003). Summary of the HIPAA Privacy Rule. Available at: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html (Accessed: 16 November 2024).

HHS. (2013). HIPAA Security Rule and Cloud Computing. Available at: https://www.hhs.gov/hipaa/for-professionals/special-topics/health-information-technology/cloud-computing/index.html (Accessed: 16 November 2024).

Hong, S., Park, J. H., Cho, W., Choe, H., & Cheon, J. H. (2022). Secure tumor classification by shallow neural network using homomorphic encryption. BMC genomics, 23(1), 284.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

ICO. (2018). Guide to the General Data Protection Regulation (GDPR). Available at: https://ico.org.uk (Accessed: 16 November 2024).

Izabachène, M., Sirdey, R., & Zuber, M. (2019, January). Practical fully homomorphic encryption for fully masked neural networks. In Proceedings of the International Conference on Cryptology and Network Security (pp. 24–36).

Jaber, S. (2024). Machine Learning Algorithms for Intelligent Medical Devices: Addressing Data Security Challenges in Cloud-Based Healthcare.

Janosi, A., Steinbrunn, W., Pfisterer, M., & Detrano, R. (1989). Heart disease [dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C52P4X

Jiang, Y., Mosquera, L., Jiang, B., Kong, L., & El Emam, K. (2022, June). Measuring re-identification risk using a synthetic estimator to enable data sharing. PLoS ONE, 17(6), e0269097.

Jouppi, N. P., Yoon, D. H., Kurian, G., Li, S., Patil, N., Laudon, J., ... & Patterson, D. (2020). A domain-specific supercomputer for training deep neural networks. Communications of the ACM, 63(7), 67-78.

Khan, T., & Michalas, A. (2023, November). Learning in the Dark: Privacy-Preserving Machine Learning using Function Approximation. In 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (pp. 62-71). IEEE.

Kurek, W., Pawlicki, M., Pawlicka, A., Kozik, R., & Choraś, M. (2023, July). Explainable artificial intelligence 101: Techniques, applications and challenges. In International Conference on Intelligent Computing (pp. 310-318). Singapore: Springer Nature Singapore.

KVKK. (2016). Law on the Protection of Personal Data (Law No. 6698). Available at: https://kvkk.gov.tr (Accessed: 16 November 2024).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lee, J.-H. (2024). Efficient polynomial approximations for non-arithmetic functions in inference on fully homomorphically encrypted data (Doctoral dissertation). Seoul National University, Seoul, South Korea.

Lee, J. W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., ... & No, J. S. (2022). Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. iEEE Access, 10, 30039-30054.

Lei, D., Hu, C., & Dong, J. (2023, June). NPNNL: A Non-interactive Privacy-preserving Neural Network Learning Scheme. In 2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom) (pp. 129-133). IEEE.

Lin, Y., Zhang, T., Mao, Y., & Zhong, S. (2024). CrossNet: A low-latency MLaaS framework for privacy-preserving neural network inference on resource-limited devices. IEEE Transactions on Dependable and Secure Computing.

Mahmood, F. (2025). Controlling Noise Budget of Fully Homomorphic Encryption in Secure Machine Learning (Master's thesis, Hamad Bin Khalifa University (Qatar)).

Marcel, A., Miu, D., & Rancea, A. (2023, October). Privacy Preserving Neural Network Models based Homomorphic Encryption: A Case Study for Diabetes Prediction. In 2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 269-275). IEEE.

Mehta, U., Vekariya, J., Mehta, M., Kaur, H., & Kumar, Y. (2025). A review of privacy-preserving machine learning algorithms and systems. Applied Data Science and Smart Systems, pp. 220–225.

Mercier, D., Lucieri, A., Munir, M., Dengel, A., & Ahmed, S. (2021). Evaluating privacy-preserving machine learning in critical infrastructures: A case study on time-series classification. IEEE Transactions on Industrial Informatics, 18(11), 7834-7842.

Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. IEEE Access, 7, 81542–81554.

Mordor Intelligence (2024) Machine Learning As A Service (MLaaS) Market Report | Industry Analysis, Size & Forecast. Available at: https://www.mordorintelligence.com/industry-reports/global-machine-learning-as-a-service-mlaas-market (Accessed: 16 November 2024).

Munjal, K., & Bhatia, R. (2023). A systematic review of homomorphic encryption and its contributions in healthcare industry. Complex & Intelligent Systems, 9(4), 3759-3786.

Narasimhan, G., & Victor, A. (2025, March). A hybrid approach with metaheuristic optimization and random forest in improving heart disease prediction. Scientific Reports, 15(1), 10971.

Natarajan, D., Loveless, A., Dai, W., & Dreslinski, R. (2023, July). CHEX-MIX: Combining Homomorphic Encryption with Trusted Execution Environments for Oblivious Inference in the Cloud. In 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P) (pp. 73-91). IEEE.

National Institute of Standards and Technology (NIST) (2011) The NIST Definition of Cloud Computing (SP 800-145). Gaithersburg, MD: U.S. Department of Commerce. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (Accessed: [insert access date]).

Nguyen, D. T. K., Duong, D. H., Susilo, W., Chow, Y. W., & Ta, T. A. (2023). HeFUN: Homomorphic Encryption for Unconstrained Secure Neural Network Inference. Future Internet, 15(12), 407.

Park, J. & Lim, H. (2022) Privacy-preserving federated learning using homomorphic encryption. Applied Sciences, Vol. 12, No. 2, Article 734.

Powers, D. M. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061. https://doi.org/10.48550/arXiv.2010.16061

Raja, V. (2024). Exploring challenges and solutions in cloud computing: A review of data security and privacy concerns. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 4(1), 121-144.

Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. Foundations of Secure Computation.

Robaa, M., Balat, M., Awaad, R., Omar, E., & Aly, S. A. (2024). Explainable AI in handwriting detection for dyslexia using transfer learning. arXiv preprint, arXiv:2410.19821.

Rocher, L., Hendrickx, J. M., & de Montjoye, Y.-A. (2019, July). Estimating the success of re-identifications in incomplete datasets using generative models. Nature Communications, 10(1), 3069.

Sarkar, E., Chielle, E., Gürsoy, G., Mazonka, O., Gerstein, M., & Maniatakos, M. (2021). Fast and scalable private genotype imputation using machine learning and partially homomorphic encryption. IEEE access, 9, 93097-93110.

Scheibner, J., Ienca, M., & Vayena, E. (2022). Health data privacy through homomorphic encryption and distributed ledger computing: an ethical-legal qualitative expert assessment study. BMC Medical Ethics, 23(1), 121.

Scott, M., & Su-In, L. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30, 4765-4774.

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM Digital Library.

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 3–18). IEEE.

Srinivasan, S., Gunasekaran, S., Mathivanan, S. K., Jayagopal, P., & Dalu, G. T. (2023, August). An active learning machine technique-based prediction of cardiovascular heart disease from UCI-repository database. Scientific Reports, 13(1), 13588.

Sivan, R., & Zukarnain, Z. A. (2021). Security and privacy in cloud-based e-health system. Symmetry, 13(5), 742.

Solove, D. J., & Schwartz, P. M. (2018). Information Privacy Law (6th ed.). Wolters Kluwer Law & Business.

Song, C., & Shi, X. (2024). ReActHE: A homomorphic encryption friendly deep neural network for privacy-preserving biomedical prediction. Smart Health, 32, 100469.

Su, G., Wang, J., Xu, X., Wang, Y., & Wang, C. (2024). The utilization of homomorphic encryption technology grounded on artificial intelligence for privacy preservation. International Journal of Computer Science and Information Technology, 2(1), 52–58.

Sweeney, L., Yoo, J. S., Perovich, L., Boronow, K. E., Brown, P., & Brody, J. G. (2017, January). Re-identification risks in HIPAA safe harbor data: A study

of data from one environmental health study. Technology Science, 2017, Article 2017082801.

Ter-Minassian, L., Ghalebikesabi, S., Diaz-Ordaz, K., & Holmes, C. (2024). Explainable AI for survival analysis: a median-SHAP approach. arXiv preprint arXiv:2402.00072.

T'Jonck, K., Kancharla, C. R., Pang, B., Hallez, H., & Boydens, J. (2022, September). Privacy preserving classification via machine learning model inference on homomorphic encrypted medical data. In 2022 XXXI International Scientific Conference Electronics (ET) (pp. 1-6). IEEE.

Usman, O. L., & Muniyandi, R. C. (2020). Cryptodl: Predicting dyslexia biomarkers from encrypted neuroimaging dataset using energy-efficient residue number system and deep convolutional neural network. Symmetry, 12(5), 836.

Usman, O. L., Muniyandi, R. C., Omar, K., & Mohamad, M. (2022, February). Privacy-Preserving Classification Method for Neural-Biomarkers using Homomorphic Residue Number System CNN: HoRNS-CNN. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-8). IEEE.

Usman, O. L., Muniyandi, R. C., Omar, K., Mohamad, M., Owoade, A. A., & Kareem, M. A. (2025). HoRNS-CNN model: An energy-efficient fully homomorphic residue number system convolutional neural network model for privacy-preserving classification of dyslexia neural-biomarkers. Brain Informatics, 12(1), 11.

Vizitiu, A., Niţă, C. I., Puiu, A., Suciu, C., & Itu, L. M. (2020). Applying deep neural networks over homomorphic encrypted medical data. Computational and Mathematical Methods in Medicine, 2020, 1–26.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the General Data Protection Regulation. International Data Privacy Law, 7(2), 76-99.

Wood, A., Altman, M., Bembenek, A., Bun, M., Gaboardi, M., Honaker, J., ... & Vadhan, S. (2018) Differential privacy: A primer for a non-technical audience. Vanderbilt Journal of Entertainment & Technology Law, Vol. 21, pp. 209.

Xiong, A., Nguyen, M., So, A., & Chen, T. (2020, November). Privacy preserving inference with convolutional neural network ensemble. In 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC) (pp. 1-6). IEEE.

Xiong, J., Chen, J., Lin, J., Jiao, D., & Liu, H. (2024). Enhancing privacy-preserving machine learning with self-learnable activation functions in fully homomorphic encryption. Journal of Information Security and Applications, 86, 103887.

Xu, R., Baracaldo, N., & Joshi, J. (2021). Privacy-preserving machine learning: Methods, challenges and directions. arXiv preprint, arXiv:2108.04417.

Yan, A., Huang, T., Ke, L., Liu, X., Chen, Q., & Dong, C. (2023). Explanation leaks: Explanation-guided model extraction attacks. Information Sciences, 632, 269–284.

Yao, Y., Zhao, Z., Chang, X., Mišić, J., Mišić, V. B., & Wang, J. (2021, June). A novel privacy-preserving neural network computing approach for E-Health information system. In ICC 2021-IEEE International Conference on Communications (pp. 1-6). IEEE.

Zar, J. H. (2005). Spearman rank correlation. Encyclopedia of biostatistics, 7.

Zhu, F., Hu, F., Zhao, Y., Chen, B. & Tan, X. (2024) A secure and fair federated learning framework based on consensus incentive mechanism. Mathematics, Vol. 12, No. 19, Article 3068.

APPENDICES

APPENDIX A: PRACTICAL DEPLOYMENT ARCHITECTURE FOR ENCRYPTED INFERENCE

OVERVIEW

To validate the real-world feasibility of the proposed framework, a practical, proof-of-concept implementation of the client-server encrypted inference workflow was deployed. This architecture utilizes the CKKS homomorphic encryption scheme (via the TenSEAL library) to demonstrate a secure, scalable deployment on a serverless cloud platform.

The implementation is composed of two primary actors:

● Client: The data owner (e.g., patient or clinic). The client owns the CKKS secret key, encrypts the input data (e.g., QEEG features), and is the only party that can decrypt the encrypted result returned by the server. All communication occurs through secure HTTPS API calls.

● Server (CSP): This component follows the "honest-but-curious" threat model. It holds only the public CKKS context (without the secret key) and the trained model parameters. The server executes the entire ANN forward pass—including the ANN-based activation function estimators—directly on the ciphertexts and returns an encrypted result.

This deployment provides a concrete validation that the server never accesses the client's secret key and never processes or observes plaintext data, ensuring end-to-end privacy.

SYSTEM ARCHITECTURE AND COMPONENTS

The server-side architecture was deployed with managed cloud services. In general, the following functional groupings were established:

- Serverless Compute: A fully managed, auto-scaling compute service (e.g. Google Cloud Run) was utilized to host the stateless HTTPS inference API. This service automatically scales down to zero when there are no requests, to limit operational costs. The service is capable of scaling up to support multiple simultaneous requests.
- Model Artifact Storage: A secure, managed object storage service (e.g. Google Cloud Storage) was used to store the trained model parameters (e.g. weights.h5,.weights.npz files, etc). These models (sigmoid.npz, tanh.npz, relu.npz) were pre-trained offline (e.g. in Google Colab) and exported as .npz weight files for upload into the storage service.
- Container and Build Services: The server application was packaged as a Docker container and stored in a container registry (e.g. Artifact Registry). A managed cloud build service (e.g. Cloud Build) was utilized to automatically build the container and push it to the registry.
- Security and Access Management: A dedicated runtime service account was created with limited privileges (e.g. using Google IAM). This service account was only granted the ability to read from the model artifact storage bucket and only read from the container registry.

SERVICE CONFIGURATION AND API PROTOCOL

The service was configured with a resource profile sufficient for HE operations (e.g., 2 vCPUs, 4 GiB RAM). The API protocol is designed to be stateless, requiring the client to manage all cryptographic keys.

API Endpoint Specification

The server exposes three primary endpoints for client interaction:

Method	Path	Purpose
GET	/	Service liveness and route discovery.
POST	/register_context	Client uploads the public CKKS context (without the secret key). This context includes re-linearization keys required for encrypted multiplication.
POST	/infer/{model}	Client submits a Base64-encoded ciphertext for inference. The server returns a Base64-encoded ciphertext of the prediction.

ENCRYPTED INFERENCE WORKFLOW

1. **Client-Side:** The client initializes its CKKS context, including the secret key.
2. **Client-Side:** The client serializes its context without the secret key and POSTs it to the /register_context endpoint.
3. **Client-Side:** The client encrypts its input vector (e.g., 70 QEEG features) into a CKKSVector and Base64-encodes it.
4. **Client-Side:** The client POSTs the encoded ciphertext to the /infer/{model} endpoint.
5. **Server-Side:** The server loads the specified model weights (e.g., sigmoid.npz) from storage. It performs the HE forward pass using the received ciphertext and the public context.
6. **Server-Side:** The server returns the resulting encrypted CKKSVector (the prediction) as a Base64-encoded string.
7. **Client-Side:** The client decodes the response and decrypts the CKKSVector locally using its secret key to retrieve the plaintext prediction.

SECURITY AND ERROR HANDLING

The architecture's design directly mitigates several security risks and provides clear error feedback.

Security and Privacy Guarantees:

- The secret key never leaves the client device.
- The server only receives the public context and encrypted data.
- At no point does the server process, store, or have access to plaintext patient data.

The API provides explicit error codes to the client, primarily for cryptographic parameter mismatches.

HTTP Code	**Error Message**	**Cause & Mitigation**
400	"Inference failed: end of modulus switching chain reached"	**Cause:** The CKKS parameters (modulus chain) are insufficient for the model's multiplicative depth. **Mitigation:** The client must regenerate its context with a larger polynomial modulus degree or a different coefficient modulus chain.
400	"invalid payload / decode"	**Cause:** Malformed Base64 string or a version mismatch between the client's and server's TenSEAL libraries. **Mitigation:** The client must ensure correct encoding and library alignment.
413	"Request too large"	**Cause:** The HTTP request (containing the context or ciphertext) exceeds the server's limit (e.g., 32 MiB). **Mitigation:** The client must use smaller CKKS parameters (e.g., a smaller poly_modulus_degree) or reduce the batch size of packed inputs.

CURRICULUM VITAE