UNIVERSITY OF SURREY

UNIVERSITY OF SURREY

Faculty of Engineering and Physical Sciences

MSc Dissertation

The Relation between Neural Networks and Homomorphic Encryption

A project supervised by Professor Liqun Chen

Jamie C Dance (PG/T – Computer Science)

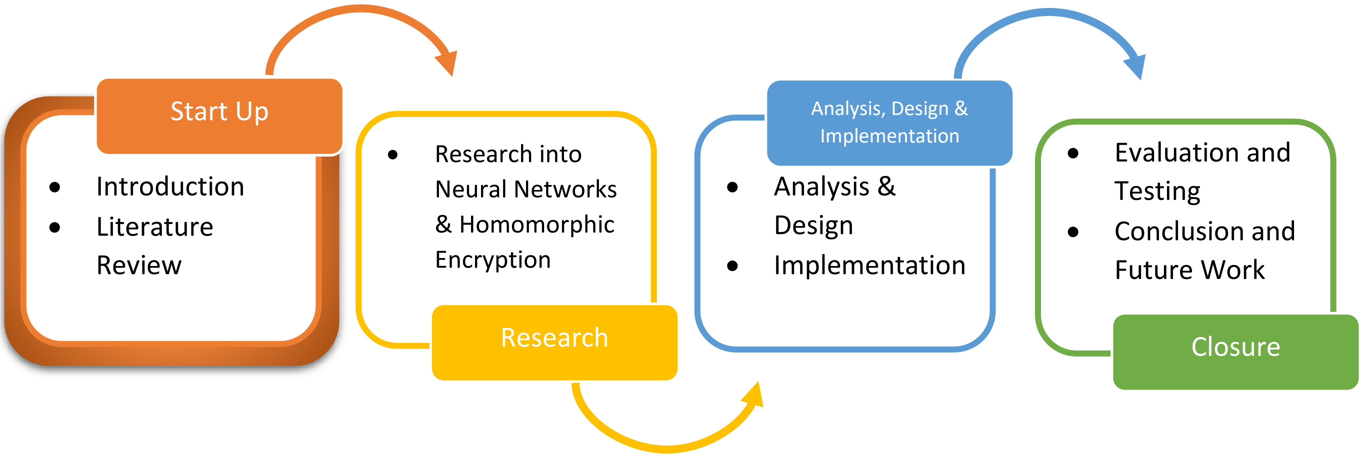
URN: 6661320

Abstract

Acknowledgements

Table of Contents

Part 1: START UP



Start Up

- Introduction
- Literature Review

- Research into Neural Networks & Homomorphic Encryption

Research

Analysis, Design & Implementation

- Analysis & Design
- Implementation

- Evaluation and Testing
- Conclusion and Future Work

Closure

Chapter 1:

1.1 Introduction

We live in the age of algorithms, a term not so long ago that was surrounded by puzzlement. With the rise of Machine Learning they are in every nook and cranny of civilization. They're not just a part of your cell phone, laptop or bank credit score, they are in your car, your house and even some of your fridges! Included within all that data can be very personal and sensitive data such as medical or financial data. So, in practice we have Data Scientists who are telling people to share their data to improve already existing mechanisms, whilst privacy experts are advising people to hide it or even delete it. This has led to the privacy dilemma: either sensitive user data must be revealed to the entity that evaluates the cognitive model (e.g, in the cloud), or the model itself must be revealed to the user so that the evaluation can take place locally REFERENCE - https://eprint.iacr.org/2017/1114.pdf. Applying machine learning to a problem that involves this data not only requires accurate predictions but also careful attention to maintaining data privacy and security.

This project will be focused on applying learned Neural Networks to encrypted data. This allows a data owner to send their data in an encrypted form to a cloud service that hosts the networks. The encryptions ensure the data remains confidential since the cloud does not have the means to decrypt it. Regardless, the cloud service will be capable of applying the neural networks to the encrypted data to make encrypted predictions that are then returned to the owner who can then decrypt them. That way the service provider does not gain any information about either the raw data or the predicted output.

The growing interest in Machine Learning as a Service (MLaaS), where a marketplace of predictors is available on a pay-per-use basis, requires attention to the security and privacy of this model. Not all data types are sensitive but in certain areas such as medical, financial and marketing MLaaS is becoming popular due to its versatility.

Neural networks are often built from pre-existing data. They are usually trained to solve a classification problem (definition of classification problem). In Neural Networks, non-linearities come from activation functions which are usually picked from a small set of non-linear functions of reference…

1.2 Aims and Motivation

As already mentioned in the previous section of this Chapter, this project aims to explore and potentially improve upon current existing methods of utilising neural networks upon encrypted data. The motivation for the project came from my desire to explore new modern methods of applying machine learning rather than researching older techniques I have previously conducted on past projects. Initially I wanted to investigate into a modern Machine Learning technique used with Big Data, but my Supervisor Professor Liqun Chen specialised in Cryptography and recommend looking into some link between Machine Learning and Cryptography. After some early research I discovered a new area of work involving using Neural Networks on Encrypted data by large organisations such as Microsoft and Amazon. Prof. Liqun Chen suggested I undertook this project and explore potential improvements and future work.

The **first aim** of this project is to research and explore some already existing methods of applying Neural Networks to Encrypted data that are available online such as Cryptonets, SageMaker and Discretized Neural Networks. This will help us build an understanding as to how developers are tackling this problem currently. In addition, it with provide us with some background that might help us to improve our own model in the future.

The **second aim** is to improve upon some these current concepts and try applying them to a new set of data of my choice. In our case the goal is to discuss the accuracy of these methods and identify the strengths and weaknesses associated with them. It is important to note that the methods explored may have different means of evaluation which could cause comparisons to be difficult.

The **third aim** is to…

1.3 Project Objectives

In the previous section we have discussed the motivations and the aims for the project. In this section we will list the objectives for this project:

1. Explore and understand some of the various elements within in the project such as Neural Networks, Encrypted data or more specifically homomorphic encrypted data.
2. Review Literature that is relevant to applying Neural Networks to encrypted data that already exist online.
3. Design, implement and test a new method of applying Neural Networks to encrypted data.
4. Evaluate the quality of the new implemented method.
5. Recommend improvements that can be made for the future.

1.4 Project Stakeholders?

The new methods implemented for the sake of this Project along with all its deliverables:

- Is the work of, and owned by Jamie Dance, student number 6661321. Jamie Dance is the Author of the Report and the developer of the new Applications of Neural Networks on Homomorphic Encrypted Data. Despite referencing throughout the project to a "We" this does not mean in any way that the Author had assistance or direct input by anyone other than himself or his supervisor.
- Is based on the work at…
- Is supervised by Professor Liqun Chen.

1.5 Project Scope and Context

In this section we will provide a better explanation of the overall scope of the project and we will discuss the importance of working with encrypted data.

Lots of work to do here.

1.6 Resources and Resource Constraints

TO BE COMPLETED AT THE END…

1.7 Project Control and Risk Assessment

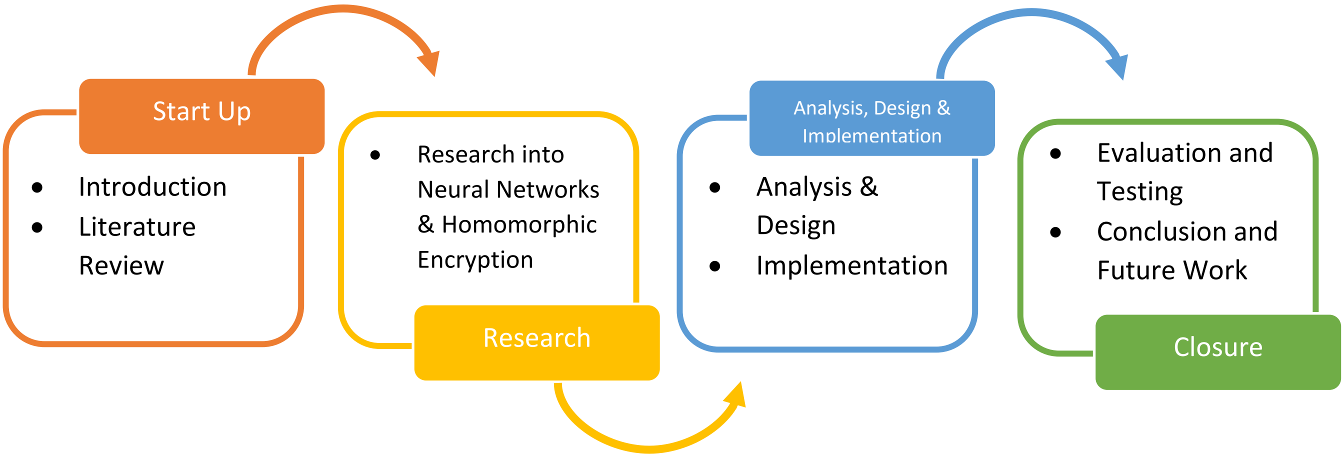
In order to keep control over the project several measures were kept in place to ensure that the project was completed smoothly and successfully in time. Many separate documents that will not be included in this report were created and used to assist in the completion of the project such as a project plan, project schedule and project notes. In addition, when necessary these notes were shared with Professor Liqun Chen for approval.

Various threats and risks have been taken in consideration in order to ensure that the project is completed successfully. These threats are identified and outlined below:

- Very new topic, 2020/2021 so there isn't a lot of content.
-

1.8 Report Structure

Inspiration for the structure of the report has been taken from one of the sample dissertations provided. The report was written by Marios Erodotou on the Automated detection and tracking of Mycobacterium Tuberculosis cells REFERENCE?. The structure of the report allows for a clear and pleasant way for the reader to navigate whilst simultaneously allowing the author to be aware of the current progress against the project schedule. Therefore, I have decided to also split the report into **four** different sections which each represented with a different theme. Within each section are a number of relevant chapters which have a number of subsections within that chapter. For example, the current section is located within "Chapter One: Introduction" which is within "Part One: Start Up". I have decided to use a colour theme for each part throughout the report to designate what content belongs to which section. As a result, any chapters or chapter subsections within a part will follow the same theme. For example, Part One follows an **Orange** theme and all following subsections also do. For each section there will be a brief introduction to the chapters that will follow, and also a summary at the end to provide a brief conclusion that sums up everything important that was discussed within the section. There will not be a summary for Part Four as the section itself is a conclusion of the project as a whole. For easier referencing I will also include an extra part at the end of each section that includes a list of corresponding references instead of one confusing block at the end of the report. A graphic will be included at the beginning of each section and will highlight the section and content that is to follow. Within each section of the graphic there is a summary of the significant chapters which allows to enhance the overall navigation and experience for the reader. The below graphic clearly illustrates the various sections and the lifecycle of this project:


Start Up

- Introduction
- Literature Review

Research

- Research into Neural Networks & Homomorphic Encryption

Analysis, Design & Implementation

- Analysis & Design
- Implementation

Closure

- Evaluation and Testing
- Conclusion and Future Work

The report contains a total of four sections and each section contains a number of chapters. Please find a brief description of each section below:

Part One: Start Up, the current part, makes an introduction to the overall project. It explains why this some basic concepts including how important the topic is. It also gives some background, including some related work, some of the motivations for the project and the project scope. Furthermore, the resources risks and objective of the project are outlined, and the structure of the report is explained. Part One closes with Chapter Two, which reviews some literature relevant to the project.

Part Two: Research, in this part we will explore various methods already developed and published to help us understand some of the possible approaches to the problem. This part will be used as a reference consistently in the following sections.

Part Three: Analysis, Design & Implementation, in this section we will describe the Analysis, Design and Implementation of our developed system. In this section we will also include all ideas considered in the design process and will incorporate a description of some of the development used for the system. It will conclude by describing the implementation stage of the project and describe the final system.

Part Four: Closure, this part concludes the report by evaluating the overall report by testing the final system. This part will also outline what went well during the project and what could be have been improved. We will discuss what could have been done differently during the project in regard to avoiding some of the pit falls that were experienced. Lastly, recommendations for how the actual system could be improved and what potential work could be done to further evolve this topic of research.

1.9 Measures of Project Success

This project will be a success if everything defined in the specification (Chapter 5) is achieved. In case the project has achieved part but not all the specification then it will be considered as incomplete but not failed. It is important to note that this field of study is still very new and is mostly still in research stage for most organisations. Therefore, the author is not expected to create a perfect system but rather attempt to improve already existing system.

Chapter 2: Literature Review

2.1 The Problem Background

In this section, some insight will be given into some of the concepts included in this report in hopes of presenting a more detailed representation of what stage this current work is at and how it has developed over the years.

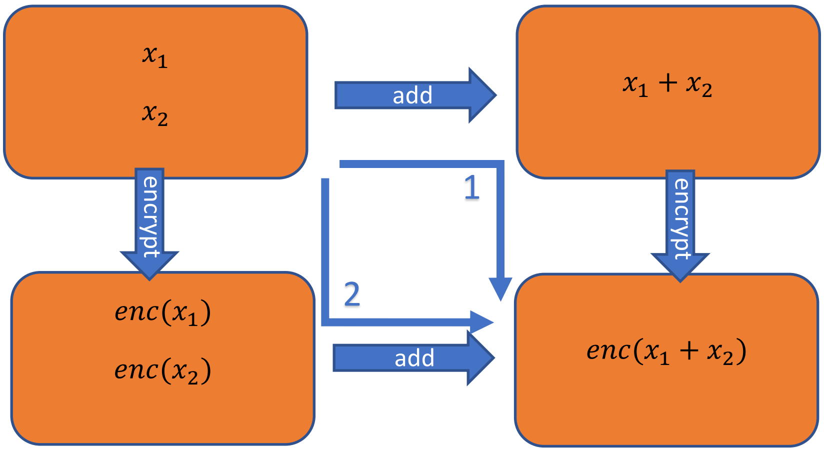
As mentioned in the Introduction, Machine Learning is a quickly developing and very powerful tool for people across the globe. But with growing success comes increasing issues and one of them issues is the topic of privacy and cybersecurity. Data scientists across the globe are encouraging people to share their sensitive material to create important developments into fields such as medicine although people aren't prepared to share information that could be detrimental for themselves. Typically for one to analyse a set of data it is important to know the input, the output and the scope/aim of the data. That cannot be the case with encrypted data as it is not possible for the analyst to see the content of the data at all. By using homomorphic encryption, we can permit users to perform computations on encrypted data without decrypting it. Processing large amounts of encrypted data can be very difficult as the processing power needed to apply machine learning techniques is significantly higher.

2.2 Recent Advances

The combination of Neural Networks and more generally Machine learning with encrypted data is a modern-day concept.

In traditional cloud storage and computation solutions the cloud needs to have unencrypted access to the customers data to compute on it, necessarily exposing the data to the cloud operators. Customers need to trust the service provider to store and manage their data appropriately, e.g., not share it with third parties without the customers' consent. As a result, data privacy relies on access control policies (such as an access control list) implemented by the cloud and trusted by the customer. With the advances in homomorphic encryption technology, it is possible to allow computations to be performed directly on encrypted data. In recent years, Fully Homomorphic Encryption development demonstrates remarkable progress. However, current literature in the homomorphic neural networks is almost exclusively addressed by practitioners looking for suitable implementations. It still lacks comprehensive and more thorough reviews. We will focus on the privacy-preserving homomorphic encryption cryptosystems targeted at neural networks identifying current solutions, open issues, challenges, opportunities, and potential research directions

The diagram below shows an example of a basic example of a traditional method of encryption (1) that requires the user to apply some function (addition in this example) to two numbers x_1 and x_2 then encrypt them. This is something that can be done with any encryption, but with Homomorphic Encryption you can go the other way around (2), you can first encrypt then apply some function. This can be done without knowing the value of x_1 or x_2 and without knowing the result. In this example we have used addition but the same can be done with multiplication and a multitude of functions.



2.3 Related Work

Homomorphic encryption schemes that are not

LOTS MORE WORK TO BE DONE

2.3.1 MLaaS with Homomorphic Encryption

MLaaS refers to services in cloud computing for deploying machine learning tools [1]. It has emerged as a flexible and scalable solution for training and predicting remotely. However, its most serious limitation is security and privacy concerns [2]. For example, prediction and classification models can involve extremely sensitive data: medical, advertising and financial, among others. Homomorphic Encryption offers a graceful solution to the problem of security in the cloud. It allows a blind process of encrypted data in a remote server, i.e., the third-party does not learn anything about the input data and output.

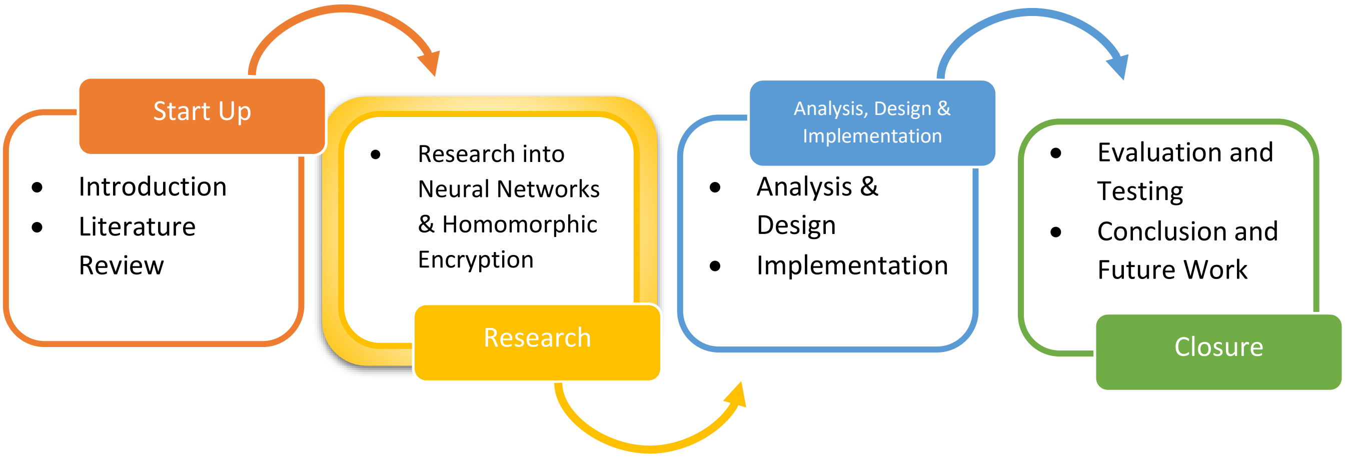
Summary of Part One

Part one has been constructed to serve as an informative introduction to the report that clearly lays out some of the initial key concepts of the report and creates a foundation to be developed upon further into the report. During this part we went through chapters One and Two which were composed of an introduction and a literature review. In the first chapter, an overview of the whole project was given that outlines all the key factors as to how the report was developed and what was intended to be provided to the reader by the author. This included the project’s aims, motivations, objectives and stakeholders. In addition, it listed the resources used and defined the measures of project success. In Chapter Two, a literature review was completed in which some initial conclusions were made as how to Neural Networks can be used with encrypted data. The conclusion was…

References for Part One

1. Hunt T, Song C, Shokri R, Shmatikov V, Witchel E (2018) Chiron: privacy-preserving machine learning as a service. arXiv: 1803.05961
2. Zheng Q, Wang X, Khurram Khan M, Zhang W, Gupta BB, Guo W (2018) A lightweight authenticated encryption scheme based on chaotic SCML for railway cloud service. IEEE Access 6:711– 722. https://doi.org/10.1109/ACCESS.2017.2775038

Part 2: Research



Start Up

- Introduction
- Literature Review

Research

- Research into Neural Networks & Homomorphic Encryption

Analysis, Design & Implementation

- Analysis & Design
- Implementation

Closure

- Evaluation and Testing
- Conclusion and Future Work

Chapter 3: Research into Neural Networks and Homomorphic Encryption

3.1 Research into Privacy-preserving neural networks with Homomorphic Encryption

Modern encryption techniques ensure security and are considered as the best option to protect stored data and data in transit from an unauthorised third-party. However, a decryption process is necessary when the date must be processed and or analysed, falling into the initial problem of data vulnerability. Fully Homomorphic Encryption (FHE) is considered the holy grail of cryptography, an elusive goal that could solve cybersecurity problems [1-3]. Within this section we will discuss the fundamental concepts of Fully Homomorphic Encryption, practical implementations, state-of-the approaches, limitations, advantages, disadvantages, potential applications, and development tools focusing on neural networks.

Furthermore, this section is also meant to be used as a reference for the following chapters at the Design and Implementation phase of the project. This chapter is meant as more of a guidance or general explanation of some of the core concepts to avoid confusion and to allow the reader to have a grasp some of the algorithms used.

3.2 What is Encryption?

To avoid forgery and ensure confidentiality of the contents of a letter, for centuries it has been a common practice for the originator of the letter to sign his/her name on it and then seal it in an envelope, before handing it over to the deliverer. Public key cryptography was discovered nearly five decades ago [4] has revolutionised the way for people to conduct secure and authenticated communications. It is now possible for people to communicate or send information with one another in a secure and authenticated way over an open and insecure network. In doing so the same two-step approach has been followed. Namely before the message/data is sent out, the sender of the message/data would sign it using a digital signature scheme, and then encrypt the message/data (and signature) using a private key encryption algorithm under a randomly chosen encryption key. The random encryption key would then be encrypted using the recipient's public key. This is called the two-step approach signature-then-encryption [5].

In public key cryptography we use two keys, one **public** and one **private**, related in a mathematical way. The public key can be published in a directory along with the user's name. Anyone who then wishes to send a file to the holder of the associated private key will take the public key, encrypt a message under it and send it to that key holder. The idea is that only the holder of the private key will be able to decrypt the message. Below shows an example of how the process works:

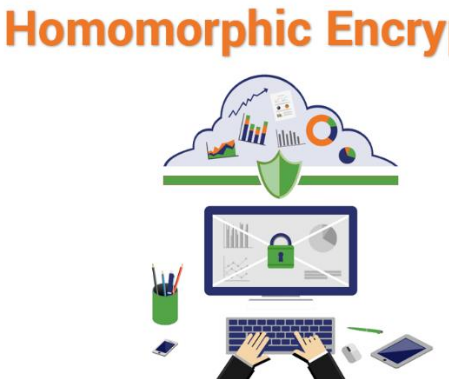
$$\text{File} + \text{Bob's public key} = \text{Ciphertext},$$
$$\text{Ciphertext} + \text{Bob's private key} = \text{File}$$

Hence anyone with Bob's public key can send Bob a secret message. But only Bob can decrypt the message, since only Bob has the corresponding private key.
Public keys systems work because the two keys are linked in a mathematical way, such that knowing the public key does not allow you to compute anything about the private key. But knowing the private key allows you to unlock information encrypted with the public key.

The concept of being able being able to encrypt using a key which is not kept secret was so strange it was not until 1976 that anyone thought of it. The idea was first presented in the seminal paper of Diffie and Hellman named *New Directions in Cryptography* [4]. It was not until a year or so later that the first and most successful system, namely RSA (Rivest-Shamir-Adleman), was invented.

3.2 Homomorphic Encryption

In this section, we discuss the basic concepts of Homomorphic Encryption and their evolution based on representative works in the area.



In the cryptograph field, the term Homomorphic Encryption defines a kind of encryption system able to perform certain computational functions over ciphertexts. The output maintains the features of the function and input format. The system has no access to information on ciphertexts and secret keys. It only uses publicly available information without risks of the data breach. As explained in the previous section the Homomorphic encryption concept refers to a mapping between functions on the space of messages and ciphertexts. A homomorphic function applied to ciphertexts provides the same (after decryption) result as applying the function to the original unencrypted data.

Let m_1 and m_2 be messages, c_1 and c_2 be their corresponding ciphertexts. The operation $\ddot{+}$ in an additively homomorphic encryption produces the ciphertext $c_+ \leftarrow c_1 \ddot{+}\, c_2$ that can be decrypted to $m_1 + m_2$.
Similarly, for $\ddot{\times}$ in a multiplicatively homomorphic encryption, it generates the ciphertext $c_\times \leftarrow c_1 \ddot{\times}\, c_2$ that is decrypted to $m_1 \times m_2$. Both encryptions obtain ciphertexts c_+ and $c_\times$, without knowing m_1 and m_2.

There are three types of homomorphic encryption cryptosystems: Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE). In the following subsections we will discuss their limitations and scopes.

3.2.1 Partially Homomorphic Encryption

Partially Homomorphic Encryption supports an unlimited number of one type of operation. For example, additive Homomorphic Encryption allows an unbounded number of additions but no multiplications.
Ronald Rivest, Adi Shamir, and Leonard Adleman (RSA) cryptosystem is the first multiplicative Partially Homomorphic Encryption [6]. In general, given two messages m_1 and m_2 and their respective ciphertexts $c_1 = (m_1^e) mod\ n$ and $c_2 = (m_2^e) mod\ n$, where e is chosen such that $\gcd(\mathrm{e}, \emptyset) = 1$ for $\emptyset = (q_1 - 1) \cdot (q_2 - 1)$ with large primes q_1 and q_2, and $n = q_1 \cdot q_2$. The ciphertext with the product of the original plaintexts is computed as

$$\begin{aligned} c_\times &\leftarrow (m_1 \cdot m_2)^e\ mod\ n = (m_1^e)\ mod\ n \cdot (m_2^e)\ mod\ n \\ &= c_1 \ddot{\times}\, c_2 \end{aligned}$$

RSA is not semantically secure as a result of its deterministic encryption algorithm. Taher El-Gamal is another relevant multiplicative PHE [7].

Shafi Goldwasser and Silvio Micali (GM) cryptosystem is the first additively Partially Homomorphic Encryption [8]. According to GM cryptosystem, there are two message m_1 and m_2 and their respective ciphertexts $c_1 = (b_1^2 \cdot e^{m_1}) mod\ n$ and $c_2 = (b_2^2 \cdot e^{m_2}) mod\ n$, where b_1^2 and b_2^2 are quadratic nonresidue values such that $\gcd(b_1^2, n) = \gcd(b_2^2, n) = 1$, and e is one of the quadratic nonresidue modulo n values with $\left(\frac{x}{n}\right) = 1$.
The GM scheme has a homomorphic property, where the encryption of $m_1 + m_2$, is

$$\begin{aligned} &c_+ \leftarrow [(b_1 \cdot b_2)^2 \cdot e^{m_1+m_2}]\ mod\ n \\ &= [(b_1^2 \cdot e^{m_1}) \cdot (b_2^2 \cdot e^{m_2})]\ mod\ n \\ &= (b_1^2 \cdot e^{m_1})\ mod\ n \cdot (b_2^2 \cdot e^{m_2})\ mod\ n = c_1 \ddot{+} c_2 \end{aligned}$$

However, GM is not an efficient scheme, as ciphertexts may be several hundred times larger than the initial plaintexts.
Relevant additive PHE cryptosystems are invented by and named after Josh (Cohen) Benaloh in 1994 [9], David Naccache and Jacques Stern (NS) in 1997 [10], Tatsuaki Okamoto and Shigenori Uchiyama (OU) in 1998 [11], Pascal Paillier in 1999 [12], Ivan Damgård and Mads Jurik (DJ) in 2001 [13], Steven Galbraith in 2002 [14], and Akinori Kawachi, Keisuke Tanaka and Keita Xagawa (KTX) in 2007 [15].

The encryption process in Partially Homomorphic Encryption does not guarantee a given level of security. The worst-case hardness of "noisy" problems is one direction in the security solution. The noise term denotes a moderate quantity of error injected in the encrypted message and generates a not exact relation [16].

3.2.2 Somewhat Homomorphic Encryption

Somewhat Homomorphic Encryption supports a predetermined amount of different homomorphic operations, limiting the number of allowed operations. Each operation increases the underlying noise, so its correct evaluation depends on performing only a bounded number of actions. Message decryption fails when noise overpasses a certain threshold.
Dan Boneh, Eu-Jin Goh, and Kobbi Nissim (BGN) scheme [17] was the first approach that allowed both additions and multiplications with constant-size ciphertexts. BGN hardness is based on the subgroup decision problem [18], which decides whether an element is a member of a subgroup G_p of group G of order $n = q_1 \cdot q_2$. In BGN, ciphertexts $c_1 = g^{m_1} \cdot h^{e_1}$ and $c_2 = g^{m_2} \cdot h^{e_2}$ encrypts m_1 and m_2 messages, where g and u are two random generators from G, $h = u^{q_2}$ is a random generator of the subgroup of G of order q_1 and random numbers e_1 and e_2 from the set $\{0,1, \dots, n - 1\}$.
The encryption of $m_1 + m_2$ is computed as:

$$\begin{aligned} &c_+ \leftarrow g^{m_1+m_2} \cdot h^{e_1+e_2+e} = (g^{m_1} \cdot h^{e_1}) \cdot (g^{m_2} \cdot h^{e_2}) \cdot h^e \\ &= c_1 \cdot c_2 \cdot h^e = c_1 \ddot{+} c_2 \end{aligned}$$

Nonetheless, BGN is impractical due to it computing c_+ only one using the bilinear map property, which maps $s : G \times G = G_1$, where G_1 is a group of order $n = q_1 \cdot q_2$.
Let $g_1 = s(g,g)$ and $h_1 = s(g,h)$, where g_1 is of order n and h_1 is of order q_1. Thus, there is α such that $h = g^{\alpha q_2}$.
The encryption of $m_1 \cdot m_2$ is computed as:

$$\begin{aligned} &C_\times \leftarrow g_1^{m_1 m_2} \cdot h_1^{m_1 e_2 + e_2 m_1 + \alpha q_2 e_1 e_2 + e} \\ &= g_1^{m_1 m_2} \cdot h_1^{m_1 e_2 + e_2 m_1 + \alpha q_2 e_1 e_2} \cdot h_1^e \\ &= g_1^{m_1} \cdot g_1^{\alpha q_2 (m_1 e_2 + e_2 m_1 + \alpha q_2 e_1 e_2 + e)} \cdot h_1^e \end{aligned}$$

$$
\begin{aligned}
&= g_1^{m_1 m_2 + \alpha q_2 (m_1 e_2 + e_2 m_1 + \alpha q_2 e_1 e_2 + e)} \cdot h_1^e \\
&= s(g, g)^{(m_1 + \alpha q_2 e_1)(m_2 \alpha q_2 e_2)} \cdot h_1^e \\
&= s(g^{m1 + \alpha q_2 e_1}, g^{m_2 + \alpha q_2 e_2}) \cdot h_1^e \\
&= s(g^{m_1} \cdot g^{\alpha q_2 e_1}, g^{m_2} \cdot g^{\alpha q_2 e_2}) \cdot h_1^e \\
&= s(g^{m_1} \cdot h^{e_1}, g^{m_2} \cdot h^{e_2}) \cdot h_1^e \\
&= s(c_1, c_2) \cdot h_1^e = c_1 \ddot{\times} c_2
\end{aligned}
$$

where $m_1 e_2 + e_2 m_1 + \ \alpha q_2 e_1 e_2 + e$ is uniformly distributed in $\mathbb{Z}_{\mathrm{N}}$, and $c_{\times}$ is uniformly distributed exncryption of $(m_1 \cdot m_2) mod\ n$, but now in G_1 rather than G. However, BGN is still additively homomorphic in G_1.

3.2.3 Fully Homomorphic Encryption concept

3.2.4 Timeline

CREATE A TIMELINE FROM https://link.springer.com/content/pdf/10.1007/s12083-021-01076-8.pdf

3.3 Fully Homomorphic Encryption

3.4 Neural Networks

A Neural Network is a computing system inspired by biological brains. Here, we consider a neural network that is composed of a population of artificial neurons arranged in layers.

3.5 Limitations

3.6 Development Tools

3.7 Review of Current Systems

Research on the 2/3 different developed methods (Microsoft, etc…)

3.7.1 CryptoNets - Microsoft

- Cryptonets – **Microsoft**
  - https://github.com/microsoft/CryptoNets

3.7.2 XGBoost – Amazon

- XGBoost – **Amazon** - https://www.amazon.science/blog/machine-learning-models-that-act-on-encrypted-data
  - https://github.com/awslabs/privacy-preserving-xgboost-inference

3.7.3 Others

- Discretized Neural Networks - https://eprint.iacr.org/2017/1114.pdf
- CryptoNN

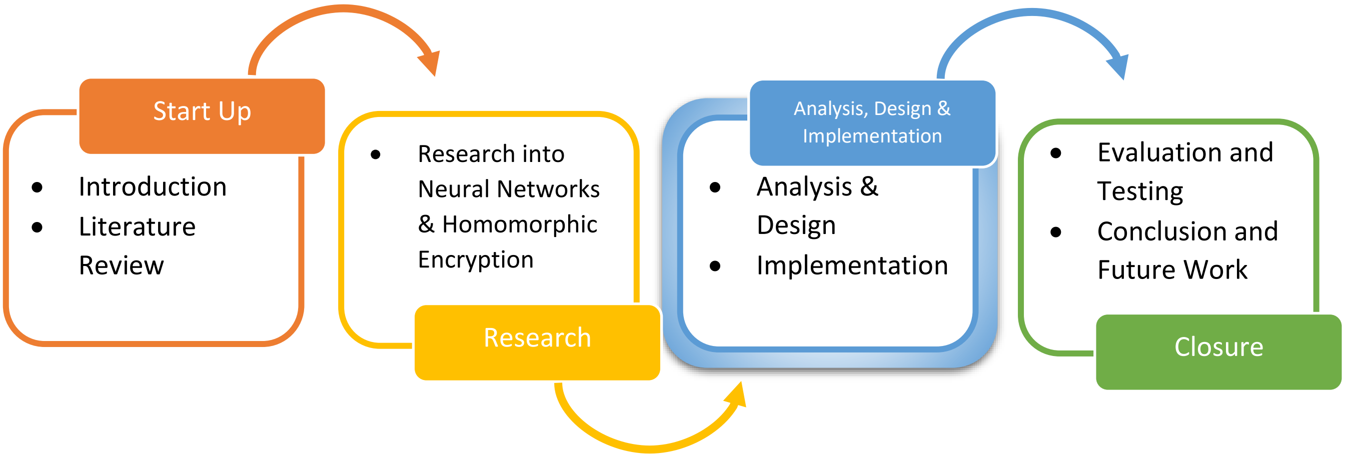
Summary of Part Two

References for Part Two

1. Vaikuntanathan V (2011) Computing blindfolded: new developments in fully Homomorphic Encryption. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science. Palm Springs, CA, pp 5–16. https://doi.org/10.1109/FOCS.2011.98
2. Gentry C (2009) A fully Homomorphic encryption scheme. In: Stanford University. Stanford, PhD Thesis
3. Gentry C, Halevi S (2011) Implementing gentry's fully homomorphic encryption scheme. In: Paterson KG (ed) Advances in Cryptology – EUROCRYPT 2011. Lecture notes in computer science, vol 6632. Springer, Berlin, Heidelberg, pp 129–148. https://doi.org/10.1007/978-3-642-20465-4_9
4. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory **IT-22** (1976) 472–492.
5. https://link-springer-com.surrey.idm.oclc.org/chapter/10.1007%2FBFb0052234
6. Rivest R, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21:120–126. https://doi.org/10.1145/359340.359342
7. ElGamal T (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory 31: 469–472. https://doi.org/10.1109/TIT.1985.1057074
8. Goldwasser S, Micali S (1982) Probabilistic encryption and how to play mental poker keeping secret all partial information. In: Proceedings of the fourteenth annual ACM symposium on Theory of computing (STOC '82). ACM, New York, USA, pp 365–377. https://doi.org/10.1145/800070.802212
9. Benaloh J (1994) Dense probabilistic encryption. Proceedings of the workshop on selected areas of cryptography, In, pp 120–128
10. Naccache D, Stern J (1998) A new public key cryptosystem based on higher residues. In: Proceedings of the 5th ACM conference on Computer and communications security (CCS '98). ACM, New York, USA, pp 59–66. https://doi.org/10.1145/288090.288106
11. Okamoto T, Uchiyama S (1998) A new public-key cryptosystem as secure as factoring. In: Nyberg K (ed) Advances in Cryptology — EUROCRYPT'98. Lecture notes in computer science, vol 1403. Springer, Berlin, Heidelberg, pp 308–318. https://doi.org/ 10.1007/BFb0054135
12. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: Stern J (ed) Advances in Cryptology - EUROCRYPT '99. Lecture notes in computer science, vol 1592. Springer, Berlin, Heidelberg, pp 223–238. https:// doi.org/10.1007/3-540-48910-X_16
13. Damgård I, Jurik M (2001) A Generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Kim K (ed) Public Key Cryptography. PKC 2001. Lecture Notes in Computer Science, vol 1992. Springer, Berlin, Heidelberg, pp 119–136. https://doi.org/10.1007/3-540-44586-2_9
14. Galbraith SD (2002) Elliptic curve paillier schemes. J. Cryptology 15:129–138. https://doi.org/10.1007/s00145-001-0015-6
15. Kawachi A, Tanaka K, Xagawa K (2007) Multi-bit cryptosystems based on lattice problems. In: Okamoto T, Wang X (eds) Public Key Cryptography – PKC 2007. Lecture Notes in Computer Science, vol 4450. Springer, Berlin, Heidelberg, pp 315–329. https://doi.org/10.1007/978-3-540-71677-8_21
16. Minelli M (2018) Fully homomorphic encryption for machine learning. In: PSL Research University. PhD Thesis, Paris
17. Boneh D, Goh EJ, Nissim K (2005) Evaluating 2-DNF formulas on ciphertexts. In: Kilian J (ed) Theory of cryptography. TCC 2005, Lecture Notes in Computer Science, vol 3378. Springer, Berlin, Heidelberg, pp 325–341. https://doi.org/10.1007/978-3- 540-30576-7_18

18. Gjøsteen K (2004) Subgroup membership problems and public key cryptosystem. In: Norwegian University of Science and Technology. PhD Thesis, Trondheim

Part 3: Analysis, Design & Implementation


Start Up
Introduction
Literature Review
Research into Neural Networks & Homomorphic Encryption
Research
Analysis, Design & Implementation
Analysis & Design
Implementation
Evaluation and Testing
Conclusion and Future Work
Closure

Chapter 4: Analysis & Design

4.1

THIS IS WHERE MY PERSONAL IMPLEMENTATION ON A DATASET IS INCLUDED.

FOCUS IS ON THE METHODS NOT THE RESULTS.

TALK ABOUT PROGRAMMING LANGUAGE. USAGE/LIMITATIONS

https://link.springer.com/content/pdf/10.1007/s12083-021-01076-8.pdf

^ This has an example of an implementation but doesn’t provide the code.

Chapter 5: Implementation

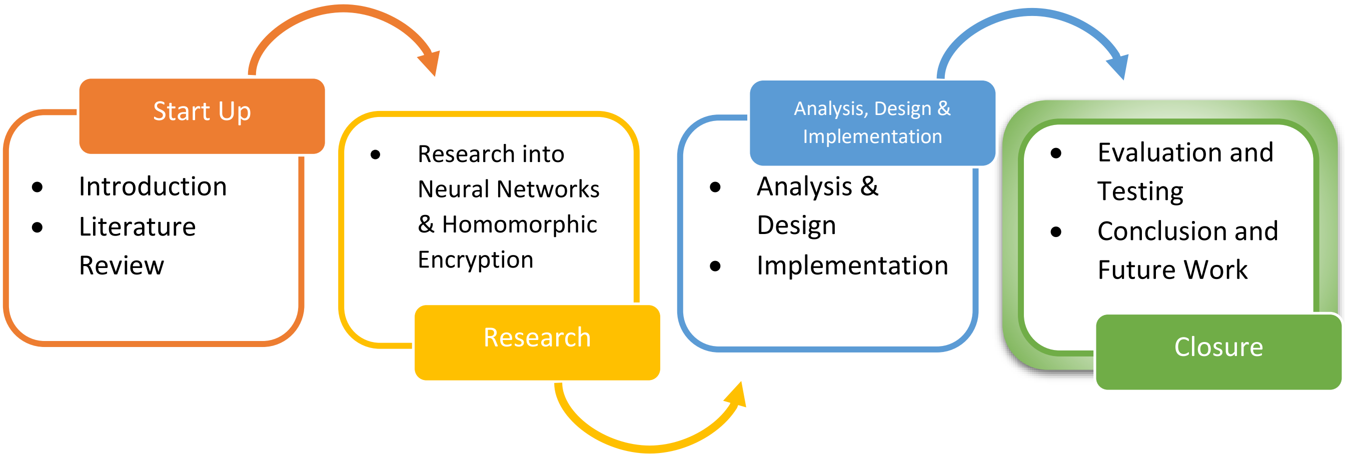
IMPLEMENTATION ON DATASET – TALK ABOUT THE WHOLE PROCESS, EVERY LITTLE THING

RESULTS ARE TALKED ABOUT IN THE NEXT SECTION

Summary of Part Three

References for Part Three

Part 4: Closure



Start Up

- Introduction
- Literature Review

- Research into Neural Networks & Homomorphic Encryption

Research

Analysis, Design & Implementation

- Analysis & Design
- Implementation

- Evaluation and Testing
- Conclusion and Future Work

Closure

Chapter 6: Evaluation & Testing

6.1 Defining an Evaluation Method

6.2 Proposed Evaluation Methods

6.3 Comparing the Systems

Chapter 7: Conclusion and future work

7.1 Evaluation of Overall Work

7.2 Work Achieved

7.3 Challenges

Although the Homomorphic Encryption standards, platforms, and implementations presented in this report help advancing Privacy Preserving Neural Networks, there are still some open challenges to be solved: overhead, performance, interoperability, bootstrapping bottlenecks, sign determination, common frameworks, etc.

Overhead

Homomorphic Encrypted Neural Networks has a significant overhead compared with its unencrypted analogous, making it impractical for many applications. The training phase of neural networks consists of a computationally intensive task for non-homomorphic encrypted models. With homomorphic encryption, it becomes more challenging even with modern day technology and computational power. A new tendency is to avoid the training phase by using pre-trained models to achieve a balance between complexity and accuracy.

Parallelization

One way to deal with the computational overhead is to incorporate well-known and new parallelizing techniques. Homomorphic Encrypted Neural Network models can be adapted to use high performance computing, distributed systems, and specialized resources. Multi-core processing units (GPU, FPGA, etc.) or customized chips (ASIC) technologies give the possibility of friendlier and efficient homomorphic encrypted neural networks environments. Another way to improve the overall efficiency is related to the possibility of batching and parallelizing several bootstrapping operations together.

Polynomial Approximation

A crucial challenge in developing NN-HE consists of the computational design for the homomorphic processing of the neuron’s inner functions. NN-HE requires operations not supported by HE, so it is necessary to find cryptographically compatible replacement functions to operate over encrypted data.

The activation function is an essential element in the construction of an effective NN-HE. It determines NN-HE accuracy and computational efficiency. Moreover, activation functions have a significant effect on the converging speed of the network. Also, its derivative, also known as gradient, is fundamental in the training phase.

Multiple approaches address the limitation by polynomially approximating non-compatible functions with a cryptographically consistent polynomial form. These functions should exhibit a trade-off between complexity and accuracy, limiting the efficiency of conventional approximation techniques [117].

In practice, an inadequate approximation function can result in poor performance and long processing time of NN-HE. Moreover, it produces larger encrypted messages that increase memory use.

The challenge of designing a cryptographically computable approximation of the activation function is in identifying low-degree polynomial with a minimal error and good accuracy.

Levelled Homomorphic Encryption Schemes

Another vital direction focuses on designing schemes without bootstrapping that supports the NN evaluation of bounded (pre-determined) depth. Such leveled HE schemes dramatically improves performance by removing the bottleneck and complexity generated by the bootstrapping recrypt function. However, this approach limits the deep learning implementation. While it is efficient for bounded NN-HE, the complexity may become undesirably high for deep learning models.

Binary Neural Networks

(BNN) are emerged as an area of opportunity to achieve blind non-interactive NN-HE models. Since the space of functions is restricted, the solution should be limited by the number of possible inputs and outputs. In BNN, every layer maps a binary input to a binary output, using a set of binary weights and a binary activation function. For the bias, it applies a batch normalization before each activation function. The input data is binarized using a predefined threshold.

In general, data and weights in non-standard binary representations {−1, 1} can be mapped to binary space {0, 1} by replacing −1 with 0. The weighted-sum can be performed by an element-wise product. It uses the logical operator XNOR and subsequently sums the result of the previous step by counting the number of ones. The binary activation function f(y) returns 1 if $y > 0$ and −1 otherwise.

Interoperability

Interoperability of existing ML tools is another challenging problem to achieve friendly NN-HE models. Popular NN frameworks have simplified the development of novel NN methods, but they do not provide HE support. The development of NN-HE depends on the current tools and their flexibility to supply or incorporate new approaches. The low flexibility of several HE libraries restricts their interaction with other frameworks. It makes more complicated the design, testing, and implementing new models, hence, increases the development time.

Automatization

The development of HE applications implies manual configuration and high expertise in different domains: scheme-specific optimizations, complicated security parameter setting, low-level programming, among others. Improper setup can generate low performance, encryption insecurity, and corrupted or unrecoverable information. The automatization and simplification of the development lifecycle are required. The implementation should be easily employed by beginners and highly configurable for expert users.

Common Framework

Most of the related works focus on specific environments with different characteristics. It limits the possibility of comparing new approaches to state-of-art algorithms. A standard framework can simplify the comparative analysis and show the advantages of new models. It should simplify the adoption of libraries, algorithms, measures, and statistical analysis.

7.4 Evaluation against the objectives

LOOK AT PART 1.3 AND COMPARE

7.5 Reflections and Potential Future Work

7.6 Conclusion

Easy deploying of machine learning in clouds makes Homomorphic Encryption a very important mechanism to solve online security and privacy.

References for Part Four

Appendix

To facilitate understanding the described ideas, we summarise the main terminology used throughout the report. Find below the general acronyms, main notations and terms used in this paper.

Acronyms

Notation

Terminology