Homomorphic Encryption Friendly Multi-GAT for Information Extraction in Business Documents

Djedjiga Belhadj[a], Yolande Belaïd[b] and Abdel Belaïd[c]
Université de Lorraine-LORIA,Campus Scientifique, 54500 Vandoeuvre-Lès-Nancy, France

Keywords: Information Extraction, Homomorphic Encryption, Polynomial Approximation, Attention Mechanism.

Abstract: This paper presents a homomorphic encryption (HE) system to extract information from business documents. We propose a structured method to replace the nonlinear activation functions of a multi-layer graph attention network (Multi-GAT), including ReLU, LeakyReLU, and the attention mechanism Softmax, with polynomials of different degrees. We also replace the normalization layers with an adapted HE algorithm. To solve the problem of accuracy loss during the approximation, we use a partially HE baseline model to train a fully HE model using techniques such as distillation knowledge and model fine-tuning. The proposed HE-friendly Multi-GAT models the document as a graph of words and uses the multi-head attention mechanism to classify the graph nodes. The first partially HE-Multi-GAT contains polynomial approximations of all ReLU, LeakyReLU and the attention Softmax activation functions. Normalization layers are used to handle values exploding when approximating all the nonlinear activation functions. These layers are approximated as well using an adapted algorithm that doesn't rely on the training data and minimizes performances loss while avoiding connections between the server and the data owner. Experiments show that our approach minimizes the model accuracy loss. We tested the architecture on three different datasets and obtained competitive results (F1-scores greater than 93%).

1 INTRODUCTION

One of the most recent data protection regulations, the General Data Protection Regulation (GDPR), aims to protect the privacy of personal data of European residents. Therefore, any processing of personal data, e.g. collecting, recording, storing or extracting is covered by GDPR.

Evaluating neural network inference models on GDPR cloud environment is a critical task for many industries, including finance. For example, when extracting information from confidential personal documents, such as payslips and invoices, it may be necessary to encrypt the data before uploading to the cloud. It is also important to ensure that the cloud inference model can accurately evaluate the encrypted data. Homomorphic encryption (HE) allows calculations to be performed on encrypted data. It is a technique to maintain the security of private data in untrusted environments. Multiple encryption protocols could be used to ensure data and model confidentiality. These protocols provide a limited number of arithmetic operations and functions.

Most high-performance information extraction systems from business documents have complex architectures with a large number of trainable parameters. Most state-of-the-art systems are not compatible with existing encryption protocols due to the nature of the nonlinear functions used. The use of such systems in secured and encrypted platforms requires them to be adapted to the existing encryption protocols. Here we aim to tackle the main issue of the HE protocol by presenting a fully homomorphic Multi-GAT. This model approximates the non-polynomial operations while minimizing the loss of accuracy, all without requiring multiple client and cloud server connections.

HE systems only support basic arithmetic operations like addition and multiplication and only a limited number of consecutive multiplications is possible. These systems use bootstrapping operations to enable additional computations. The bootstrapping process is quite costly in terms of execution time, so reducing the multiplication depth can reduce or eliminate the bootstrapping process while making the overall computation easier.

[a] https://orcid.org/0000-0003-0548-3948
[b] https://orcid.org/0000-0002-2611-751X
[c] https://orcid.org/0000-0002-9107-1204

Belhadj, D., Belaïd, Y. and Belaïd, A.
Homomorphic Encryption Friendly Multi-GAT for Information Extraction in Business Documents.
DOI: 10.5220/0012432400003654

In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2024)*, pages 817-825
ISBN: 978-989-758-684-2; ISSN: 2184-4313

One way to overcome this constraints is to employ client-assisted design (Lloret-Talavera et al., 2021). In this case, the computationally complex operation is sent to the data owner who decrypts the data, carries out the calculation, encrypts the result and sends it to the cloud for further computation. Due to its complex communication processes, increased attack surface and open vulnerability to external attacks, we aim to avoid this approach. Another way is to substitute an operation with one that is similar but different and more HE-friendly. For example, a max-pooling operation could be replaced with an HE-friendly average-pooling operation (Gilad-Bachrach et al., 2016). The third option to overcome this constraint is to approximate non-linear functions through polynomial approximations (Hesamifard et al., 2017; Lee et al., 2022; Mohassel and Zhang, 2017).

In this paper, we present a novel approach that converts a Multi-GAT node classifier which makes use of the standard attention mechanism, into a HE-friendly model. The Multi-GAT baseline includes the ReLU, LeakyReLU, and Softmax functions, as well as normalization layers. We use customized polynomial activation functions to replace ReLU, LeakyReLU, and Softmax activations. In order to keep the values within a given range and to prevent them from exploding, we also replace the normalization layer with an approximation algorithm. To train the final HE-friendly model, we use Knowledge Distillation (KD) and fine-tuning. Moreover, our method enables execution of the inference process in a cloud environment without the requirement of data owner interaction. The experiments demonstrate that the HE-friendly model's inference accuracy is comparable to that of the original Multi-GAT model.

The paper is composed of the following sections: Section 2 outlines the state-of-the-art approaches that approximate the various activation functions and the normalization layer. Section 3 describes the proposed He-friendly Multi-GAT approach in detail. Section 4 presents the experiments conducted and results obtained. Section 5 concludes the paper, highlighting the overall contribution of the system.

2 RELATED WORK

Several recent approaches have proposed different approximations of activation functions and nonlinear layers. Existing privacy-preserving machine learning (PPML) approaches, proposed in the literature, can be classified into interactive and non-interactive approaches. The first category requires a connection between the client and the server, while the second category does not require any connection and all computations are performed on the server.

The ReLU activation function is the most approximated function in the literature. The authors of (Gilad-Bachrach et al., 2016; Ghodsi et al., 2017; Mohassel and Zhang, 2017; Liu et al., 2017) replace the ReLU function with a simple square function. Even though the square function is a simple function, it can reduce the performance of the model. Other methods try to minimize the difference between the ReLU and the polynomial approximation within a interval $[a,b]$ using multiple techniques like least square (Davis, 1975) method, derivatives based calculation proposed by (Chiang, 2022) and minimax optimisation algorithm. All the approaches proposed in (Chiang, 2022; Wang et al., 2022; Ishiyama et al., 2020; Zheng et al., 2022) methods have proposed different polynomial approximations using least square method to calculate the polynomial coefficients. Both (Chiang, 2022) and (Wang et al., 2022) use polynomial of degree 2, (Ishiyama et al., 2020) used a 4 degree polynomial, while (Zheng et al., 2022) approximated the ReLU using a 3 degree polynomial in the complex reference. (Chiang, 2022) and (Zheng et al., 2022) proposed also polynomials of degree 2 and 3 using the derivatives based calculation. (Ali et al., 2020) suggested a second degree polynomial using minimax optimisation algorithm.

On the other hand, (Baruch et al., 2022) propose a second-degree polynomial with two trainable coefficients (a and b) to be learned individually for each layer during the training process. To learn these two coefficients they use a smooth transition approach. They first train the model with ReLU activation layers for the first e_0 epochs, and during the rest of the epochs, they smoothly switch from ReLU functions to polynomial activation functions. After that, they continue to train the model using only the approximations. Similarly, (Qian et al., 2023) use a set of trainable low-order Hermite polynomials that are linear combinations of the first three terms of Hermite polynomials. This implies three trainable weight parameters learnable during model training. They also adapt a smooth transition between the ReLU-based model and the Hermite-based model during training by adding the difference between the outputs of the activation layer of the ReLU-based model and those of the Hermite polynomial model to the model output loss.

Other papers have proposed approximations to the Softmax function. Some methods propose non linear functions that need the use of the MPC (secure multi-party computation) protocol and others use polynomial approximations that don't use the MPC.

(Ali et al., 2022) replace the Softmax by a sigmoid function and then approximate it by the three degree polynomial suggested by (Kim et al., 2018). (Chiang, 2023) also replace the Softmax by the sigmoid function and then use the least square method to retrieve a polynomial of degree 11, this latter is then approximated with a three degree polynomial. Whereas (Al Badawi et al., 2020) replace the Softmax with a two degree polynomial using the Minimax approximation algorithm (Meinardus, 2012). In the other hand, (Mohassel and Zhang, 2017) replace the Softmax with a ReLUSoftmax, by simply replacing the exponential by a ReLU function in the Softmax formula. (Chen et al., 2022) as well propose another formula based on the ReLU function and a three layers linear neural network. (Li et al., 2022) suggest the 2Quad function where the exponential in the Softmax formula is replaced by $(x+c)^2$.

There are also (Dathathri et al., 2019; Jang et al., 2022) that propose the approximation of any non linear function by polynomials of different degrees (2,3,5 or 7) with learnable coefficients.

Other works propose approximations to the normalization and batch-normalization layers. (Chen et al., 2022) replace the normalization layer by the formula $x * \gamma + \beta$, where γ and β are learnable parameters. Ibarrondo and Önen in (Ibarrondo and Önen, 2018; Lou and Jiang, 2021) and (Liao et al., 2019) reformulate the batch normalization layer by proposing an approximation that could be mitigated by proposing an approximation that can be mitigated by reparameterizing the weights and biases of the layers before the normalization layer. Once the network is trained, the mean and variance values are fixed during inference (Ioffe and Szegedy, 2015). This is done by calculating unbiased estimates of the mean and variance across all batches. The normalization coefficients are learned from the server's training data and not from the client's input data.

3 PROPOSED METHOD

In this section we will first present the task of information extraction in business documents, our Multi-GAT baseline and then we explain our proposed He-Mutli-GAT Model.

3.1 Information Extraction in Business Documents

In this paper we focus on the extraction of information from business documents, including invoices, payslips and receipts. The information to extract includes named entities such as names, customer or company identifiers, dates and addresses. These confidential documents typically take a semi-structured format. This means they do not follow a standard template and can have varying layouts. For example, the same information may appear in different places in two invoices from different providers. Nevertheless, these documents present key information with a context that helps to identify its class.

3.2 The Multi-GAT Baseline Model

The document image is modeled as a graph of words where each word is represented by a fusion of its multimodal features (textual, 2D positional and visual) as proposed in (Belhadj et al., 2023b). The word multimodal features vector is obtained by concatenating the results of two Dense layers applied to the three modalities. The first Dense layer combines the textual features got by the pre-trained BPEmb (Heinzerling and Strube, 2018) model with the normalized position of each word's bounding box. The second Dense layer is applied to the word visual features resulting from the application of a pre-trained ResNet followed by the calculation of the region of interest (ROI).

Each word is connected to its nearest neighbors distributed on three lines (the line of the word, the one above and below) as proposed in (Belhadj et al., 2023a; Belhadj et al., 2023b). The document graph is represented by two matrices X and A: $G = (X, A)$, where X is the features matrix and A is the adjacency matrix and than is fed into a Multi-GAT model. This latter is composed of four layers of multi-head graph attention networks and three normalization layers and it outputs the graph nodes classes.

3.2.1 Multi-Head Attention Mechanism

A multi-head attention mechanism of k heads is a combination of K attention mechanisms. In the three first GAT layers, the hidden vector $\overrightarrow{h'}_i$ of the node i is calculated using the attention mechanism:

$$\overrightarrow{h'}_i = \|_{k=0}^{K} \sigma\Big(\sum_{j \in N_i} \alpha_{ij}^k W^k \overrightarrow{h}_j\Big) \tag{1}$$

where $\|$ represents the concatenation. In the output layer, an average is used instead of concatenation:

$$\overrightarrow{h'}_i = \sigma\Big(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in N_i} \alpha_{ij}^k W^k \overrightarrow{h}_j\Big) \tag{2}$$

σ is the activation function, α_{ij}^k are normalized attention coefficients calculated by the k-th attention

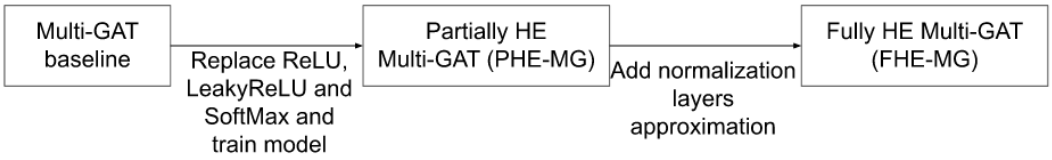


Figure 1: Approximation operations replacement in the Multi-GAT baseline: we begin by replacing the activation functions to form the partially HE Multi-GAT, then by adding the normalization layers approximations, we obtain the final Fully HE Multi-GAT.

mechanism (a^k) and W^k is the weight matrix of the corresponding input linear transformation.

The α_{ij}^k coefficients calculated by the attention mechanism correspond to:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k\in N_i} exp(e_{ik})} \tag{3}$$

$$\alpha_{ij} = \frac{exp(LeakyReLU(\overrightarrow{a}^T[W\overrightarrow{h}_i||W\overrightarrow{h}_j]))}{\sum_{k\in N_i} exp(LeakyReLU(\overrightarrow{a}^T[W\overrightarrow{h}_i||W\overrightarrow{h}_k]))} \tag{4}$$

$\overrightarrow{a}^T$ is a learned attention weights matrix.

3.2.2 Nonlinear Operations

The set of nonlinear operations used in the Multi-GAT baseline model which must be approximated to form a HE Multi-GAT, are as follows:

- The ReLU activation function applied to each GAT output
- The LeakyReLU and the softmax in the calculation of the attention
- The normalization layer applied between the GAT layers

The first and most studied function to approximate is the ReLU function.

$$ReLU(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{else.} \end{cases} \tag{5}$$

This function is modified slightly to form the LeakyReLU function defined as follows:

$$LeakyReLU(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{else.} \end{cases} \tag{6}$$

Whereas the Softmax activation function is defined as shown in the equation:

$$Softmax(x) = \frac{exp(x_i)}{\sum_{k\in N_i} exp(x_k)} \tag{7}$$

As can be seen in the equation, the Softmax contains the exponential and the division operation. Both these operations are non homomorphic.

3.2.3 Normalization Layer

The baseline normalization layer is defined as follows:

$$y = \frac{x-\mu}{\sqrt{\sigma+\varepsilon}} * \gamma + \beta \tag{8}$$

where $\mu = \frac{1}{N}\sum_{i\in N} x_i$, $\sigma = \frac{1}{N}\sum_{i\in N}(x_i-\mu)^2$ and γ and β are respectively learned scaling and offset factors.

3.3 HE-Friendly Mutli-GAT Model

In this part we will detail our approach to approximate all the nonlinear operations in the Muli-GAT basline in order to adapt it to the HE protocol.

3.3.1 Activation Layers Approximation

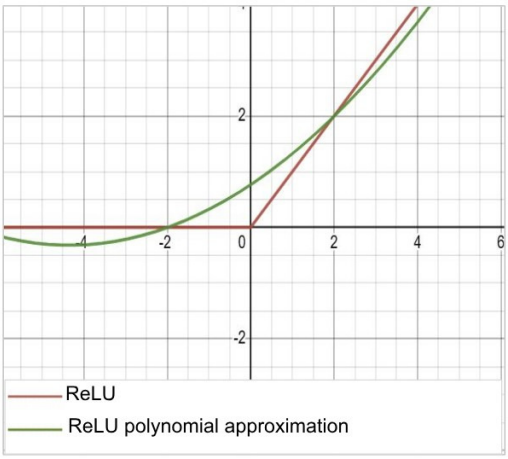
We replace the activation functions, namely: the ReLU, LeakyReLU and the Softmax by low degree polynomials as shown in the Table 1. The ReLU and LeakyReLU are approximated by two fixed coefficients polynomials whereas the Softmax is approximated by a two degree polynomial with variable coefficients initialized with the coefficients of the Softmax polynomial mentioned in Table 1.

Table 1: Polynomial approximations of the activation functions.

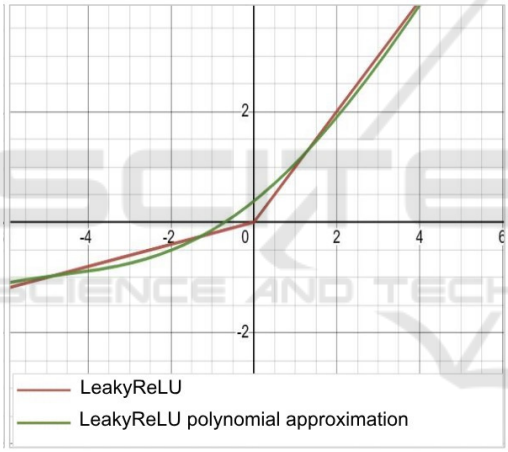
Activation function	Polynomial approximation
ReLU	$0.765 + 0.499x + 0.0574x^2 + 2.2865e^{-11}x^3$
LeakyReLU	$-6.056e^{-4}x^4 - 1.101e^{-17}x^3 + 8.048e^{-2}x^2 + 0.6x + 0.382$
Softmax	$0.25 + 0.5x + 0.125x^2$ (Al Badawi et al., 2020)

We use the least square method in the range $[-6,6]$ to generate five polynomial approximations of different degrees (from 2 to 6) for the ReLU and LeakyReLU functions. We replace the three Multi-GAT non-linear functions with these generated polynomials and others proposed in the literature. We then evaluate the resulting models and select the approximations that achieve the best performance. The cho-

sen polynomials are detailed in Table 1. ReLu and LeakyReLU approximations are also shown in Figure 2.



(a) Comparison of the ReLU function and its polynomial approximation.



(b) Comparison of the LeakyReLU function and its polynomial approximation

Figure 2: Comparison of ReLU and LeakyReLU functions and their polynomial approximations

3.3.2 Normalization Layer Approximation

We replace each normalization layer by its approximation as can be seen in Algorithm 1. We first scale the normalization input by multiplying it by *x_scale* which is a value between 0 and 1 that aims to reduce the range of the input variation. We calculate after that the μ and σ of the new input. To approximate the square inverse of $\sigma+\varepsilon$ in the normalization formula, we use the method proposed in (Panda, 2022) summed up in the Algorithm 2.

Here we approximate the *sgn* function by the function composition proposed in (Cheon et al., 2020):

Data: x, x_scale, ε
Result: y
$x \leftarrow$ x * x_scale;
$\mu \leftarrow \frac{1}{N}\sum_{i\in N} x_i$;
$\sigma \leftarrow \frac{1}{N}\sum_{i\in N}(x_i-\mu)^2$;
$\sigma_{Sqr\_Inv} \leftarrow Sqr\_Inv\_Appr(\sigma+\varepsilon)$;
$y \leftarrow (x-\mu)*\sigma_{Sqr\_Inv}*\gamma+\beta$;
return y;

Algorithm 1: $Poly\_Norm_i$.

Data: $[a,b],\varepsilon,d,k_1,k_2,x_1,x_2,P,x,err$
Result: $y_d = \frac{1}{\sqrt{x}}$
$\beta(P,x) \leftarrow comp(\frac{P}{b-a}, \frac{x}{b-a})$;
/* $comp(x,y) = \frac{1+sgn(x-y)}{2}$ */
$L_1 \leftarrow \frac{-1}{2}k_2 * x_1^{\frac{-3}{2}} * x + \frac{3}{2}\frac{k_2}{\sqrt{x_1}}$;
$L_2 \leftarrow \frac{-1}{2}k_2 * x_2^{\frac{-3}{2}} * x + \frac{3}{2}\frac{k_2}{\sqrt{x_2}}$;
$h_0(x) \leftarrow (1+err-\beta(x))*L_1(x)+(\beta(x)-err)*L_2(x)$;
for $i \leftarrow 1$ *to* d **do**
  $y_i \leftarrow \frac{1}{2}*y_{i-1}(x*y_{i-1}^2+3)$; /* Computed Newton's iterations to obtain y_d */
end
return y_d;

Algorithm 2: Sqr_Inv_Appr (Square inverse approximation $\frac{1}{\sqrt{x}}$).

$sgn(x) = f_3^{d_f}(x)og_3^{d_g}(x)$. The polynomial $f_3(x)$ and $g_3(x)$ are: $f_3(x) = \frac{1}{2^4}(35x-35x^3+21x^5-5x^7)$ and $g_3(x) = \frac{1}{2^{10}}(4589x-16577x^3+25614x^5-12860x^7)$. We use the same parameters values used in (Panda, 2022) for the $[a,b] = [10^{-4}, 10^3]$

3.4 Training Process

To obtain the final inference model, we first start by pre-training a partially HE-friendly Multi-GAT (we note it PHE-MG) by replacing the three activation functions with their polynomial approximations and keeping the original normalization layers. We obtain the first baseline (PHE-MG) and use it to gradually replace the normalization layers and obtain the FHE-MG as shown in the Figure 1 and explained in the Algorithm 3.

Each time we replace a normalization layer, knowledge distillation is used by freezing the model weights and calculating the loss between the normalization layer and its approximation layer. After each normalization layer replacement, we adjust the model

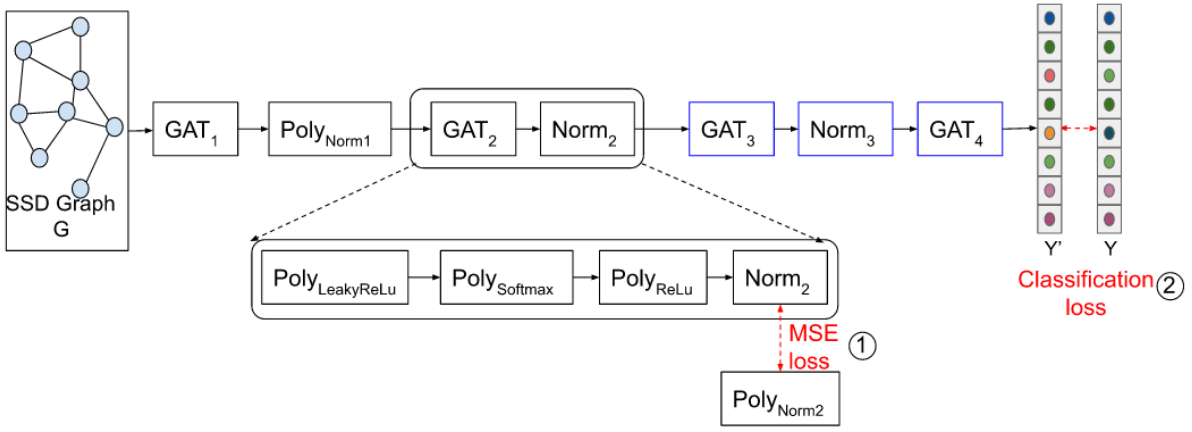


Figure 3: Approximation workflow overview: step 1 shows the normalization approximation and the knowledge distillation while step 2 shows the fine tuning step, the blue boxes are the layers to update during the second step.

weights to the new normalization parameters by fine-tuning all the model layers between the replaced layer and the model output as shown in the Figure 3 (the blue boxes).

Data: MG = PHE-MG, x_scale_tab, ε
Result: HE friendly Multi-GAT (FHE-MG)
;
for $i \leftarrow 1$ *to nb_norm_layers* **do**
  Freeze MG weights;
  $MG \leftarrow$ MG + $Poly\_Norm_i(x, x\_scale\_tab_i, \varepsilon)$;
  $Loss_{norm} = MSE(Norm_i, Ploy\_Norm_i)$;
  Optimize the $Poly\_Norm_i$ layer with $Loss_{norm}$;
  $MG \leftarrow$ MG - $Norm_i(x)$;
  Fine-tune the layers in MG between $Norm_i$ and MG output
end
return *MG*;

Algorithm 3: Approximation workflow.

4 EXPERIMENTS

In this section, experiments are conducted to test the effectiveness of the proposed method on different documents datasets.

4.1 Datasets

We test our approach on three business documents databases: SROIE, generated Gen-Invoices-Fr and Gen-Invoices-En.

SROIE (Huang et al., 2019): is a database of real receipts. It is divided into 626 receipts for training and 437 for testing and contains four entities to extract.

Gen-Invoices-En and Gen-Invoices-Fr: two artificial invoices databases generated using a generic business documents generator (Belhadj et al., 2021) and (Blanchard et al., 2019). Gen-Invoices-En and Gen-Invoices-Fr contain 1500 invoices in English and French language respectively. They are splitted into: 1000 documents for training, 200 for validation and 300 for test. Each dataset provides 28 classes to predict (including the undefined class).

4.2 Implementation Details

The proposed model has been implemented using Tensorflow and Keras frameworks. We reuse the Multi-GAT implemented in (Belhadj et al., 2023b). The learning rate is set to 0.001 during the PHE-MG baseline training and the normalization approximation and set to 0.00001 for the fine-tuning steps. The maximum number of epochs is set to 2000 to train the PHE-MG and to 100 for the normalization approximation and the fine-tuning. We use Adam optimizer and set n-heads to 8 in all the GAT layers for SROIE and to 26 for the Invoices datasets.

4.3 Tests and Results

We perform several experiments to choose the different polynomial approximations of the activation functions as well as the normalization layers.

4.3.1 ReLU and LeakyReLU Polynomial Approximation

In this experiment, we variate the ReLU and LeakyReLU polynomial approximation degrees as well as their coefficients nature (fixed or learnable) as can be seen in the Table 2.

Table 2: F1 score % obtained on the two datasets (Invoices-En and SROIE) by varying the ReLU and LeakyReLU polynomial approximations (Fx refers to fixed polynomial coefficients and Lr to learned coeficients.

	Function	Degree				
		2	3	4	5	6
Invoices-En	ReLU_Fx	99.15	99.05	**99.31**	99.28	99.04
	ReLU_Lr	99.14	**99.16**	99.14	99.14	99.06
	LeakyReLU_Fx	99.09	98.98	**99.14**	98.78	90.38
	LeakyReLU_Lr	**99.10**	82.51	71.24	67.71	42.08
SROIE	ReLU_Fx	98.21	98.30	**98.36**	98.20	98.21
	ReLU_Lr	**98.37**	98.31	98.13	98.05	97.97
	LeakyReLU_Fx	98.41	98.30	**98.43**	98.42	98.11
	LeakyReLU_Lr	97.70	**97.74**	93.08	92.18	92.93

As shown in Table 2, the best approximations of the ReLU and LeakyReLU for both datasets are obtained by the fixed coefficient polynomial approximations of degree 4.

4.3.2 Softmax Polynomial Approximation

We compare the different Softmax approximations proposed in the literature on our PHE-MG as shown in Table 3.

Table 3: F1 score % obtained by varying the Softmax polynomial approximation on the Invoices-En ans SROIE datasets.

Softmax approximation	Invoices-En	SROIE
(Ali et al., 2022)	99.20	98.23
(Al Badawi et al., 2020)	98.94	98.30
(Dathathri et al., 2019)	**99.33**	**98.30**
(Jang et al., 2022)	98.77	**98.35**

As can be seen in Table 3, the approximation that best fits the Invoices_En dataset is the one proposed in (Dathathri et al., 2019) which is a 2nd degree polynomial with learnable coefficients. For SROIE, the learnable coefficients polynomial of degree 5 proposed in (Jang et al., 2022) gives the best results but the result is very close to the 2nd degree polynomial (Dathathri et al., 2019). Therefore, we will keep the latter as it requires fewer resources and parameters while giving results as good as the 5th degree polynomial. For more stability during the learning process, we choose to initialize the second-degree polynomial with the polynomial coefficients proposed in (Al Badawi et al., 2020).

4.3.3 Normalization Layer Approximation

We evaluate the effect of each replaced normalization layer with the additional fine-tuning of the model as described in the algorithm 1. We first highlight the effect of each replacement of a normalization layer as shown in Table 4.

Table 4: F1 score % obtained on the three datasets by replacing the three normalization layers progressively.

Dataset	PHE-MG	Numbers of approximated layers		
		1	1, 2	1, 2, 3
Invoices-En	99.25	99.35	98.97	68.16
Invoices-Fr	98.14	98.14	98.10	63.04
SROIE	98.42	89.78	89.58	70.32

As we can see in the Table 4, replacing the first two normalization layers does not cause the model to lose much performance (nearly 1% for the Invoices datasets and 9% for SROIE), however replacing the three layers together significantly degrades the performance.

To guarantee a fully homomorphic model without significant performance loss, we present two distinct strategies. The first strategy is to delegate the last normalization layer to the client, which computes it and sends the results to the server. The second strategy is to reduce the number of GAT layers to three, with two normalization layers.

Table 5: F1 score % on the three datasets by replacing the three normalization layers progressively in a 3 GAT layers PHE-MG.

Dataset	PHE-MG	Numbers of approximated layers	
		1	1 and 2
Invoices-En	97.55	97.25	95.96
Invoices-Fr	98.32	98.33	95.38
SROIE	98.30	94.35	93.41

The second strategy gives the results shown in the Table 5. As could be noticed, we get less performance loss by reducing the number of layers in the Multi-GAT (around 3% loss for Invoices and 5% for SROIE). So depending on what we prioritise in the model, i.e. doing all the calculation on the server or keeping the most performance, we can choose between the two strategies.

4.3.4 Overall Results

We compare the results and complexity of our FHE model with different state-of-the-art information extraction systems on the SROIE dataset.

Table 6: F1 score and complexity comparison between our FHE model and other state-of-the-art systems. M refers to million, B to the Base model and L to the large model.

System	Params	F1
BERT(B) (Devlin et al., 2019)	340M	92.00
LayoutLM (B) (Xu et al., 2020a)	113M	94.38
LayoutLM (L) (Xu et al., 2020a)	343M	95.24
LayoutLMV2 (B) (Xu et al., 2020b)	200M	96.25
LAMBERT (Garncarek et al., 2021)	125M	98.17
FHE-MG (ours)	42M	93.41

As shown in Table 6, our proposed FHE model achieves fairly good performance compared to the other systems, and it is much less complex. Unlike the other systems, our model is adapted to the homomorphic encryption. It is also purely supervised, with no pre-training step, contrary to the other transformer-based systems.

5 CONCLUSIONS

In this paper, we presented a fully homomorphic model for extracting information from business documents utilizing a Multi-GAT graph nodes classifier. We proposed low-degree polynomial approximations for the three activation functions - ReLU, LeakyReLU, and the Softmax of the attention mechanism - in the Multi-GAT with no performance loss. We have also proposed a normalization layer approximation that does not rely on training data and can be fully computed on the server, avoiding any connection with the data owner. Our model suggests low degree approximations for all nonhomomorphic operations, effectively avoiding multiple connections with the data owner. Our system accurately classifies 28 entities from two Invoices datasets, with an interesting overall F1 score of 95%, and records a remarkable score of 93.41 on the SROIE dataset.

REFERENCES

Al Badawi, A., Hoang, L., Mun, C. F., Laine, K., and Aung, K. M. M. (2020). Privft: Private and fast text classification with homomorphic encryption. *IEEE Access*, 8:226544–226556.

Ali, H., Javed, R. T., Qayyum, A., AlGhadhban, A., Alazmi, M., Alzamil, A., Al-utaibi, K., and Qadir, J. (2022). Spam-das: Secure and privacy-aware misinformation detection as a service. *TechRxiv*.

Ali, R. E., So, J., and Avestimehr, A. S. (2020). On polynomial approximations for privacy-preserving and verifiable relu networks. *arXiv preprint arXiv:2011.05530*.

Baruch, M., Drucker, N., Greenberg, L., and Moshkowich, G. (2022). A methodology for training homomorphic encryption friendly neural networks. In *International Conference on Applied Cryptography and Network Security*, pages 536–553.

Belhadj, D., Belaïd, A., and Belaïd, Y. (2023a). Improving information extraction from semi-structured documents using attention based semi-variational graph auto-encoder. In *International Conference on Document Analysis and Recognition*, pages 113–129.

Belhadj, D., Belaïd, A., and Belaïd, Y. (2023b). Low-dimensionality information extraction model for semi-structured documents. In *International Conference on Computer Analysis of Images and Patterns*, pages 76–85.

Belhadj, D., Belaïd, Y., and Belaïd, A. (2021). Automatic generation of semi-structured documents. In *Document Analysis and Recognition–ICDAR 2021 Workshops: Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pages 191–205.

Blanchard, J., Belaïd, Y., and Belaïd, A. (2019). Automatic generation of a custom corpora for invoice analysis and recognition. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 7, pages 1–1. IEEE.

Chen, T., Bao, H., Huang, S., Dong, L., Jiao, B., Jiang, D., Zhou, H., Li, J., and Wei, F. (2022). The-x: Privacy-preserving transformer inference with homomorphic encryption. *arXiv preprint arXiv:2206.00216*.

Cheon, J. H., Kim, D., and Kim, D. (2020). Efficient homomorphic comparison methods with optimal complexity. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 221–256.

Chiang, J. (2022). On polynomial approximation of activation function. *arXiv preprint arXiv:2202.00004*.

Chiang, J. (2023). Privacy-preserving cnn training with transfer learning. *arXiv preprint arXiv:2304.03807*.

Dathathri, R., Saarikivi, O., Chen, H., Laine, K., Lauter, K., Maleki, S., Musuvathi, M., and Mytkowicz, T. (2019). Chet: an optimizing compiler for fully-homomorphic neural-network inferencing. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pages 142–156.

Davis, P. J. (1975). *Interpolation and approximation*. Courier Corporation.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Garncarek, Ł., Powalski, R., Stanisławek, T., Topolski, B., Halama, P., Turski, M., and Graliński, F. (2021). Lambert: layout-aware language modeling for information extraction. In *International Conference on Document Analysis and Recognition*, pages 532–547.

Ghodsi, Z., Gu, T., and Garg, S. (2017). Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. *Advances in Neural Information Processing Systems*, 30.

Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. (2016). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pages 201–210. PMLR.

Heinzerling, B. and Strube, M. (2018). Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*.

Hesamifard, E., Takabi, H., and Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*.

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., and Jawahar, C. (2019). Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE.

Ibarrondo, A. and Önen, M. (2018). Fhe-compatible batch normalization for privacy preserving deep learning. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2018 International Workshops, DPM 2018 and CBT 2018, Barcelona, Spain, September 6-7, 2018, Proceedings 13*, pages 389–404.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Ishiyama, T., Suzuki, T., and Yamana, H. (2020). Highly accurate cnn inference using approximate activation functions over homomorphic encryption. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3989–3995. IEEE.

Jang, J., Lee, Y., Kim, A., Na, B., Yhee, D., Lee, B., Cheon, J. H., and Yoon, S. (2022). Privacy-preserving deep sequential model with matrix homomorphic encryption. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 377–391.

Kim, A., Song, Y., Kim, M., Lee, K., and Cheon, J. H. (2018). Logistic regression model training based on the approximate homomorphic encryption. *BMC medical genomics*, 11(4):23–31.

Lee, J.-W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., Lee, E., Lee, J., Yoo, D., Kim, Y.-S., et al. (2022). Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054.

Li, D., Shao, R., Wang, H., Guo, H., Xing, E. P., and Zhang, H. (2022). Mpcformer: fast, performant and private transformer inference with mpc. *arXiv preprint arXiv:2211.01452*.

Liao, Z., Luo, J., Gao, W., Zhang, Y., and Zhang, W. (2019). Homomorphic cnn for privacy preserving learning on encrypted sensor data. In *2019 Chinese Automation Congress (CAC)*, pages 5593–5598. IEEE.

Liu, J., Juuti, M., Lu, Y., and Asokan, N. (2017). Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 619–631.

Lloret-Talavera, G., Jorda, M., Servat, H., Boemer, F., Chauhan, C., Tomishima, S., Shah, N. N., and Pena, A. J. (2021). Enabling homomorphically encrypted inference for large dnn models. *IEEE Transactions on Computers*, 71(5):1145–1155.

Lou, Q. and Jiang, L. (2021). Hemet: A homomorphic-encryption-friendly privacy-preserving mobile neural network architecture. In *International conference on machine learning*, pages 7102–7110. PMLR.

Meinardus, G. (2012). *Approximation of functions: Theory and numerical methods*, volume 13. Springer Science & Business Media.

Mohassel, P. and Zhang, Y. (2017). Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE.

Panda, S. (2022). Polynomial approximation of inverse sqrt function for fhe. In *International Symposium on Cyber Security, Cryptology, and Machine Learning*, pages 366–376.

Qian, J., Zhang, P., Zhu, H., Liu, M., Wang, J., and Ma, X. (2023). Lhdnn: Maintaining high precision and low latency inference of deep neural networks on encrypted data. *Applied Sciences*, 13(8):4815.

Wang, C.-C., Tu, C.-H., Kao, M.-C., and Hung, S.-H. (2022). Tensorhe: a homomorphic encryption transformer for privacy-preserving deep learning. In *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, pages 124–130.

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M. (2020a). Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., et al. (2020b). Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Zheng, P., Cai, Z., Zeng, H., and Huang, J. (2022). Keyword spotting in the homomorphic encrypted domain using deep complex-valued cnn. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1474–1483.