Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Secure CNN Computation Using Random Projection and Support Vector Regression

Alaa Mahmoud Ibrahim[1,*], Mohamed Waleed Fakhr[2], Mohamed Farouk[3]

[1] College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport (AAST), Heliopolis, Cairo, Egypt
[2] Department of Computer Engineering, Arab Academy for Science, Technology and Maritime Transport (AAST), Heliopolis, Cairo, Egypt
[3] College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport (AAST), Abu Qir, Alexandria, Egypt

ARTICLE INFO

ABSTRACT

Convolutional Neural Networks (CNNs) are foundational in numerous machine learning applications, particularly in image processing, where they excel in identifying patterns within visual data. At the core of CNNs lies the 2D convolution operation, which is essential for extracting spatial features from images. However, when applied to sensitive data, such as in medical imaging or surveillance, preserving the privacy of both the input data and the convolutional filters is crucial. This paper introduces a novel approach to secure the 2D convolution operation in CNNs, leveraging random projection and machine learning techniques. By encrypting the input images and convolutional filters using random projection, the method ensures that the convolution feature maps are computed securely without exposing the underlying data. The proposed technique maintains the accuracy and efficiency of CNN while offering a privacy-preserving solution that is more computationally efficient than traditional methods such as Homomorphic Encryption (HE). Experimental results using synthetic Gaussian data demonstrate the feasibility and effectiveness of this approach in securely computing convolutions, making it a promising solution for protecting sensitive information in CNN-based applications. Additionally, the paper compares the proposed method with homomorphic encryption, showing that while both methods ensure data confidentiality, the random projection approach offers a more efficient solution with lower computational overhead.

Keywords:
Privacy Preservation; Random Projection; Private Computing, Support Vector Machine for Regression

1. Introduction

Privacy preservation in machine learning has become a critical concern due to the increasing reliance on data-driven algorithms and the sensitive nature of the data used [1,2]. Various techniques have been developed to ensure that privacy is maintained throughout the machine learning process, addressing challenges from data collection to model deployment and inference. There are multiple techniques used solely for privacy preserving computing, these techniques are now integrated with

[*] *Corresponding author.*
E-mail address: alaa.mahmoud@aast.edu

https://doi.org/10.37934/araset.65.1.209225

machine learning to reach maximum benefits of both worlds. HE allows computations to be done on encrypted data without the need for decryption, while it has different implementations, when executed fully it becomes the most powerful algorithm that ensures maximum data confidentiality, however, it is considered to be computationally intensive [3]. Secure Multi-Party Computation (SMPC) enables multiple parties to compute a function collaboratively over their individual inputs while keeping those inputs private [3, 4]. Differential Privacy (DP) is one of the most widely adopted techniques in privacy preserving machine learning. It provides a mathematical guarantee that the removal or addition of a single data point will not significantly affect the outcome of the analysis, thus protecting individual data points. This approach has been implemented in various machine learning algorithms, including deep learning models, to ensure that the information about individuals cannot be reverse engineered from the model outputs [4]. Federated learning is another innovative approach that enhances privacy by keeping data localized on user devices. Instead of sending raw data to a central server, federated learning trains machine learning models locally on devices and only aggregates the model updates. This method significantly reduces the risk of data breaches and preserves the privacy of the data owners. Federated learning has been particularly useful in applications involving sensitive data, such as healthcare and finance [3].

Despite these advancements, several challenges remain in the implementation of privacy-preserving techniques in machine learning. These include balancing privacy and utility, managing the increased computational overhead, and addressing new types of privacy attacks such as model inversion and membership inference attacks [5]. Ongoing research is focused on developing more efficient algorithms, improving the scalability of privacy-preserving techniques, and ensuring robust privacy guarantees in dynamic and adversarial environments.

Random projection is an effective privacy-preserving technique in machine learning that involves reducing or expanding the dimensionality of data while maintaining its essential structure and properties. This method is usually used to project high-dimensional data onto a lower-dimensional subspace using a random matrix, effectively obscuring the original data's features. The random matrix is often generated using a distribution, such as Gaussian or sparse random matrices, which ensures that the distance between data points is approximately preserved. This approach not only helps in reducing computational complexity and storage requirements but also enhances privacy by preventing the reconstruction of the original data from the projected data. Random projection is particularly useful in scenarios where data anonymization and privacy are critical, such as in healthcare and financial applications, where sensitive information must be protected while still enabling meaningful data analysis [3,4,6].

This research aims to exploit the importance of secure computation on encrypted CNN, by focusing on the 2D convolution estimation using random projection and Support Vector Machine Regression, the proposed work encrypts the image and the filter values using random projection, and the SVM model calculates the real convolution feature map, without decrypting neither the image nor the filter values.

Paper organization is as follows: section 2 showcases review of literature, section 3 explains the presented model methodology, section 4 presents the experimental results, section 5 is the discussion of the results and finally section 6 summarizes the work done and where the future of the study is headed.

2. Literature Review

Recent studies, such as in [7] have highlighted the effectiveness of random projection in behavioural authentication systems. They demonstrated that by applying random projection, the system can maintain a high level of privacy while still achieving low false rejection and false acceptance rates. Additionally, the use of sparse random projection has been shown to significantly reduce computational loads, making it suitable for low performance computing devices. Further research by Miyaji et al. explored the broader applications of privacy-preserving techniques, including random projection, in various machine learning contexts. They underscored the method's robustness against privacy attacks, such as membership inference attacks, thereby reinforcing its importance in safeguarding sensitive data in practical applications [8]. These findings underline the practical utility of random projection in enhancing privacy without compromising the efficiency and accuracy of machine learning models.

In the study in [9], a privacy-preserving federated learning algorithm using CNNs and homomorphic encryption was proposed. This technique was particularly applied to medical data, demonstrating that it can effectively protect deep learning models from adversaries while maintaining the confidentiality of sensitive medical information. The primary strength of this method lies in its robust security model, which prevents data leakage during model training and inference. However, the major drawback is the significant computational overhead, which can result in slower processing times and increased resource requirements. Another notable work,[10], addresses the limitations of earlier models by implementing the ResNet-20 model with the RNS-CKKS HE scheme. This approach successfully approximates non-arithmetic functions like ReLU and Softmax, achieving high accuracy close to that of unencrypted models. The study demonstrates the feasibility of applying Fully Homomorphic Encryption (FHE) to advanced deep learning models, achieving 92.43% accuracy on the CIFAR-10 dataset. The strengths of this approach include high accuracy and strong security guarantees. However, the trade-off comes in the form of extensive computational time, with inference taking approximately three hours on high-performance computing infrastructure. The paper in used Homomorphic Encryption with Federated Learning [11] it explores this combined approach to train a CNN model, showcasing its application in a real-world medical scenario COVID-19. The study highlights that while Federated Learning preserves data privacy by not sharing raw data, the integration with Homomorphic Encryption ensures that even the intermediate computations remain secure. The combined approach provides robust security and maintains high accuracy, however, it also inherits the computational challenges associated with HE, such as increased processing time and resource consumption.

The paper in [12] presents a robust solution for privacy-preserving CNN feature extraction and retrieval in medical imaging, balancing the trade-offs between accuracy, efficiency, and privacy. proposes a novel scheme for feature extraction and image retrieval in the medical domain using CNNs. This approach leverages secret sharing techniques to split the image data between two non-colluding cloud servers, thereby protecting the privacy of the images. The primary contribution includes the development of three secure two-party protocols: secure mixed multiplication, secure compare, and secure mixed addition protocols. These protocols are deployed between the two cloud servers to perform CNN operations without compromising data privacy. The results of the proposed scheme demonstrate that the accuracy of image classification and retrieval in the encrypted domain is comparable to that of the original CNN operating in plaintext. This indicates the feasibility of the approach in maintaining high accuracy while ensuring privacy. Additionally, when compared with existing schemes such as Securing SIFT, MiniONN, and PPIS, the proposed scheme showed superior performance in terms of classification accuracy. It also demonstrated efficient image retrieval, successfully identifying and retrieving relevant images from the dataset. While the scheme shows promise, addressing communication overhead and ensuring security beyond the semi-honest model could further enhance its applicability.

The work in [13] focuses on improving the efficiency of CNN inference using FHE. The authors propose optimization techniques to make FHE-based CNN inference more practical for real-world applications. FHE is used to ensure that all CNN computations are performed on encrypted data, ensuring that data privacy is maintained throughout the inference process. The CNN architecture is optimized specifically to reduce the computational overhead when used in conjunction with FHE. The optimizations focus on reducing the size of encrypted data and the complexity of operations performed on the encrypted data. The paper presents significant improvements in the efficiency of privacy-preserving CNN inference, showing that the proposed optimizations can reduce the inference time and computational load. This makes FHE-based CNNs more feasible for practical applications. Authors in [14] focused on optimizing the use of HE for secure and efficient inference in Deep Neural Networks (DNNs). The authors aimed to balance the need for privacy with the demand for efficient and accurate classification, making HE more practical for real-world use in DNNs. HE is utilized to perform all DNN operations on encrypted data, ensuring that the data remains confidential throughout the inference process. This approach is critical for applications where data privacy is non-negotiable, such as in healthcare and finance. The DNN is optimized to work efficiently with HE by reducing the computational overhead associated with encrypted operations. The focus is on adapting the DNN architecture and processing steps to minimize the impact of HE on performance. The paper demonstrates that with the right optimizations, HE can be integrated into DNNs without severely compromising performance. The results show that it is possible to achieve secure and efficient inference, which is crucial for privacy-sensitive applications.

In [15] the paper proposes a new method for classifying encrypted network traffic by combining BERT (Bidirectional Encoder Representations from Transformers) and CNN. This hybrid approach, called BERT-Fused CNN (BFCN), leverages the strengths of both models to improve the accuracy of encrypted traffic classification, which is vital for network security. The method is designed to work directly on encrypted data, ensuring that the classification process does not compromise the privacy of the network traffic. BERT is used to understand the context within the encrypted traffic data, while CNN is employed to extract features from these contextual embeddings, making the classification process more accurate and efficient. The integration of BERT helps in capturing complex patterns in encrypted traffic that traditional methods might miss. The BFCN model shows significant improvements in the accuracy of encrypted traffic classification compared to traditional methods.

The work in [16] addresses one of the primary technical obstacles: the nonlinearity of CNN activation functions, such as ReLU, which makes them incompatible with traditional homomorphic encryption methods. To address these challenges, researchers have proposed various privacy-preserving techniques for CNN classification. One such solution is the Distributed Two Trapdoors Public-Key Cryptosystem (DT-PKC), which supports encrypted CNN operations without exposing sensitive data. This method provides a security protocol toolkit that enables secure multiplication, activation function computation, and average pooling. Furthermore, to approximate the ReLU function and enhance accuracy, a novel continuous and derivative-based Tanhplus function has been introduced. This function, combined with a homogenization algorithm, allows for precise computation of activation functions while operating under ciphertext. In addition to ensuring privacy, the DT-PKC-based approach supports lightweight users and multiple key management, making it suitable for real-world applications. Security analyses and performance evaluations have demonstrated that this scheme not only maintains privacy but also ensures high accuracy and efficiency in encrypted CNN classification.

3. Methodology

In the study presented in [17] the proposed model outlined a secure computation approach using SVMs. The system was expandable and showed promise in comparison to other private computing techniques used on a large scale, in terms of complexity, efficiency and expansion rate. The model proposed here is a derivative of that study, where it explores the potential of having secure 2D convolution computations, which is the major computing step in any CNN. This is accomplished by encrypting both the filter and the image using the random projection approach.

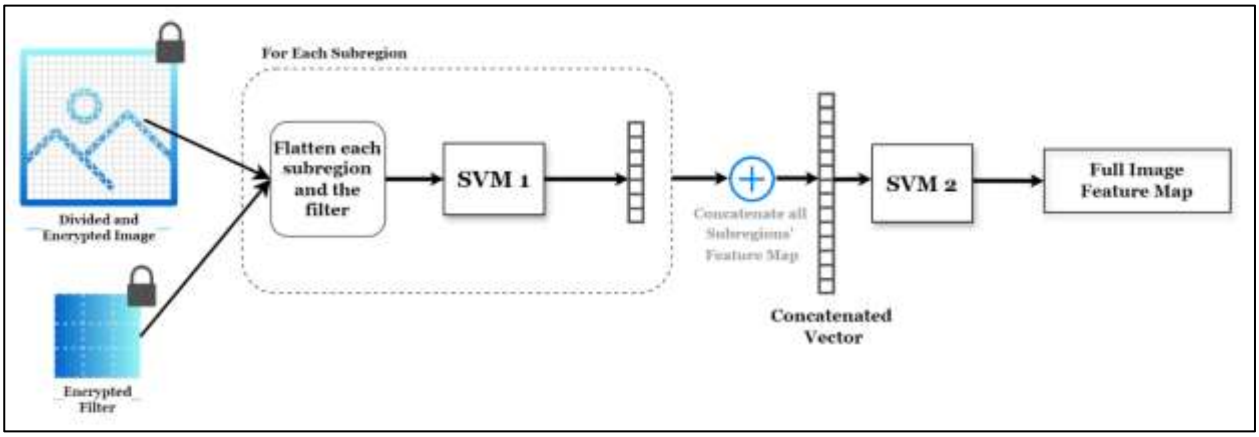
For Each Subregion
Divided and Encrypted Image
Encrypted Filter
Flatten each subregion and the filter
SVM 1
Concatenate all Subregions' Feature Map
Concatenated Vector
SVM 2
Full Image Feature Map

Fig. 1. Proposed Model: SVM1 calculates the convolution of a subregion and svm2 calculates the full convolution of the image

Fig. 1 demonstrates the main architecture of the proposed model; The image is divided into a number of non-overlapped subregions; each is encrypted using a different random projection matrix Φ_{img}. Each subregion and filter are input to SVM1 simultaneously, and the SVM1 is trained to produce the subregion-filter convolution $Conv(sub_{img_{i,j}})$. All the subregion results from SVM1 are then concatenated and inserted to SVM2 which is trained to calculate the full image's convolution in plaintext. Algorithm 1 explains the process in pseudocode. Table 1 contains the variables use in the algorithm.

Table 1
List of Variables

Variable	Meaning
img	Input image of size mxm
f	Filter with size pxp
ϕ_{img}	Encryption random projection matrix for the image.
ϕ_f	Encryption matrix for the filte
n	Size of each image's subregion
SVM_1	Pretrained SVM used to calculate each subregion convolution
SVM_2	Pretrained SVM uses the concatenated outputs of SVM_1 to calculate the full image's convolution and produces a feature map
$sub_{img_{i,j}}$	Subregion of an encrypted image
enc_f	Encrypted filter
enc_{img}	A fully encrypted image

Algorithm 1: Pseudocode of the Proposed Model

```
Encrypt Image: $enc_{img} = img * \phi_{img}$
Encrypt Filter: $enc_f = f * \phi_f$
Calculate number of subregions per dimension: $sub_{reg_{count}} = \frac{img_{size}}{n}$
Divide image into $subregions\ of\ size\ nxn$ :
for $k$ from 0 to $sub_{reg_{count}}$ step 1:
 for $i$ from 0 to $m - n$ step $n$:
  for $j$ from 0 to $m - n$ step $n$:
   $sub_{img_{k,i,j}} = extract_{subregion(enc_{img},i,j,n)}$
   $SVM_{1_{input_k}} = concatenate\left(sub_{img_{k,i,j}}, enc_f\right)$
Calculate subregion convolution:
for $i$ from 0 to $k$ step 1:
 $sub_{conv_i} = SVM_1\left(SVM_{1_{input_i}}\right)$
 $SVM_{2_{input}} = concatenate\left(sub_{conv_i}, SVM_{2_{input}}\right)$
Calculate full image convolution:
$full_{conv} = SVM_2\left(SVM_{2_{input}}\right)$
```

The size of the convolution output assuming the size of the image $m\ x\ m$ and the size of the filter $p\ x\ p$ is calculated as follows:

$$Convolution\ Size\ (ConvSize) = \frac{m - f}{1} + 1 \tag{1}$$

The stride in this case equals 1. To explain how the presented model works this is an example demonstrated with numbers. To calculate the convolution of a given $15\ x\ 15$ image, and a $3\ x\ 3$ filter, after the convolution the result size should be $= {(15 - 3)}/{1} + 1 = 12 + 1 = 13$. The convolution produces a $13\ x\ 13$ feature map which when flattened the output vector is going to be $169\ x\ 1$.

When using the proposed model, the target is to have the SVM-cascade acting as a convolution layer. The proposed approach uses two different SVMs for regression $SVM1$ & $SVM2$, to calculate both convolutions of the image's subregion and the image full image's one, the image and the filter both get encrypted using random projection as a first step, then the image is divided into subregions, each region size is a $5\ x\ 5$ sub-image, therefore, there are 9 subregions in this case. The stride is assumed to be 1 in this model and any other stride value would require retraining the SVM model.

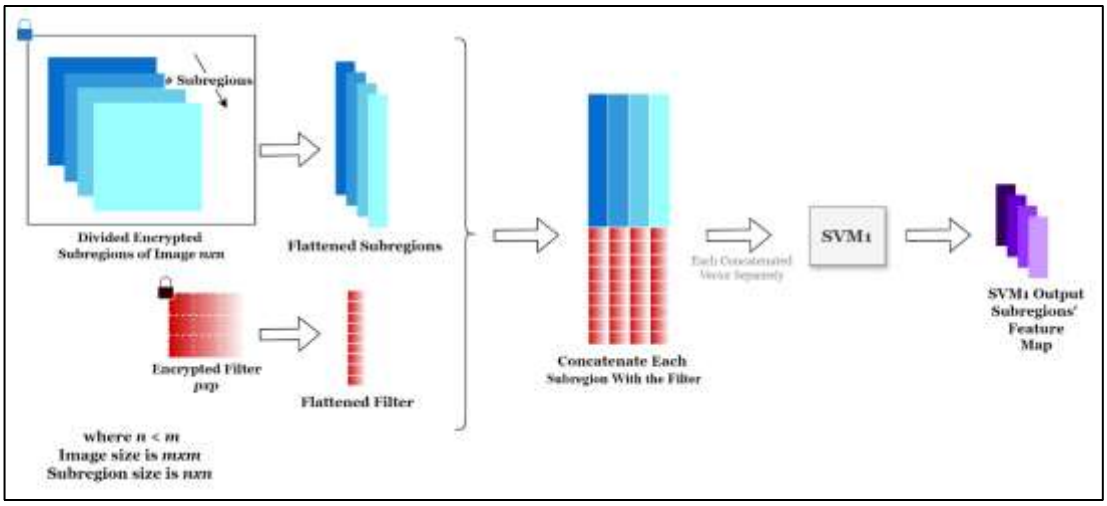

Fig. 2. SVM1 Input and Output

Each image is encrypted by multiplying it by a random matrix Φ_{img} the filter also gets encrypted using another random projection key Φ_f, the image is divided into subregions each of which is flattened. The filter is treated the same way as well then both flattened subregion and filter get concatenated together. The concatenated vector represents SVM1 input and used to produce the convolution between a subregion and the filter. The output size is a $9\ x\ 1$ vector for each inserted subregion. If there exist 9 subregions the output size of SVM1 is $81\ x1$ Fig. 2 explains this process.

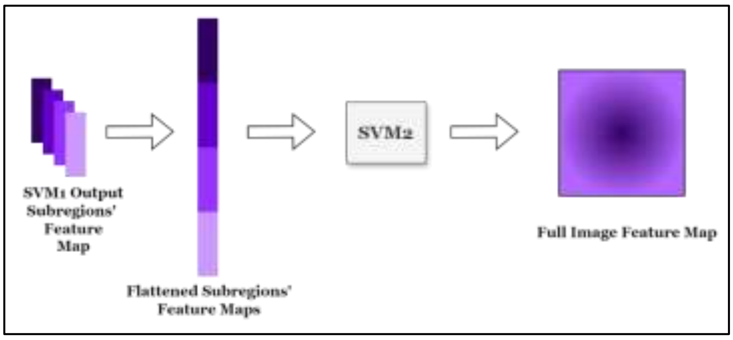
SVM1 Output Subregions' Feature Map
Flattened Subregions' Feature Maps
SVM2
Full Image Feature Map

Fig. 3. SVM2 Input and Output

Fig. 3 showcases that the second SVM purpose is to read the flattened output vector from $SVM1$ then produces the full true convolution size which is $169\ x\ 1$.

4. Security Analysis of Proposed approach

Our metric of security is how much data required by the attacker to break the system. The previous paper [17] showed experimentally that increasing the number of random matrices *K*, has the major effect of increasing the complexity of breaking the system. The following section will summarize the mathematical verification of this finding.

This section delves into the mathematical foundations of Cipher only attack where the attacker has only knowledge of the encrypted vector and attempts to break the system. We aim to identify weaknesses and highlight the parameters crucial for enhancing security. This analysis not only underscores the complexity of breaking the system but also provides insights into configuring defenses to balance efficiency and privacy [18,19].

4.1 Ciphertext Only Attack

4.1.1 System Architecture

The system operates by projecting sensitive data vectors into a higher-dimensional space using random matrices. These projections enable secure computations while masking the original data. The key components include the following:

Table 2
Table of Parameters for the Security Analysis

Variable	Definition
V	Original Vector
K	Number of Matrices
L	Original Dimension of Vector
D	Projected Dimension
R	Encrypted Vector V
n	Observations Required to Break the System

1. Input Data: Original vectors $V(i) \in R^D$ containing sensitive information.
2. Random Projection Matrices:
   K random matrices $P(k) \in R^{L\times D}$. Each projection matrix is a randomly generated Gaussian and independent.
3. Random Projection:
   - Each vector $V(i)$ is projected as $R(i) = P(k) \cdot V(i)$, where $R(i) \in R^L$.
   - The matrix $P(k)$ is selected randomly from K matrices for each vector.
4. Cloud Processing:
   - The cloud receives pairs of projected vectors $\big(R(i), R(j)\big)$.
   - Using an SVM-trained model, the system computes the dot product $V(i) \cdot V(j)$ without revealing $V(i)\ or\ V(j)$.

The original vectors and projection matrices remain hidden. Only the projected vectors $R(i)$ and $R(j)$ are exposed to the cloud, preventing direct access to the sensitive data.

4.1.2 Attacker Capabilities

An attacker has the following capabilities:

- Observations: Full access to projected vectors $R(i)$.
- Limitations:
  - No access to original vectors $V(i)$.
  - No knowledge of which matrix $P(k)$ was used for each $R(i)$.
  - No access to SVM model outputs (dot products).

Attacker Goal: Recover the projection matrices $P(k)$, enabling the attacker to reconstruct the original vectors $V(i)$.

4.1.3 Attack Complexity

The attacker’s challenge is to solve the following system:

$$R(i) = P(k) \cdot V(i), \quad \forall i \in \{1, \dots, n\} \tag{2}$$

4.1.4 Mathematical Analysis

4.1.4.1 Combined Projection Matrix Representation

To simplify the attack model, all K projection matrices are concatenated into a single matrix:

$$P' = \begin{bmatrix} P(1) \\ P(2) \\ \vdots \\ P(K) \end{bmatrix} \in R\ total\ size\ is\ K \cdot L \times D \tag{3}$$

4.1.4.2 System Equations

Each observation corresponds to:

$$R(i) = P_{slice} \cdot V(i) \tag{4}$$

where P_{slice} represents a subset of rows of P', determined by the matrix $P(k)$ used.

The total number of equations is $n \cdot L$, while the number of unknowns is:

$$K \cdot L \cdot D + n \cdot D \tag{5}$$

4.1.4.3 Minimum Observations

For the system to be solvable:

$$n \cdot L \geq K \cdot L \cdot D + n \cdot D \tag{6}$$

Rearranging for n:

$$n \geq \frac{K \cdot L \cdot D}{L - D} \tag{7}$$

4.1.5 Complexity Analysis with Different Parameters

Table 3
Different samples of observations of the cipher only attack

Case	K Metricies	Original Dimension D	Projected Dimension L	$L - D$	Min Observations ${(K.L.D)}/{(L-D)}$
1	5	32	36	4	1,440
2	10	32	36	4	2,880
3	10	64	72	8	5,760
4	10	32	48	16	960
5	15	128	144	16	17,280
6	8	256	288	32	18,432

It is clear that the security of the system can be enhanced by carefully selecting the number of projection matrices (K) and the difference between the plaintext and encrypted vector dimensions $(L - D)$. As $L - D$ is decreased, the system becomes more difficult to break. If $L - D$ becomes negative the system enters a compressed sensing scenario, however, the decryption may be very difficult and requires the sparsity of the input vector. As shown in Table 3 increasing the number of projection matrices linearly increases the minimum number of observations required, improving security. A larger gap $(L - D)$ exponentially improves security by increasing randomness. Configurations with higher K and $L - D$ values offer better security but incur greater computational and storage costs.

5. Experiments and Results

The training in all experiments and the data is generated using synthetic randomized Gaussian data, these data represent the image, the filter, the encryption keys as well as testing data. All experiments were conducted on a twelve core AMD Ryzen 9 5900X, and a 32 GB RAM.

5.1 Matrix Generation Process and Assumptions:

5.1.1 Matrix Generation Process:

- Encryption Matrices: The encryption and random projection matrices are assumed to be generated using cryptographically secure pseudorandom number generators (CSPRNGs). This ensures that the matrices are statistically independent and uniformly distributed, providing the randomness required for security.
- Randomness Assurance: Each matrix is generated for a specific session or query to prevent reuse, minimizing the risk of replay attacks.
- Dimensionality Constraints: The dimensions of the matrices ($L\ and\ D$) are chosen such that $L > D$ for enhanced obfuscation while ensuring efficient computation.

5.1.2 Security Assumptions:

- Independence: It is assumed that projection matrices are independent of each other and not shared across operations.
- Secrecy: Projection matrices are known only to the data owner or the trusted system components and not exposed to adversaries or untrusted cloud servers.
- Non-invertibility: The projection matrices are designed to obscure the original data sufficiently, making reverse-engineering infeasible under normal circumstances.

5.2 Experiment 1: Encrypted Image and Filter Convolution Prediction

5.2.1 Training the Support Vector Machine Regression Models:

This experiment required training two Support Vector Machines for Regression; each one is trained using synthetic randomly generated Gaussian images' features. Also, encryption keys are randomly generated as set to choose randomly from at the time of training. The first SVM1 is trained using 4000 examples. Each example resembles a $(5x5)$ image, as for the filter size is $3x3$, the testing was done on 2000 examples. The second SVM2 uses 10000 examples for training. The image size is a $(25x25)$, and the filter size is same as it is in SVM1, the testing length was 2000 examples as well in SVM2.

5.2.2 Convolution Prediction

In this section, the focus is on predicting convolutions of encrypted data using SVMs. The objective is to perform convolution operations on encrypted inputs, maintaining privacy while ensuring accurate computations.

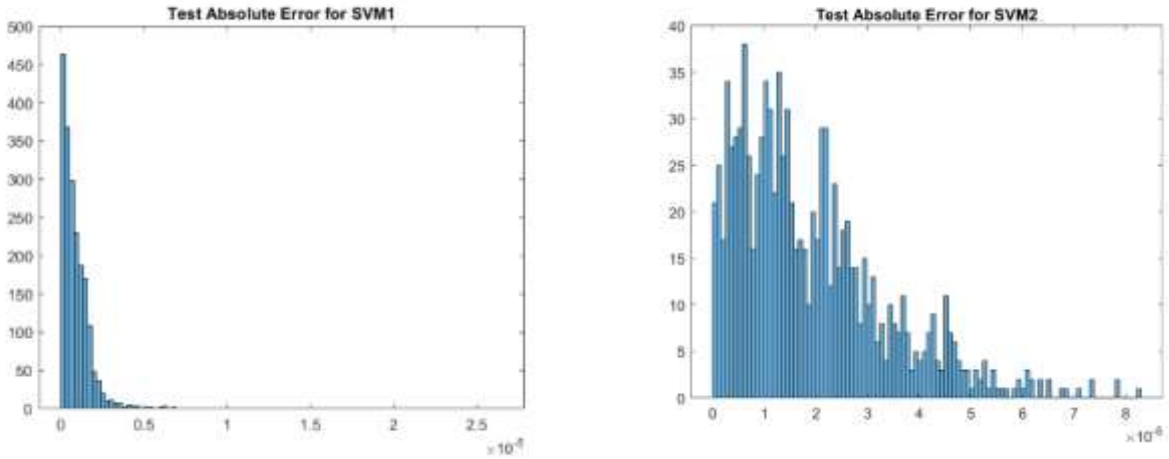

Fig. 4. Absolute Error for 2000 Test Examples for SVM1 & SVM2

- SVM1 Prediction and Results:

The first Support Vector Machine SVM1 is tasked with calculating the convolution of a small, encrypted image $(5x5)$ using an encrypted $(3x3)$ filter. Both the image and filter are encrypted with randomly projected keys to ensure privacy. This initial convolution operation forms the foundation for predicting larger convolutions, as it handles the basic convolution operation within a secure framework.

SVM1 successfully estimated the encrypted convolution for the $5x5$ image with a $3x3$ filter, yielding highly accurate results. Across 2,000 test cases, the average mean absolute error per dimension was exceptionally low, calculated at 8.4285e-05. This demonstrates that SVM1 can effectively handle encrypted convolutions with minimal error, ensuring that the predicted convolution remains close to the true convolution values, even when working with encrypted data.

- SVM2 Prediction and Results:

Building on the success of SVM1, the second Support Vector Machine (SVM2) is designed to handle the full convolution of a larger, encrypted image (25x25 pixels). The image is divided into subregions, and each subregion is processed using SVM1. The outputs from SVM1 are then aggregated by SVM2 to produce a final convolution result that matches the size and structure of a traditional, fully calculated convolution. This hierarchical approach allows SVM2 to scale the secure convolution process to larger images while still preserving the accuracy of the computations.

SVM2 was evaluated using two distinct test scenarios. In the first scenario, where the input data consisted solely of 2000 positive values, the mean absolute error was remarkably low at $1.1081e - 06$, indicating high accuracy in the predicted convolutions. In the second scenario, which included a broader range of both positive and negative input values with the same assigned testing length, the mean absolute error slightly increased to $1.9070e - 05$. Despite this increase, the error remained within acceptable limits, demonstrating that SVM2 effectively generalizes to a wider range of input data while maintaining the integrity of the encrypted convolution process. Fig. 4 presents the absolute error histogram for test cases of SVM1 and SVM2.

5.3 Experiment 2: Convolution Calculation Using Homomorphic Encryption

This experiment was conducted for comparison purposes. In this experiment, homomorphic encryption is utilized to compute the convolution of encrypted data. The goal was to evaluate the convolution of a 5x5 encrypted image using an encrypted 3x3 filter, mirroring the hyperparameters used in SVM1. Both the image and the filter were encrypted to ensure privacy during the computation. This experiment was conducted on 2,000 examples, consistent with the test cases used in the SVM1 experiment.

The computation was carried out using the TenSEAL python library [20], which implements homomorphic encryption techniques designed for secure and private machine learning operations. By leveraging homomorphic encryption, the convolution was computed directly on encrypted data without the need for decryption at any stage of the process. In the experiment Cheon-Kim-Kim-Song (CKKS) scheme is used, with polynomial modulus degree that equals 8192, and with the coefficient sizes (60, 40, 40, 60) bits.

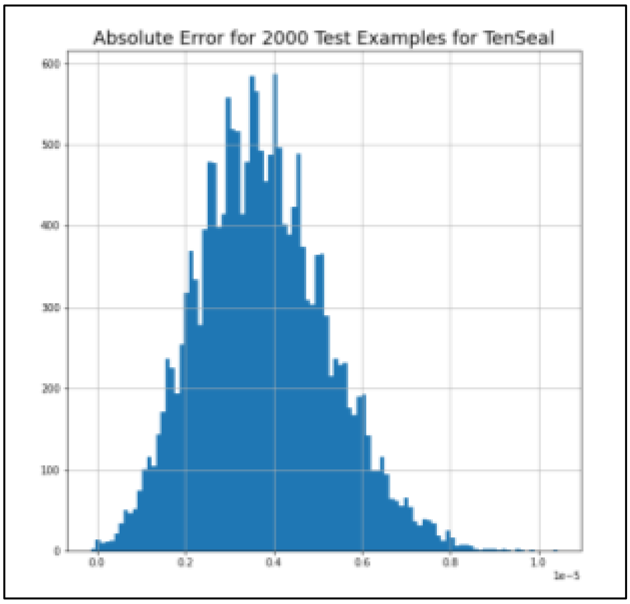

Fig. 5. Absolute error for homomorphic encryption convolution calculation using TenSEAL

The results of this experiment showed a mean absolute error of $3.5584e - 06$, demonstrating that homomorphic encryption can achieve highly accurate convolution results while preserving the privacy of the input data. Fig. 5 demonstrates the absolute error calculated on 2000 test examples.

5.4 Experiment 3: Number of Computations

5.4.1 Number of computations in a convolution layer:

Convolutional layers are the foundational building blocks of CNNs, where the bulk of computations occur. For an input and a convolution filter, the convolution operation involves sliding the filter over the image and computing the dot product at each location. This results in a feature map that encodes spatial information from the input image.

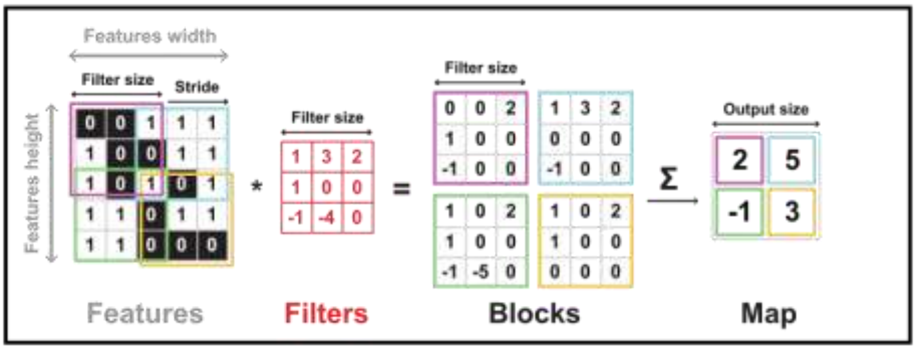

Fig. 6. A 2D Convolution Calculation Process with A 3x3 Filter and Stride = 1. Figure Courtesy [21]

In a standard convolution operation (without any encryption) Fig. 6, the total number of multiplications required is proportional to the product of the input image size and the filter size. Specifically, for a single convolutional filter applied to an image. In instance, the number of computations required to do convolution between a $3x3$ filter and a $200x200$ image:

i) Multiplications per position: 9 multiplications.
ii) Additions per position: 8 additions (after all multiplications).

Now, let's calculate the number of positions the filter can take over the image. Since the filter has to traverse the entire image, the number of positions can be computed by subtracting the filter size from the image size and adding 1 to account for the starting position. Accordingly:

Number of positions horizontally: $200 - 3 + 1 = 198$

- Number of positions vertically: $200 - 3 + 1 = 198$

Therefore, the total number of computations required for convolution is given by the following equation:

$$\begin{aligned} &Total\ computations\ = \\ &(Multiplications\ per\ position\ +\ Additions\ per\ position) * Number\ of\ positions^2 \\ &= (9\, multiplications\ +\ 8\, additions) * 198^2 \\ &= (17\, operations) * 39204 \\ &Total\ computations\ \approx\ 666{,}468\, operations \end{aligned} \tag{8}$$

Therefore, approximately $666{,}468$ computations are required to perform the convolution between the $3x3$ filter and the gray image of size $200x200$.

5.4.2 Number of computations in the proposed framework:

In traditional CNNs, convolution operations are computationally intensive, and this complexity escalates when applying privacy-preserving techniques, such as HE. In the presented framework, we mitigate this overhead by reducing the dimensionality of the input data through random projection before performing the convolution in the encrypted domain. This strategy minimizes the number of computations required, making it feasible to execute secure CNN operations without the prohibitive computational costs typically associated with encrypted computations. For a $200x200$ image and a $5x5$ subregion and a $3x3$ filter. The number of computations per image and one SVM, is the number of subregions which is $= 40$ multiplied by the number of features per subregion which is $= 40 *$

$25 = 1000$ computation per image in a single SVM, so for a single $200x200$ image and two SVMs there will be the equivalent of 2000 computations in total in the plaintext format. However, in the case of random projection, the feature vector might be expanded, and the result would be then multiplied by the expansion rate as well.

$$Num.Computations = \sum_{SVM=0}^{l-1} count(subregions) * count(features) * ProjectionRate \quad (9)$$

where l represents the number of available SVMs, the features count in the previous equation is per a subregion.

5.5 Experiment 4: CIFAR 10 Dataset

The experiment done previously on synthetic data along with this one achieved to test one neuron from the 1st convolution layer of a CNN. to basically have tested One neuron from the 1st convolution layer of a CNN). This experiment aimed to address how the training of the SVMs done on synthetic data, the experiment was done on a well-known dataset [22] CIFAR-10. The images went through a preprocessing phase first, the colored images were converted to greyscale, the values of the image were then normalized from 0 to 1. Filter values ranged from -3 to 3. And the image size was resized from $32\ by\ 32$ to $15\ by\ 15$. The images are then divided into regions using the same methodology used on the synthetic data. The Mean Absolute Error $= 3.4e^{-11}$ while the Mean Squared Error $= 1.8e^{-11}$.

6. Discussion

The results of presented experiments demonstrate the effectiveness of the proposed random projection-based method for securing 2D convolution operations within CNNs. By encrypting both the input images and convolutional filters, the approach successfully preserves the privacy of data without compromising the accuracy of the resulting feature maps. This outcome is significant, as it addresses a critical need in fields where CNNs process sensitive information, such as in medical imaging or financial analysis.

The proposed method offers several advantages over traditional privacy-preserving techniques, particularly HE and SMPC. While HE provides a prominent level of security, it does so at the cost of significant computational overhead, making it impractical for real-time applications or large-scale CNNs. In contrast, the random projection approach achieves similar levels of privacy protection with a fraction of the computational resources required by HE. This efficiency is particularly evident in our experiments, where the time required to compute secure convolutions using the presented method was considerably lower, without sacrificing accuracy.

Moreover, SMPC, though effective in a multi-party context, introduces complexities that can be avoided with this model. The random projection approach simplifies the process of securing convolutions, making it easier to implement and integrate into existing CNN frameworks, particularly in scenarios where computational efficiency is critical.

In experiments 1 & 2 it was demonstrated using the same experimental setup that a fully encrypted image and filter using random projection gives the same incredible performance done using homomorphic encryption, with much less space and complexity, which was studied in detail in the presented paper in [17] however, the homomorphic encryption library that was used were Microsoft SEAL [23] whereas in this paper the used library was TenSEAL Python Library [20]. These experiments revealed minimal errors in the computation of convolutions, both for individual subregions and for the entire image. These errors were within acceptable ranges and did not significantly impact the CNN's overall performance. A sensitivity analysis showed that the accuracy of the convolution results was robust across various configurations of the random projection matrices, suggesting that the method is resilient to changes in the projection parameters.

The third experiment compared the number of computations used to do a convolution in plaintext normal computation, and the presented model, the presented model surpasses the normal computation with much reduced number of computations, therefore, it also surpasses the needed number of computations used in homomorphic encryption. In practical terms, the reduction in the number of computations translates into faster processing times and lower resource consumption. For example, in our experiments, the number of operations required for secure convolution using random projection was reduced to approximately 10^4, compared to 10^6 operations required by traditional Homomorphic Encryption. This efficiency makes the presented method particularly suitable for real-time applications and scenarios where computational resources are constrained, such as mobile and edge computing environments.

The presented model inherits the security robustness of the study in [17], as well as the expansion, this study achieved these results using one SVM, but this could be expanded using many layers of SVMs and expanding the list of keys as well in size and count. Integration of the proposed method into existing CNN architectures is straightforward, as it does not require significant modifications to the network structure. This ease of integration, combined with the method's scalability, ensures that it can be applied across a wide range of CNN applications. However, the complexity of the CNN and the size of the dataset could introduce challenges in scalability, particularly as the number of convolutional layers increases. Future research should explore methods to optimize the scalability of this approach, potentially by parallelizing the random projection process or integrating it with other privacy-preserving techniques.

The current model is not trained to apply any activation function after computing the feature map, as the activation function process is non-linear in its nature, SVMs fail to learn the problem, which requires a kernel that can support non-linear computations along with linear ones, although an experiment was done on a number of activation functions (ReLU, SiLU, GELU) the error was at best within 10% of the real value. Moreover, increasing the training examples did not affect the quality of the prediction. To solve this problem, either compute the activation function separately, or use a different machine learning technique other than the SVM which will be able to predict the feature map and apply the activation function in one step, i.e. being able to predict a naturally linear and non-linear computation at the same time. However, changing the used machine learning technique could affect the time complexity.

Finally, in case of having larger image sizes, the model could be tuned to have multilevel SVMs that divides and calculates the convolution for different subregion sizes, each layer produces a calculation of a combined part of the subregion convolution. However, the two SVMs model should work on larger images as well. And should the image that is given have a smaller size than the trained one zero padding should be done first as a preprocessing step.

7. Conclusion and Future Work

This paper presents an enhancement in the domain of privacy-preserving Convolutional Neural Networks by focusing on securing the 2D convolution operation which is considered as a fundamental building block of CNNs. Given the increasing application of CNNs in sensitive areas using imaging, the need to protect the privacy of input data and convolutional filters has become critical. This study's approach, based on random projection and SVM, provides a secure method for computing 2D convolutions without revealing the underlying data. By encrypting both the input images and convolutional filters, the presented model ensures that the essential feature extraction process of CNNs remains private, thereby safeguarding sensitive information.

The method's efficiency, demonstrated through experiments with synthetic Gaussian data, shows that it preserves the accuracy of the CNN while significantly reducing computational overhead compared to traditional privacy-preserving techniques like HE. This makes the approach particularly suited for real-time and large-scale CNN applications, where computational resources are a concern.

The implications of this work extend beyond just the 2D convolution operation, offering a framework that can potentially be adapted to secure other aspects of CNNs. Future research will focus on optimizing the scalability of the proposed method, ensuring that it can oversee more complex networks and larger datasets. Additionally, exploring the integration of this technique with other privacy-preserving methods could further enhance its robustness and applicability across a wider range of machine learning tasks. This work marks a crucial step towards secure, privacy preserving CNNs, ensuring that the power of deep learning can be harnessed without compromising data confidentiality.

References

[1] Dasi, U., N. Singla, R. Balasubramanian, S. Benadikar, and R. R. Shanbhag. "Privacy-preserving machine learning techniques: balancing utility and data protection." *Int. J. Multidiscip. Innov. Res. Methodol* 3 (2024): 251-261.

[2] Nasr, Milad, Saeed Mahloujifar, Xinyu Tang, Prateek Mittal, and Amir Houmansadr. "Effectively using public data in privacy preserving machine learning." In *International Conference on Machine Learning*, pp. 25718-25732. PMLR, 2023.

[3] Nasr, Milad, Saeed Mahloujifar, Xinyu Tang, Prateek Mittal, and Amir Houmansadr. "Effectively using public data in privacy preserving machine learning." In *International Conference on Machine Learning*, pp. 25718-25732. PMLR, 2023.

[4] Xu, Runhua, Nathalie Baracaldo, and James Joshi. "Privacy-preserving machine learning: Methods, challenges and directions." *arXiv preprint arXiv:2108.04417* (2021).

[5] Rigaki, Maria, and Sebastian Garcia. "A survey of privacy attacks in machine learning." *ACM Computing Surveys* 56, no. 4 (2023): 1-34. https://doi.org/10.1145/3624010

[6] Tiwari, Kapil, Samiksha Shukla, and Jossy P. George. "A systematic review of challenges and techniques of privacy-preserving machine learning." *Data Science and Security: Proceedings of IDSCS 2021* (2021): 19-41. https://doi.org/10.1007/978-981-16-4486-3_3

[7] Islam, Md Morshedul, and Md Abdur Rafiq. "Privacy Preserving Machine Learning for Behavioral Authentication Systems." *arXiv preprint arXiv:2309.13046* (2023).

[8] He, Bingchang, and Atsuko Miyaji. "Balanced Privacy Budget Allocation for Privacy-Preserving Machine Learning." In *International Conference on Information Security*, pp. 42-56. Cham: Springer Nature Switzerland, 2023. https://doi.org/10.1007/978-3-031-49187-0_3

[9] Wibawa, Febrianti, Ferhat Ozgur Catak, Salih Sarp, and Murat Kuzlu. "BFV-based homomorphic encryption for privacy-preserving CNN models." *Cryptography* 6, no. 3 (2022): 34. https://doi.org/10.3390/cryptography6030034

[10] Lee, Joon-Woo, HyungChul Kang, Yongwoo Lee, Woosuk Choi, Jieun Eom, Maxim Deryabin, Eunsang Lee et al. "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network." *iEEE Access* 10 (2022): 30039-30054. https://doi.org/10.1109/ACCESS.2022.3159694

[11] Wibawa, Febrianti, Ferhat Ozgur Catak, Murat Kuzlu, Salih Sarp, and Umit Cali. "Homomorphic encryption and federated learning-based privacy-preserving cnn training: Covid-19 detection use-case." In *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, pp. 85-90. 2022. https://doi.org/10.1145/3528580.3532845

[12] Cai, Guopeng, Xiaochao Wei, and Yao Li. "Privacy-preserving CNN feature extraction and retrieval over medical images." *International Journal of Intelligent Systems* 37, no. 11 (2022): 9267-9289. https://doi.org/10.1002/int.22991

[13] Kim, Dongwoo, and Cyril Guyot. "Optimized privacy-preserving cnn inference with fully homomorphic encryption." *IEEE Transactions on Information Forensics and Security* 18 (2023): 2175-2187. https://doi.org/10.1109/TIFS.2023.3263631

[14] Akram, Aftab, Fawad Khan, Shahzaib Tahir, Asif Iqbal, Syed Aziz Shah, and Abdullah Baz. "Privacy preserving inference for deep neural networks: Optimizing homomorphic encryption for efficient and secure classification." *IEEE Access* 12 (2024): 15684-15695. https://doi.org/10.1109/ACCESS.2024.3357145

[15] Shi, Zhaolei, Nurbol Luktarhan, Yangyang Song, and Gaoqi Tian. "BFCN: A novel classification method of encrypted traffic based on BERT and CNN." *Electronics* 12, no. 3 (2023): 516. https://doi.org/10.3390/electronics12030516

[16] Wang, Baocang, Yange Chen, Furong Li, Jian Song, Rongxing Lu, Pu Duan, and Zhihong Tian. "Privacy-preserving convolutional neural network classification scheme with multiple keys." *IEEE Transactions on Services Computing* 17, no. 1 (2024): 322-335. https://doi.org/10.1109/TSC.2023.3349298

[17] Ibrahim, Alaa Mahmoud, Mohamed Farouk, Mohamed Waleed Fakhr. "Privacy Preserving Image Retrieval Using Multi-Key Random Projection Encryption and Machine Learning Decryption." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 2, (2024):155–174. https://doi.org/10.37934/araset.42.2.155174

[18] Liu, Kun, Hillol Kargupta, and Jessica Ryan. "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining." *IEEE Transactions on knowledge and Data Engineering* 18, no. 1 (2005): 92-106. https://doi.org/10.1109/TKDE.2006.14

[19] Strang, Gilbert. *Linear algebra and its applications*. 2000.

[20] Benaissa, Ayoub, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. "Tenseal: A library for encrypted tensor operations using homomorphic encryption." *arXiv preprint arXiv:2104.03152* (2021).

[21] Convolution 2D (CNN) — EpyNN 1.0 documentation. (n.d.)

[22] CIFAR-10 - Object Recognition in Images

[23] Chen, Hao, Kim Laine, and Rachel Player. "Simple encrypted arithmetic library-SEAL v2. 1." In *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*, pp. 3-18. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-70278-0_1