CAP 6938: Trustworthy Machine Learning
Fall 2024 - Final Project Report
Justin Morera & Sebastian Perez

Member Contributions:

Justin Morera	Sebastian Perez
Identified past model	Researched HE packages
Implemented HE	Implemented HE
Constructed models	Added metrics
Researched dataset	Created figures & graphs
Gathered results	Contributed to report
Contributed to report	

CAP 6938: Trustworthy Machine Learning
Fall 2024 - Final Project Report
Justin Morera & Sebastian Perez

Introduction

The goal of this project is to more uniformly measure and understand just how impactful the drawbacks of homomorphic encryption (HE) are on privacy-preserving machine learning (PPML) models. Multiple HE schemes have been proposed for PPML, but many have differing metrics showcasing how efficient these schemes are in conjunction with models; furthermore, most of these proposals do not utilize healthcare-specific datasets or apply their models in a meaningful way for the healthcare field. We attempt to address these shortcomings, at least in terms of one commonly used HE scheme, the Cheon-Kim-Kim-Song (CKKS) fully homomorphic encryption scheme, by training, testing, and comparing four similar ML models on both encrypted and unencrypted data. We also attempt to cover a wider range of metrics in order to generalize our findings and add more information to the literature. Our hope is that by using a uniform set of metrics, we increase the overlap between measurements of PPML models' effectiveness to advance our collective understanding of the advantages and disadvantages of HE when applied to healthcare scenarios.

This project is aimed toward setting an example for future research into proper comparisons between HE methods on PPML for healthcare purposes. The desired outcome for this project is to exhibit a clear difference between the metrics evaluated from a model trained and tested on encrypted MNIST data versus models with identical architecture trained and tested on unencrypted MNIST data, encrypted medical-based data, and unencrypted medical-based data. Key insights to obtain include understanding the impact on accuracy, recall, and precision when comparing encrypted versus unencrypted data, gauging additional runtime due to computational overhead on encrypting and decrypting inputs, as well as analyzing the difference in performance metrics between models using simplistic MNIST data versus more complex medical data.

Literature Review

Machine learning has become an essential and prominent practice in the healthcare field over the recent decade. Due to the increasing complexity of tasks and the advancing nature of healthcare research, practitioners have had to more heavily rely on machine learning models to resolve issues that have previously been too complex for general healthcare practices. Machine learning has allowed for unprecedented success in the diagnosis and treatment of countless conditions (Guerra-Manzanares et al., 2023), as well as the research and understanding of drug interactions and genomic studies. However, due to the sensitive nature of these types of use cases, it is often the case that such applications of machine learning involve direct manipulation of sensitive data, usually personal health information (PHI); because of the risks associated with the potential exposure of such information to unauthorized parties, PPML has come to the forefront of machine learning use in the healthcare field.

PPML involves using methods to obscure sensitive information from the machine learning model without significantly affecting its ability to accomplish its designated tasks. There are a number of privacy-preserving methods, but the one this project focuses on is HE. HE, a form of encryption that allows a number of specific operations (in our case, unlimited additions, with a limited number of multiplications) performed on encrypted data to be reflected in the corresponding decrypted data without disturbing the encryption, is useful because it allows healthcare providers to send encrypted data to untrusted PPML models to accomplish a task and receive an output that can be decrypted with the approximate operations applied. The issue is that since the encryption method involves approximated operations, there is a level of accuracy loss involved in the output of HE models. This issue is further exacerbated by the additional overhead caused by the complex encryption/decryption process involved in using such models.

Guerra-Manzanares et al. gathered a collection of recent articles presenting healthcare-oriented PPML research, listing the various strategies employed in these articles while also reviewing each study's analysis and areas of application in a comprehensive table in their 2023 paper "Privacy-preserving machine learning for healthcare: open challenges and future perspectives". The authors emphasize the lack of uniform analysis present in this wide range of PPML publications and stress that future research can be made more uniform and generalized by adhering to standardized metrics for easier comparability. They also note that the majority of the studies publish specifically for healthcare applications only manage to test their proposed techniques on benchmark datasets, usually the various MNIST datasets; this also poses a concern in that the data obtained from testing may not be representative of the real efficiency and computational overhead seen in a more realistic application. Regardless, a number of the papers cited within Guerra-Manzanares et al.'s own feature PPML models tasked with addressing similar functions as the own this paper seeks to address, including HE federated learning, and differential privacy, on image, text, and multimodal inputs.

"Privacy-Preserving Deep Learning With Homomorphic Encryption: An Introduction" by Falcetta et al. (2022) offers a brief overview of PPML utilizing HE and summarizes how the process works, as well as how to implement a basic example of a HE LeNet-1 model tested on MNIST and Fashion MNIST. While not oriented toward healthcare-related applications, this article served as our first starting point, as it was the most similar to other papers listed by Guerra-Manzanares et al., which did not have publicly available code. The model in Falcetta et al. paper uses the Brakerski–Fan–Vercauteren (BFV) scheme with their model, which ultimately proved to be too complex for our implementation, as well as the Pyfhel HE Python package, which is also deprecated and no longer works, forcing us to move to other implementations as a basis for our own. Despite these obstacles, the paper itself, and the general outline of the publicly available code, offers a great overview of what HE entails and how it can be used for important healthcare tasks.

OpenMind's 2021 TenSEAL GitHub repository features the core of our own implementation by presenting a Python package that enables a tensor-focused method of HE geared toward PPML. This package removed much of the strain of developing HE algorithms from scratch, and also offered a coded example of using the CKKS encryption scheme with a model on the MNIST dataset. The architecture from this model was used for our base model architecture, then extended to work with the medical dataset inputs used in this paper.

Methodology

The project consists of four basic convolutional neural networks (CNNs) of similar architecture. Two of the models are trained and tested solely on inputs from the MNIST dataset, with 60,000 training inputs, and 10,000 validation inputs. The other two models are tested and trained on inputs from a 253-image dataset of neuro MRIs, with 202 used for training and 51 used for validation. The images are RGB, but had to be converted to grayscale due to complications with the coded implementation. One of the two models trained on each dataset passes its trained weights to an encrypted version of the same model, allowing it to be tested on encrypted inputs. Additionally, the unencrypted versions of these two models receive altered training inputs that simulate quantization and noise accumulation due to encryption. MNIST models were trained over 10 epochs while MRI models were trained over 20, and a wide array of metrics were tracked for each: accuracy, recall, F1-score, precision, AUROC, and end-to-end runtime measurements (training time, encryption time, evaluation time, decryption time, total time).

Originally, the project was adapted from Guerra-Manzanares et al.'s publicly available code; however, after several unsuccessful attempts to get the essential Pyfhel package operational, new HE packages had to be researched. Numerous packages were attempted, but it became evident that most HE packages were not built for PPML application, particularly because of the complexity of working with tensors, and most PPML packages did not use fully homomorphic encryption. Eventually, TenSEAL was found and became the HE package used in our model. Additionally, the departure from Guerra-Manzanares et al.'s model meant we now no longer had access to pre-trained weights, and thus had to implement a training process for each model that incorporated simulated quantization and noise accumulation for the models training for encrypted data.

After initial results were obtained, it was revealed that the model was not operating on encrypted data, meaning the data for the two encrypted models was not representative of how the models would perform on properly encrypted inputs. The subsequent edits to the code led us to depart further from the original design, and the lack of support for the BFV encryption scheme caused us to shift our models from the LeNet implementation using BFV to OpenMind's example architecture of a simple CNN using the CKKS scheme. This scheme had much more support and was more easily applied to tensors.

The finalized architecture utilizes one convolution layer with four output channels, a seven-by-seven kernel, a stride of three, and no padding, two fully-connected layers, 64 hidden neurons, square activation functions, flattening before the fully-connected layers, and 10 outputs for MNIST models and 2 outputs for medical models. The medical models were tasked with the binary classification of neuro MRIs that either contained a brain tumor or did not. The encrypted models utilized the same layers as the unencrypted models, but accepted the pretrained weights and biases from their unencrypted counterparts that were trained on encryption-simulated data.

The architecture was chosen for its ease of implementation as a fallback from the difficulties of trying to use the BFV scheme with Pyfhel; the TenSEAL example architecture already accomplished the task of training and testing on MNIST data, as originally planned, and also incorporated passing trained weights and biases to an encrypted model. This made adding in encryption-simulating functions simple, and lessened the strain of altering the models to accept the MRI images instead of MNIST images.

We used an encrypted and decrypted version of two datasets. The first dataset used was MNIST which is broadly recognized as a benchmark dataset for evaluating image recognition. It is composed of 70,000 grayscale images each being handwritten digits including numbers from 0 up to 9. The data is broken up into 60,000 images used for training and 10,000 images used for testing and evaluating the performance of the model. There are ten classes in this dataset with a class for each digit 0 to 9. The second dataset used was a medical dataset covering brain MRI scans for detecting tumors. This dataset was significantly smaller with 253 images, broken up into 98 brain scans with no tumor and 155 brain scans with a tumor present. The dataset used 202 images for training and 51 images for testing and evaluating. There were two classes for “Yes tumor” and “No tumor”.

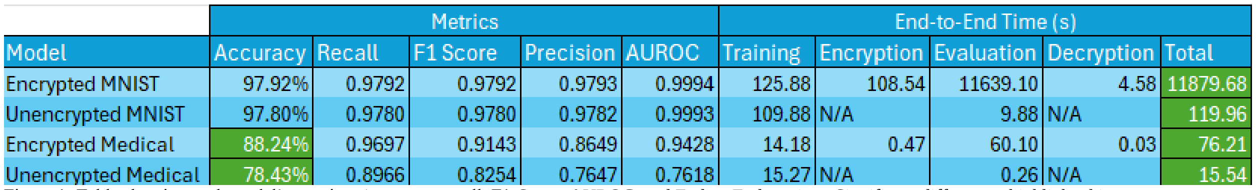
The project was completed using four .py files and imported Python packages. The model was developed using built-in Pytorch functions and the encryption was done using TenSEAL built-in functions in conjunction with Pytorch parameters and tensors. The datasets were remotely downloaded using Kaggle's API for the MRI images and Torchvision's Datasets submodule for the MNIST images. Due to complications with the models' architecture, the MRI images had to be converted from RGB to grayscale, which did not seem to significantly affect the models' performance, and simplified both the models and the encryption process. Metrics were calculated using ScikitLearn's built-in classification report, confusion matrix, F1-score, recall score, precision score, and AUROC submodules, which were then printed to the console as standard output. Results were manually copied over from the output console to an Excel spreadsheet where the corresponding graphs and tables were developed from.

As explained above, each model was trained on 80% of either the MRI or MNIST dataset over 20 epochs and 10 epochs respectively. Learning rates were kept at 0.001 for all models, and training batch size was 64 for the MNIST models while only 32 for the MRI models. All inputs were transformed 28x28 (not necessary for MNIST), converted to tensors, and the MRI images were normalized with a standard deviation of 0.5 and a mean of 0.5. For encrypted models, training inputs received a random selection of perturbations from a standardized set of values in an effort to simulate noise generated from encryption before being passed into the plaintext training model. Similarly, the training inputs were quantized by a factor of 1000 to reduce precision in a way resembling precision loss from encryption.

Results

The encrypted MNIST had an accuracy of 97.92%, recall score of .9792, F1 score of 0.9792, precision score of .9793, and an AUROC score of .9994. The unencrypted MNIST had an accuracy of 97.80%, recall score of .9780, F1 score of .9780, precision score of .9782, and an AUROC score of .9993. The encrypted medical had an accuracy of 88.24%, recall score of .9697, F1 score of 0.9143, precision score of .8649, and an AUROC score of .9428. The unencrypted medical had an accuracy of 78.43%, recall score of .8966, F1 score of .8254, precision score of .7647, and an AUROC score of .7618.

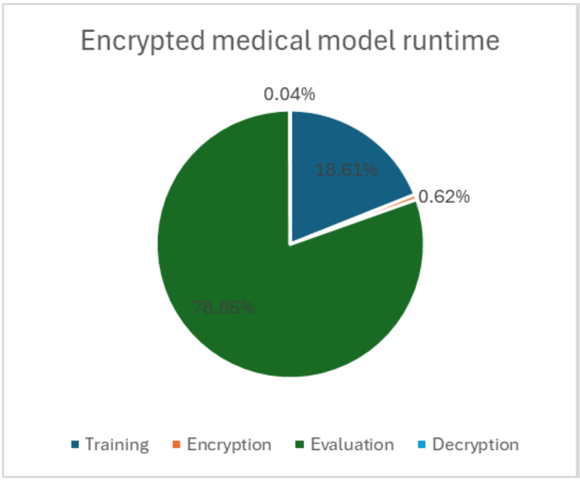
Our experiments found that models tested with MNIST data, regardless of encryption, were significantly more accurate than models tested with MRI images. Even though the architecture was the same, and the task was simpler for binary classification, the medical models still produced 78.43% accuracy for unencrypted and 88.24% accuracy for encrypted images while the MNIST models both produced just under 98% accuracy. It is also interesting to note that the encrypted medical model performed better than its unencrypted counterpart. For all models, recall, F1-score, precision, and AUROC all gave similar values to their respective model's accuracy result, reducing the amount of discussion to be made about those metrics. Figure 1 presents the experimental results in a succinct table, with significant differences highlighted in green.

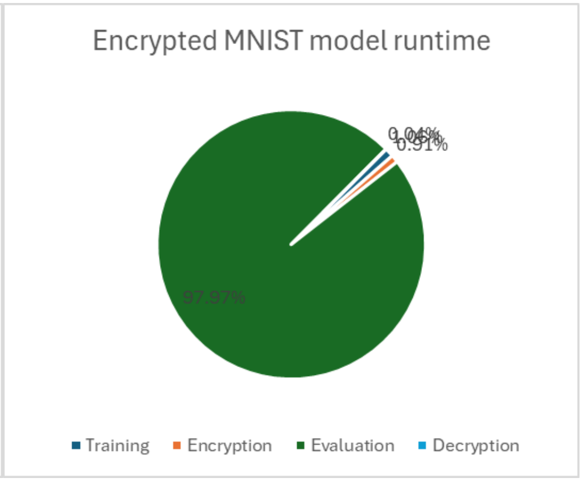


	Metrics					End-to-End Time (s)				
Model	Accuracy	Recall	F1 Score	Precision	AUROC	Training	Encryption	Evaluation	Decryption	Total
Encrypted MNIST	97.92%	0.9792	0.9792	0.9793	0.9994	125.88	108.54	11639.10	4.58	11879.68
Unencrypted MNIST	97.80%	0.9780	0.9780	0.9782	0.9993	109.88	N/A	9.88	N/A	119.96
Encrypted Medical	88.24%	0.9697	0.9143	0.8649	0.9428	14.18	0.47	60.10	0.03	76.21
Unencrypted Medical	78.43%	0.8966	0.8254	0.7647	0.7618	15.27	N/A	0.26	N/A	15.54

Figure 1: Table showing each model's metrics: Accuracy, recall, F1-Score, AUROC, and End-to-End runtime. Significant differences highlighted in green.

The first main observation made from our findings is that the MNIST dataset is very easy to outperform on, regardless of encryption, and is a weak indicator of a method's effectiveness on a model. Since the dataset is so generalizable, it does not offer much in the way of more specialized topics, such as PPML for medical-based tasks. The second is that Encryption has a significant negative impact on runtime. This isn't entirely due to the additional overhead caused by the preprocessing, encryption, and decryption, as we originally assumed it would be. It seems that the additional processing time increases exponentially as the 10'000 MNIST images took almost 100x longer to process end-to-end with encryption than they did without encryption as shown in Figure 4. Similarly, the 51 MRI images took almost 5x longer to process end-to-end with encryption than without encryption. Of that end-to-end processing time, only 2% was occupied with training, encryption, and decryption together for the encrypted MNIST, with the other 98% of the time spent on evaluation alone. For the encrypted medical model, roughly 19% of the end-to-end time was spent on training, encryption, and decryption. This trend is evidenced in Figures 2 and 3. These findings suggest that the encryption itself is not what increases the complexity of the model, but rather the computational overhead required by the encrypted models in order to actually process and inference the input images.





Figures 2 & 3: Charts showing the encrypted medical model's (left) runtime distribution, and the encrypted MNIST model's (right) runtime distribution, in percentages. Encryption and Decryption phases are disproportionately smaller due to the quickness of each process.

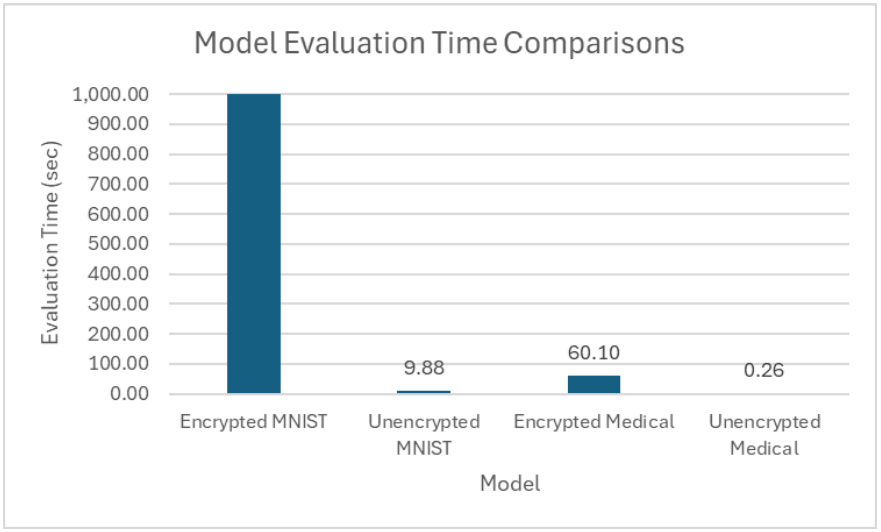


Figure 4: Chart comparing evaluation times of the four models. The Encrypted MNIST model's evaluation time extends far off the chart (11879.68 seconds), completely eclipsing the amount of time spent on evaluation by any other model. However, the encrypted medical model's evaluation runtime was also much larger than either of the two unencrypted models but had a significantly smaller dataset size.

Discussion

Let us first discuss the results from the MNIST dataset. Looking at the high accuracy on both the encrypted and unencrypted, 97.92% and 97.80% respectively, it demonstrates that the model does indeed work as intended. We chose this dataset as it is the main benchmark used for testing similar privacy-preservation methods published in the literature. The high accuracy demonstrates that the model did indeed work. The other metrics (recall, F1-score, precision, AUROC) also show to be nearly identical between both the encrypted and unencrypted MNIST dataset. This shows that the encryption had a negligible impact on model performance. Looking at the runtimes we can see that the unencrypted model ran at just under 2 minutes whereas the encrypted model took nearly 3.2 hours to finish running. This can be attributed to the massive discrepancy in evaluation time, which points to the fact that the fully encrypted models have a much higher computational overhead leading to much slower evaluation speed. Looking at the medical datasets we can see that the accuracy for both the encrypted and unencrypted datasets fell to 88.24% and 78.43% respectively. In addition to this the encrypted model has much better metrics in regards to its recall, F1-score, precision, AUROC values. Similarly the encrypted dataset took longer to run than the unencrypted dataset which would be the result of the encryption overhead, however the gap is significantly shorter. This is possibly due to the smaller dataset as medical was composed of 253 images. Based on these findings, some trends that have been shown are that encrypted models can sometimes outperform unencrypted ones when it comes to complex datasets such as the medical dataset we used. Another trend seen is that encrypted models had a much longer evaluation time likely as a result of a higher computational overhead due to the model running encrypted end-to-end.

Going into this project both of us did not have experience working with ML models became a limiting factor on what project we chose. We chose to work with a homomorphic encryption privacy preserving machine learning model found in a research paper. This model was open sourced so we had assumed we would be able to download the github and not have many issues. However the python homomorphic encryption library they used, pyfhel, was not working when we tried so we had to change to another HE library, tenseal. We also had to create new LaNet-1 instead of using the ones in the previous projects git repo. We didn't realize it at first but the encryption models were decrypting before evaluating making them not a fully encrypted process. To address this issue we had to switch schemes from BFV to CKKS which had support for tenseal making the implementation much easier as BFV did not natively support HE.

Our findings are simply a proof-of-concept to showcase the amount of data that can be obtained, and the narrowness of scope that can be applied to examining privacy-preservation techniques for the medical field. Time constraints and lack of expertise or experience resulted in the settlement of a suboptimal ML framework, as well as a smaller and more simplistic dataset. A more impactful and representative conclusion can be drawn from future experiments' findings by using a larger dataset with more class labels, keeping RGB channels, and a more realistic model architecture that can better represent the models to be used in real medical image classification tasks. There is enough data here to show the beginnings of a trend, but further research, as always, is required to truly draw out the conclusions of how encryption benefits and impairs a model's performance.

Conclusion

While this project illustrates some of the effects of encryption on a PPML model, its broader impact is that it exemplifies the need to dive deeper into analyses of privacy-preserving methods to truly gather an understanding of how such methods may affect the performance of the models they are applied to. We found that a more focused and topical study of PPML models reveals a more representative display of those models' performance on healthcare-specific tasks. This adds more relevant information to the literature that can then be generalized to the healthcare world more so than analyses based solely on benchmark evaluations and theoretical discussions. Furthermore, by including more metrics that overlap with those commonly seen in the literature, we hope to set a precedent for future research to continue using a wider range of metrics for easier and more impactful comparison between techniques.

More specifically, our experiments found that fully homomorphic encryption can be used to effectively hide sensitive data while reasonably maintaining accuracy for a model trained on a given task, but that the computational overhead required to process the encrypted inputs offers a strong barrier to the performance of such models for larger inputs. This poses a challenge for future researchers to continue to work toward lowering the computational barrier, or to develop new methods that avoid this impairment entirely. As analyses into the various encryption techniques and HE scheme continue to add to the literature, our collective understanding of the advantages and disadvantages of privacy-preserving techniques will continue to grow. This work serves as a stepping stone to a greater understanding of how to optimize and choose appropriate methods and techniques for solving complex healthcare tasks such as disease prediction, diagnosis, image analysis, genomics research, and beyond.

Next steps could include more-focused analyses into PPML models using the CKKS fully homomorphic encryption scheme by using more complex and specialized model architectures, more difficult tasks than binary classification, and larger datasets. This will result in more relevant conclusions by gathering metrics from experiments more closely aligned with real-world medical tasks, as well as gain a more accurate illustration of homomorphic encryption's effects on runtime. Additionally, the other HE schemes can be explored using similar experiments to get a broader view of homomorphic encryption's many avenues. By obtaining similar data points, hopefully using the metrics set by our precedent in this paper, the scientific community can begin to make valuable comparisons between the alternative privacy-preservation methods offered by homomorphic encryption. This, of course, presents yet another step that can be taken to extend this research: performing similar analyses on the other PPML techniques, such as federated learning and differential privacy, to gather uniform and generalizable comparisons between each method.

CAP 6938: Trustworthy Machine Learning
Fall 2024 - Final Project Report
Justin Morera & Sebastian Perez

Code and Implementation

All code files can be viewed, downloaded, and run here:

https://github.com/JustinMorera/CAP6938-Trustworthy-Machine-Learning-Final-Project

Dependencies:

- Python 3.11.15
- Python packages: (*pip install* ____)
    1. *numpy*
    2. *torch*
    3. *torchvision*
    4. *sklearn*
- Tenseal Python package: https://github.com/OpenMined/TenSEAL?tab=readme-ov-file#installation

Installation:

pip install tenseal

python import tenseal as ts

CAP 6938: Trustworthy Machine Learning
Fall 2024 - Final Project Report
Justin Morera & Sebastian Perez

References

Falcetta, A., & Roveri, M. (2022). Privacy-preserving deep learning with homomorphic encryption: An introduction. *IEEE Computational Intelligence Magazine*, *17*(3), 14-25.

Falsetta, A, Bastiaan, Q, & Andrea. (2022). Introduction to BFV HE ML [GitHub Repository]. GitHub. https://github.com/AI-Tech-Research-Lab/Introduction-to-BFV-HE-ML

Guerra-Manzanares, A., Lopez, L. J. L., Maniatakos, M., & Shamout, F. E. (2023, May). Privacy-preserving machine learning for healthcare: open challenges and future perspectives. In *International Workshop on Trustworthy Machine Learning for Healthcare* (pp. 25-40). Cham: Springer Nature Switzerland.

Navoneel Chakrabarty. (2018). *Brain MRI images for brain tumor detection* [Dataset]. Kaggle. https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection/data

OpenMined. (2021). TenSEAL: A library for doing homomorphic encryption operations on tensors [GitHub Repository]. GitHub. https://github.com/OpenMined/TenSEAL