Privacy-Preserving Machine Learning on Clinical & Genomics Data Final Report | 5/10/2024

BERK TURHAN, Weill Cornell Graduate School of Medical Sciences, NY
ANANYA DEVARAKONDA, Cornell Tech, NY
SIDHARTH VASUDEV, Cornell Tech, NY

1 INTRODUCTION

To date, health data collection spans various areas such as electronic healthcare records, genomic and pharmaceutical data, clinical trials, as well as information on well-being, behavior, and socioeconomic indicators [Pastorino et al. 2019]. Over the past few decades, there has been growing excitement about analyzing these data to improve personal care, clinical practices, and public health [Vayena et al. 2017]. The increasing complexity and abundance of healthcare data also indicate a growing use of artificial intelligence (AI) within the field [Davenport and Kalakota 2019]. Various machine learning (ML) algorithms have found many applications in different healthcare domains and have contributed to effective and accurate clinical decision-making [Kourou et al. 2015]. More recently, deep learning, a subfield of ML, has experienced significant breakthroughs owing to increased computational power, advancements in model architectures [Esteva et al. 2019], and the exponential growth of available datasets collected by many different means [Zhu et al. 2020]. By directly learning hierarchical representations from data, deep learning can effectively manage the high-dimensionality of datasets. This eliminates the need for manual feature selection and has the potential to capture more relevant biological & health data-related signals [Lee 2023]. However, to increase the performance of deep learning models in healthcare, algorithms leverage large amounts of data to learn [Esteva et al. 2019]; in contrast, the amount of patient data is still relatively small for different individual clinical studies [Zhu et al. 2020].

In order to overcome this problem, the existing biomedical and health research model has transitioned from individual institutional studies to collaborative efforts involving multiple institutions [Scheibner et al. 2021]. However, it is challenging to facilitate data sharing that preserves individual privacy and data utility [Scheibner et al. 2021], where many instances of healthcare data breaches have seen a notable increase on a global scale [Murdoch 2021]. Additionally, healthcare data for ML and deep learning applications usually have to be uploaded to cloud servers or Graphical Processing Units (GPUs), adding another step in data processing where potential data compromise can occur [Yadav et al. 2023]. Hospitals and research institutes have been working on advancing technical solutions to facilitate the sharing of patient data in a manner that prioritizes privacy preservation [Scheibner et al. 2022]. There are many studies that discuss healthcare privacy in public-private partnerships, such as those of insurance companies and pharmaceutical industries. They implement trust-based decentralized models like federated learning [Xu et al. 2020] or site-level meta-analysis that tries to limit physically transferred patient data or only share updates of the ML model parameters rather than actual data for privacy [Scheibner et al. 2022]. However, decentralized models are rarely employed in medical research due to their notable impact on data utility, therefore allowing many cryptographic privacy-enhancing technologies to surface as promising advancements, homomorphic encryption (HE) [Gentry 2010] being one of the most prominent and popular candidates [Scheibner et al. 2022]. HE is a type of encryption that enables specific computations to be conducted on encrypted data, producing an encrypted result. When decrypted, this result corresponds to the outcome of operations performed on the plaintexts (unencrypted data) [Yi et al. 2014]. Hence, it can protect sensitive health information by allowing data to be processed in encrypted form, ensuring that only encrypted data are accessible to other parties [Munjal and Bhatia 2022]. Due to this property, there is ongoing discussion about the potential benefits for the healthcare business as the performance and utility of HE will continue to improve [Munjal and Bhatia 2022]. In addition to private-public data sharing and healthcare business, HE performance and applicability are also yet to be explored in genomic and clinical settings for patient health. Even though there are survey-based expert assessment studies to review the possible applications of HE for research institutions [Scheibner et al. 2022; Wood et al. 2020], there is still a need for the application and practical examination of HE in the context of machine learning and deep learning utilizations in clinical and genomic healthcare data. A study from 2021 attempts to establish homomorphic encryption (HE) with federated learning over a dataset from cBioPortal with both clinical & genomics features [Froelicher et al. 2021]. However, the code base and pipeline developed for the project are not open-sourced due to copyright issues or possible collaborations with industry partners to commercialize the products, limiting further exploration. Therefore, in this study, we apply homomorphic encryption (HE) to clinical and genomic datasets from the cBioPortal which is an open-access, open-source resource multidimensional cancer genomics database to train classical machine learning and state-of-the-art deep learning architecture models with the encrypted data to explore the encryption's practicality and applicability in a clinical context.

After preprocessing this dataset and obtaining baseline results, we conduct multiple experiments to compare and contrast the change in performance when using homomorphically encrypted data to train or evaluate various ML models: XGBoost and a 1-layer neural network (logistic regression). As we discuss in subsequent sections, while we found that our models consistently underfit given dataset limitations and resource constraints, homomorphically encrypted data only causes slight fluctuations and noise in resulted metrics. Therefore, we propose that there is promise in using HE data with scope for future research.

Authors' addresses: Berk Turhan, bt364@cornell.edu, Weill Cornell Graduate School of Medical Sciences, Tri-Institutional PhD Program in Computational Biology, New York, NY; Ananya Devarakonda, ad834@cornell.edu, Cornell Tech, New York, NY; Sidharth Vasudev, sv355@cornell.edu, Cornell Tech, New York, NY.

2 RELATED WORK

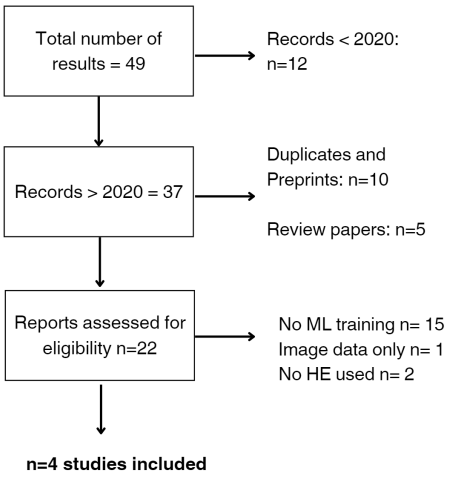

Fig. 1. PRISMA flow diagram for literature review

Our method for identifying relevant work involved searching Google Scholar and PubMed for studies on the use of HE in ML models trained on clinical data to predict cancer-related outcomes. The following keyword search: "cancer" "artificial intelligence" OR "deep learning" OR "clinical dataset" OR "cBio" OR "cBioPortal" "Homomorphic encryption" "machine learning" "genomic dataset" resulted in 49 papers. We then filtered these results based on the following eligibility criteria:

(1) Papers must have a publication date after 2020 and must be primary research articles. With the increase in open-source HE implementations available, we see an increase in the number of studies after 2020.
(2) Papers must discuss machine learning and use data that has undergone homomorphic encryption either during training, evaluation, or both.
(3) Papers must use cancer-related genomics data.

Figure 1 demonstrates our filtering process at each step, focusing on papers discussing genomic features to advance cancer care in a privacy-preserving way using homomorphic encryption. [Sarkar et al. 2023] implements a HE-based model to predict the type of cancer with a large volume of genetic mutation data that consists of more than 2 million genetic mutations from 2713 patients and focuses on somatic mutation encoding approaches. They also propose a fast matrix multiplication algorithm for homomorphic encryption. [Hong et al. 2022] implement homomorphic encryption in a 1-layer neural network with the approximation of softmax activation function for multilabel tumor classification. [Carpov et al. 2022] present GenoPPML: a privacy-preserving ML framework for breast cancer metastasis prediction using genomic data. They use a logistic regression model and encrypt data using the TFHE scheme [Chillotti et al. 2020]. More recently, [Song and Shi 2024] propose a neural network model that supports inference on HE: REsidue ACTivation HE (ReActHE). They evaluate their model, which is based on scaled power activation on genomics data, including The Cancer Genome Atlas (TCGA) dataset on various ML models and ReActHE.

Eventually, the main goal of medical algorithms, including machine learning approaches, is to promote the patient's best interests by improving healthcare outcomes such as quality and length of life [Liu et al. 2022]. Notice that these related works discussed come from single genomic studies and only try to do cancer type or tumor classification with genomic features. Although these studies to explore secure ML models in the genomic context are really important in advancing the basic science behind cancer biology, they do not focus on classifying clinical outcomes. Therefore, there is still a gap in the field to explore the usability and applicability of secure homomorphic encryption in secure ML models using integrated clinical features and genomic features together to classify & predict patient-level clinical outcomes. Furthermore, exploring homomorphic encryption on a data set coming from a collaborative platform would further strengthen the collaboration efforts between institutions if secure ML models are implemented successfully. Hence, our approach is novel as we aim to implement these homomorphic encryption techniques with machine learning methods on the MSK-IMPACT dataset [Zehir et al. 2017] that comes from the popular and widely used platform, cBioPortal, with open-source libraries and algorithms that try to classify clinical outcomes using clinical & genomic features. Our implementations seek to create a baseline for homomorphic encryption in machine learning and deep learning in an area not extensively explored or utilized in the genomics and clinical context together. This initiative will be valuable for advancing research at the intersection of AI and genomics, providing a foundation for other researchers to build upon and further explore the potential of homomorphic encryption in healthcare data analysis.

3 RESULTS AND DISCUSSION

3.1 Clinical and Genomics Feature Selection

As discussed in the Introduction, differing from previous studies, we obtained and preprocessed the MSK-IMPACT Clinical Sequencing Cohort dataset, which includes targeted genomic sequencing data from 10,000 clinical cases using the MSK-IMPACT assay [Zehir et al. 2017]. After preprocessing (see Detailed Methods for detailed preprocessing steps), we ended up with 18 features, with 13 of them being clinical features (*Cancer Type, DNA Input, Metastatic Site, Primary Tumor Site, Sample Class, Sample Collection Source, Number of Samples Per Patient, Sample Coverage, Sample Type, Sex, Smoking History, Specimen Preservation Type, Specimen Type*) and the remaining 5 being genomic features (*Fraction Genome Altered, Mutation Count, Somatic Status, Tumor Mutational Burden, Tumor Purity*) to classify ***overall survival status*** (5465 living patients, 2164 deceased patients). Out of these 18 features, 5 are further one-hot-encoded due to having multiple categorical unique entries (*Cancer Type, Primary Tumor Site, Smoking History, Specimen Preservation Type, Specimen Type*), while another 5 are encoded with binary encoding for having two categorical unique entries (*Sample Class, Sample Collection Source, Sample Type, Sex, Somatic Status*). This resulted in our final dataset, which is fed into the models and encryption algorithms, containing 67 numerical features across 7629 observations. All 67

final features can be found in the preprocessed data available on our GitHub repository.

3.2 Non-encrypted ML Models

Before encrypting the data for privacy-preserving approaches, to create a baseline, we ran logistic regression, lasso regression, ridge regression, decision trees, random forest, and XGBoost models on non-encrypted (also known as plaintext) data (see Detailed Methods for detailed model specifications). Regression-based models trained and evaluated on non-encrypted standard scaled data performed similarly to each other, averaging around 62% accuracy, 65% F-1 score and 67% ROC-AUC for training and test sets (Fig.2). On the other hand, for tree-based models trained on the same dataset, the decision tree model performed poorly with an accuracy around 50%, where random forest and XGBoost outperformed every other model with scoring around 70% F-1 score and 70% ROC-AUC (Fig.3).

Fig. 2. Accuracy, F-1 and AUC-ROC scores for regression based models.

It is important to note here that for regression-based models, L1 and L2 regularization did not significantly improve performance, due to not having an overfitting problem. Similarly, boosting did not result in a significant performance increase in our metrics for tree-based models as well. Additionally, we did not see a difference between training and test performances. Therefore, we can argue that we have an underfitting problem, where even the best metrics are stuck around 70% performance.

3.3 Encrypted Evaluation

In order to adopt a privacy-preserving approach, we initially focused on encrypting the evaluation (testing) phase of standard machine learning pipelines. To achieve this, after training our model with a non-encrypted dataset, we fed our homomorphically-encrypted test set to the model for predictions. For all results, we ran k-fold cross-validation (see Detailed Methods for specifications on the k value). We then performed arithmetic operations between the model's learned non-encrypted numerical weights and our encrypted test

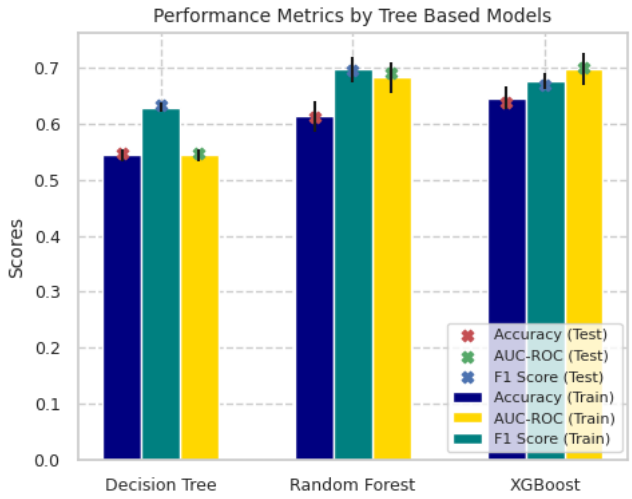

Fig. 3. Accuracy, F-1 and AUC-ROC scores for tree based models.

set numerical vector queries to generate predictions, as homomorphic schemes inherently allow for this. Furthermore, for additional privacy, we were able to homomorphically encrypt our model's weights after training on non-encrypted data. Subsequently, we were able to generate predictions by performing arithmetic operations between encrypted model weights and encrypted test queries due to HE's nature. These privacy-preserving evaluations (inference on testing data) were conducted for both XGBoost tree-based model and a 1-layer neural network (shallow NN, equivalent to logistic regression) to assess the applicability of homomorphic encryption for our dataset across different machine learning models.

3.3.1 XGBoost encrypted evaluation. In this section, we present the following:

(1) Results from the XGBoost model (with non-encrypted tree structure) on non-encrypted test data or "unencrypted plain evaluation".
(2) Results from the XGBoost model (with encrypted tree structure) on encrypted test data or "encrypted test evaluation".

In both cases, the XGBoost model is trained on non-encrypted data. For encrypted evaluation, we used the **ppxgboost** library developed by Amazon AWS labs [Meng and Feigenbaum 2020]. ppxgboost performs homomorphic encryption of the learned xgboost tree structure and test data to infer predictions in a privacy-preserving manner (see Detailed Methods for ppxgboost information).

As it can be seen from Fig. 4, encrypted test evaluation, where we encrypt the model's parameters, resulted in a comparably similar performance with unencrypted plain evaluation. It is important to note that the resulting metrics are not *exactly* the same (deviating by ± 0.04 on average), and we can see the decreased performance for encrypted evaluation. Observing the error bars, which show the standard error within our multiple evaluation runs, encrypted evaluation varies more than plain unencrypted evaluation. Hence, it has a lower mean value as a result of the higher variation. This variation is likely to be caused by the noise originating from the

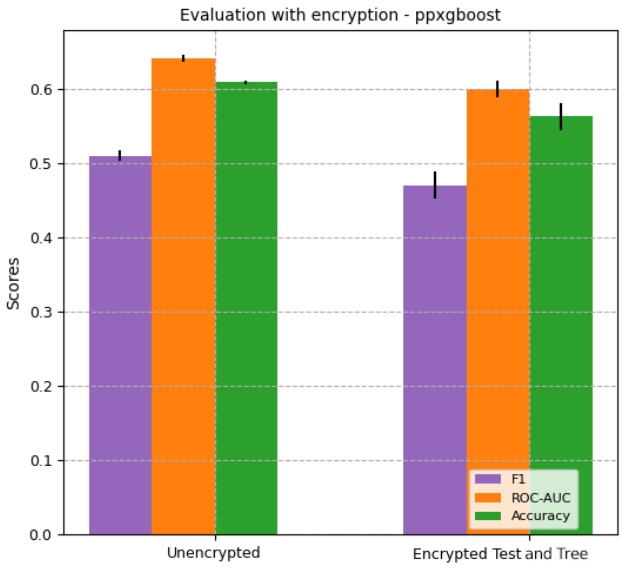

Fig. 4. Test accuracy, F-1 and AUC-ROC scores for evaluations of XGBoost model using ppxgboost encryption.

precision errors of arithmetic operations on the homomorphically encrypted data. HE schemes use certain approximations to facilitate multiplication and addition operations on encrypted numbers, making them somewhat error-prone and resulting in slightly imprecise arithmetic operation results. Nonetheless, we can argue that the drop in metrics is not significant, and the prediction performance is still comparable. Therefore, we can argue that encrypted evaluation of XGBoost models is a viable privacy-preserving approach for evaluating new data points from clinical & genomics datasets.

3.3.2 1-layer-NN encrypted evaluation. In this section, we present the following:

(1) Results from the model (with non-encrypted weights) on non-encrypted test data.
(2) Results from the model (with non-encrypted weights) on encrypted test data.
(3) Results from the model (with encrypted weights) on encrypted test data.

In all cases, the model is trained on non-encrypted data. For privacy-preserved neural network-based model evaluations, we utilized **TenSEAL**, a library for performing homomorphic encryption operations on PyTorch tensors(see Detailed Methods for TenSEAL information). TenSEAL is built upon Microsoft's popular HE scheme SEAL [Benaissa et al. 2021]. Initially, we trained a 1-layer neural network model consisting of only one linear layer with PyTorch, using Stochastic Gradient Descent (SGD) as the optimizer and Binary Cross Entropy (BCE) as our loss function. Following the training, we initially evaluated the model by feeding non-encrypted test queries and calculating metrics with predictions. Then, we repeated the evaluation process for the same model, but this time by encrypting the test queries using TenSEAL before feeding them into the models.

Additionally, we conducted a third evaluation by further encrypting the learned linear layer weights and biases before feeding encrypted test queries to obtain predictions, where the resulting metrics can be seen from Fig.5.

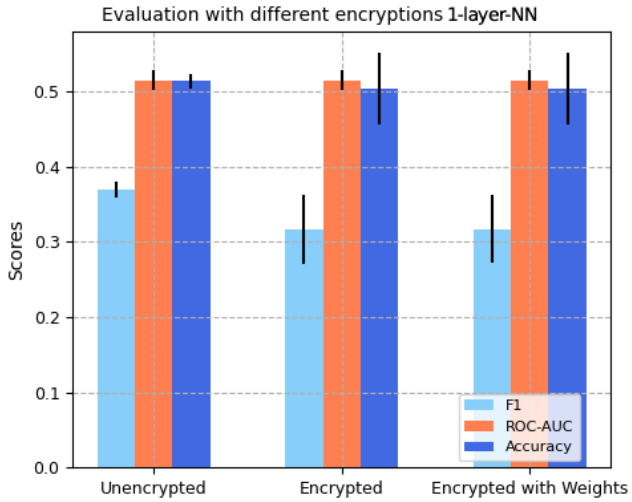

Fig. 5. Test accuracy, F-1 and AUC-ROC scores for different evaluations of 1-layer-NN model.

In our experiments with different encrypted evaluation approaches, we found that encrypting the test set and further encrypting the model's weights does not lead to a significant decrease in performance metrics, but it does introduce noise due to the arithmetic operation approximations in homomorphic encryption and decryption. As shown in Fig. 5, we observe a slight decrease in F-1 score and accuracy during encrypted evaluation. However, the error bars indicate high variability in our result metrics, resulting in lower mean scores. This can again be attributed to the noise from the approximation precision of HE arithmetic operations, where we have the same observation for encrypted XGBoost evaluation, too (see Fig. 4). Additionally, we did not observe a significant performance difference between encrypting only the test set or also encrypting the model's learned parameters. Generally, all of the models ended up having bad performance metrics. Nevertheless, we can argue that encrypted evaluation is a viable privacy-preserving approach for evaluating new data points using NN models trained on clinical genomics datasets.

Since the performance metrics on Fig. 5 are poor, we further resampled our data to address the class imbalance problem, aiming to possibly increase the performance by removing underfitting effects. This was done to achieve higher scores during training and non-encrypted evaluation, allowing us to observe the effect of encrypted evaluation on higher-performing metrics. When resampled, we observed a significant improvement in metrics across all models. For example, F-1 scores increased from 30 – 40% to over 60%, and accuracy improved from 50% to 60% (see Fig. 6). Interestingly, encrypted evaluations ended up having higher F-1 scores compared to plain test query evaluations, which again can be explained by

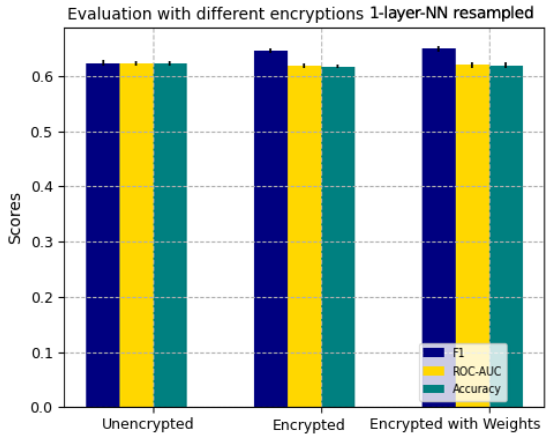

Fig. 6. Test accuracy, F-1 and AUC-ROC scores for different evaluations of 1-layer-NN model trained on resampled data.

HE precision errors that introduce noise to our results. Yet, we can once again claim that encrypted evaluation remains a viable privacy-preserving approach for evaluating new test data points using NN models trained on non-encrypted clinical & genomics datasets.

It is important to note here that even for non-encrypted evaluations of the 1-layer neural network (equivalent to logistic regression) on both original and resampled data, our resulted metrics are worse than our base regression models (see Fig. 2). This is because our base models, built with libraries like scikit-learn, utilize more advanced optimization algorithms and loss functions compared to our simple 1-layer neural network implemented with PyTorch linear layer, using SGD and BCE. Furthermore, we also tend to observe highly varying performances during different runs for all evaluation techniques. This high variability can be due to SGD converging on different local minimums, coupled with encryption's precision errors due to approximations. As a result, there are instances where all metrics fail drastically, making performance between different evaluation approaches incomparable. However, the reason for choosing such a basic architecture and optimization approach for our 1-layer neural network is that we further train it on encrypted data, which requires a considerable amount of time and space due to the high-bit requirements of encrypted vectors, heavy approximation functions, and many matrix operations. Therefore, we kept our 1-layer neural network simple enough to be able to compare the applicability of homomorphic encryption for evaluation and training on our clinical and genomics dataset.

3.4 Encrypted Training

In this section, we present the following:

(1) Results from the model (with non-encrypted weights) on non-encrypted test data.
(2) Results from the model (with non-encrypted weights) on encrypted test data.

In all cases, the model is trained on encrypted training data. It is important to consider that training machine learning models on encrypted data using the CKKS scheme [Cheon et al. 2017] of homomorphic encryption (HE) introduced substantial spatial and temporal complexities. Specifically, the memory requirements for encrypted operations increased more than 50,000 times compared to unencrypted data, while computational times for training were slowed by a factor of over 10,000 times. These drastic increases arise from the complexities involved in managing and operating on CKKS vectors, which are essential for encoding and processing encrypted data. Such overheads made it impractical to train multi-layer neural networks, as each additional layer exponentially compounded these computational and memory demands. Given these constraints, our experiments were limited to training only single-layer neural networks. Despite these limitations, the investigations into the performance of these models provided valuable insights.

Restricted to single-layer models due to the described complexities (see Detailed Methods for model specifications), our training efforts still yielded informative outcomes. The performance metrics from our experiments showed that the encrypted models, although simple, could achieve results comparable to their unencrypted counterparts. This equivalence in performance underlines the potential of HE in securing data without fundamentally undermining model effectiveness, provided the model complexity is kept manageable within the HE constraints.

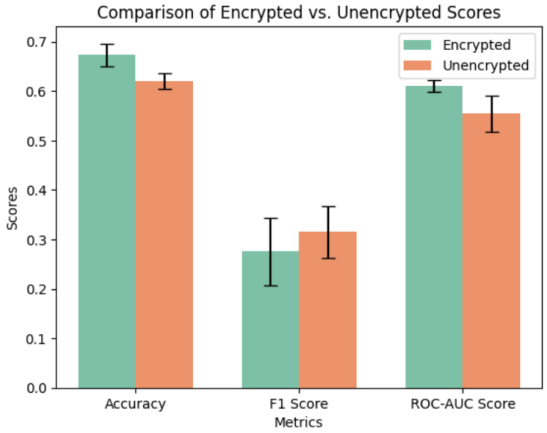

Fig. 7. Comparison of performance metrics between encrypted and unencrypted models over 5-fold cross-validation.

As can be seen in the results from Fig. 7, our algorithm for training a model using encrypted data works similarly to the one on unencrypted data barring a little noise. While we do keep the hyperparameters the same due to the inherent complexity of trying to optimize the hyperparameters on the encrypted model. However, we do see that both models clearly underfit the dataset, given the relatively lower scores on the accuracy and f1 scores demonstrated by the models. This does raise the question of if we were able to actually train a more complex model that could fit the dataset, would we still see the same similarity of performance.

However, given the computational infeasibility of training such models, we can only remark on our current results. These results

demonstrate that, despite the significant computational and spatial challenges and the restrictions on model complexity, encrypted training works on approximations that are accurate enough to mimic the performance of models trained on encrypted data, using traditional ML techniques. This shows that machine learning on homomorphically encrypted data is still a viable technique for conducting secure and private data analysis in sensitive fields such as clinical and genomics research. The main considerations when employing such a technique should be the compute requirements to train the model, especially a good model, which would require hyperparameter optimization and, therefore, more training. However, the findings encourage further development in HE techniques and training methodologies to mitigate current limitations and enhance the usability of privacy-preserving machine learning models.

4 CONCLUSION AND LIMITATIONS

Within our study, we successfully trained and evaluated privacy-preserving ML models on our clinical & genomics dataset using homomorphic encryption. Initially, we established a baseline by training widely used regression and tree-based machine learning algorithms without any encryption approach. We then performed encrypted privacy-preserving evaluations (testing) of these models using HE schemes. As a next step, we also trained a simple 1-layer neural network model (equivalent to logistic regression) on encrypted data. Therefore, we successfully demonstrated a comprehensive privacy-preserving ML approach on our clinical & genomics dataset from training to testing. However, there are a few points that are worth further discussing.

Firstly, we observe underfitting in all of our models, regardless of encryption, stemming from our dataset. Hence, when we use simple models such as 1-layer-NN with gradient descent, the model may converge to local minima, which further affects performance metrics in all of our experiments. Additionally, it is also discussed in ppxgboost tutorials that homomorphic arithmetic operations may not always yield to same exact results due to precision. Similarly, TenSEAL tutorials show a slight accuracy decrease after encrypted evaluation. Literature shows that there is an expected precision loss in approximate homomorphic encryption, including our CKKS scheme, arising from both encryption and encrypted operations [Costache et al. 2022]. Even though our optimal goal is to reduce this precision loss, it predominantly depends on scheme selection. When a model underfits, predicted probabilities tend to be closer to 0.5, since the model cannot be sure about the exact label. Considering precision loss and the fact that predicted probabilities are closer to 0.5, we can expect to see some varying results, especially in cases where gradient descent gets stuck in local minima, leading to poor performance initially. We ran our codes many times, and our Figures 4, 5, 6 are the most frequent metric results that we observed. However, in different runs, the metrics can be slightly off, even though they will generally follow the same pattern.

Secondly, training and evaluation of machine learning models on encrypted data using (HE) is markedly more complex and computationally demanding. Encrypted evaluation requires a significant amount of time for encryption of the test queries and model weights, encrypted arithmetic operations, and decryption of the encrypted predictions. ppxgboost discusses that for encrypted evaluation, the inference is approximately 10^3 times slower than the plain version of XGBoost. Additionally, the size of encrypted models is also between four and nine times larger than that of the plain models [Meng and Feigenbaum 2020]. This situation becomes even more problematic during training due to its iterative process, which requires repeated modifications to model parameters based on gradient computations and back-propagation through potentially deep network structures. One of the primary challenges in training over evaluation using HE is the need to carefully configure encryption parameters, such as the polynomial modulus degree and the coefficient modulus sizes. Additionally, training often involves approximating non-linear functions, such as activation functions in neural networks, requiring these functions to be represented as polynomial approximations, which both take time and significantly change the performance. This necessitates a higher number of coefficient encryption modulus to support the depth of arithmetic circuits needed for such approximations, allowing for a deeper level of encrypted computations. This significantly increases the performance burden and expands the size of encrypted results, adding further time and space complexity to the training process on encrypted data.

Finally, there are studies in the literature that significantly outperform us. Sarkar et al. implement a privacy-preserving cancer type prediction using a dataset consisting of more than 2 million genetic mutations from 2713 patients, achieving an accuracy of 83.61% [Sarkar et al. 2023]. However, there are a couple of reasons lying under this study that are limited in our approach. The first and most apparent reason is that they work on a cloud system with much higher resources than our study. Secondly, they only use genomic data single nucleotide variants (SNV) and copy number variants (CNV) and are able to extract 2 million genetic mutations that hold much more information than our dataset. In our approach, to evaluate the usability of HE in a dataset coming from two different domains, clinical and genomics, we operate on a patient level rather than a strictly genomic level. This means that we lose the ability to represent the many different mutations within a patient in our dataset, which limits our model's ability to learn information about the severity of specific mutations. This is important for our task of predicting overall survival status. They also develop a novel HE matrix multiplication algorithm that works well on their SNV-CNV representations, which speeds up their training process. When we work with simple models due to a lack of resources, lose the gene-level information, apply HE, and approximate activation functions, it causes precision errors. It is not surprising that we see highly variable and low-performance metrics. However, our results are still promising, especially for the encrypted evaluation of the models. This can be utilized in collaborative efforts to ensure data safety during the evaluation phase of developing a novel model that will be making predictions on clinical and genomics data domains together, such as the popular cBioPortal. In order to further increase the performance of privacy-preserving models in the clinical and genomics context, more efforts need to be taken towards efficiently representing data coming from different domains. This includes exploring different embedding approaches without losing the gene-level information and developing more efficient matrix operations or HE schemes specific to these embeddings.

5 DETAILED METHODS & SUPPLEMENTARY

5.1 Dataset and Preprocessing

5.1.1 Dataset Introduction. As noted throughout the paper, our dataset is from the cBioPortal, more specifically, the MSK-IMPACT Clinical Sequencing Cohort dataset [Zehir et al. 2017]. This dataset is curated with 10, 945 cancer samples coming from 10, 336 patients and includes molecular profiles of these samples with the addition of the patient's clinical information. We chose to work with this dataset because it is rare, has a relatively larger number of samples, which is rare in cancer studies, and requires substantial curation efforts. With more than ten thousand samples, the dataset covers different cancer types and clinical information and provides detailed genomic information.

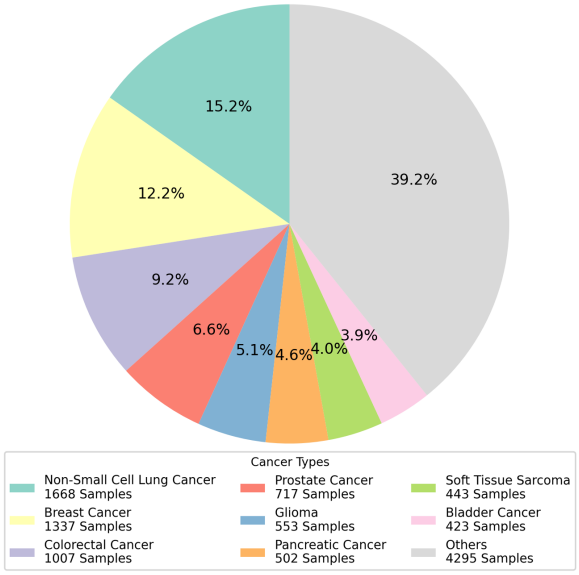

Fig. 8. 8 most abundant cancer types and their sample sizes in MSK-IMPACT dataset. In total, there are 58 different cancer types reported in this study, where 50 other types are labeled as "Others" in the pie chart.

In addition to the cancer types, the dataset includes clinical information such as sex, metastatic sites, sample collection sources, and smoking history. With these features, we trained models that will aim to predict the overall survival status of the patients (0: alive, 1: deceased).

Besides clinical information, our dataset also contains somatic mutations and novel non-coding alterations for different tumor samples, and this clinical & genomics data integration makes our approach novel. These somatic mutation counts are also incorporated in our training set by sample level by calculating the fraction of genome altered (fraction of log2 copy number variation gain or loss > 0.2 divided by the size of the genome whose copy number was profiled [Zhou et al. 2019]) and tumor mutational burden (TMB - number of somatic mutations per megabase of interrogated genomic sequence [Sha et al. 2020]) numerical attributes.

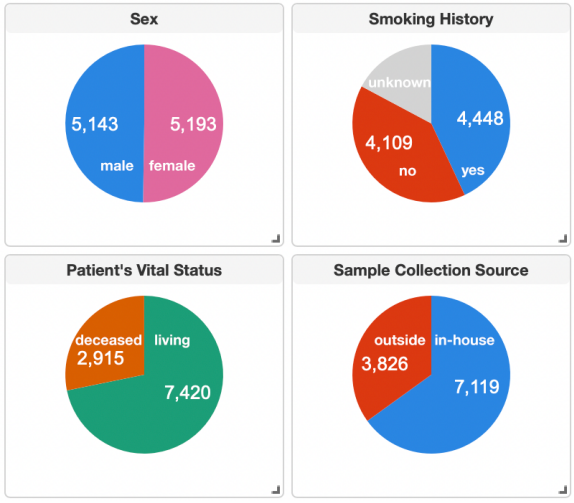

Fig. 9. Clinical variables and number of samples for each class.

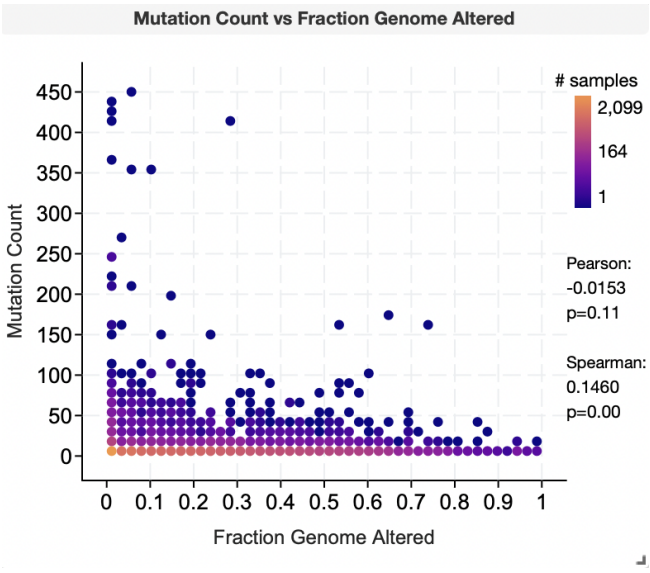

Fig. 10. Fraction genome altered and the mutation count as the features

5.1.2 Preprocessing. We retrieved our dataset from "https://www.cbioportal.org/study/summary?id=msk_impact_2017", where it is maintained as open-sourced. Originally, the dataset had 27 features and 10945 observations. After our initial investigation, we dropped multiple columns due to different reasons as follows: ["Study ID", "Patient ID", "Sample ID"] are dropped since they are just unique identifiers, ["Cancer Type Detailed"] is dropped since it just contains detailed medical text about the cancer type, ["Overall Survival (Months)", "Patient's Vital Status", "Oncotree Code"] are dropped to prevent information leakage (as we are predicting overall survival status) and since there are duplicated repetitive features where we have the exact same information in other columns (e.g., "Cancer Type" and "Oncotree Code" are just 2 different nomenclature for

same cancer). We finally further dropped the ["Matched Status"] feature, which has the same entry for 99.5% of our samples in the dataset. We ended up with 19 features, with 14 of them being clinical features (*Cancer Type, DNA Input, Metastatic Site, Primary Tumor Site, Sample Class, Sample Collection Source, Number of Samples Per Patient, Sample Coverage, Sample Type, Sex, Smoking History, Specimen Preservation Type, Specimen Type, Overall Survival Status*) and the remaining 5 being genomic features (*Fraction Genome Altered, Mutation Count, Somatic Status, Tumor Mutational Burden, Tumor Purity*). Then we separated "Overall Survival Status" as our target feature to predict, and as a result, we finished up with having 18 training features (13 clinical, 5 genomics).

After dropping columns with domain knowledge, we then separated our dataset into training and test sets with a 30% test ratio. We further removed observations that have missing values for ["Fraction Genome Altered", "Primary Tumor Site", "Specimen Type", "Smoking History", "Overall Survival Status"] features, and imputed "Tumor Purity" feature with the mode (separately for training and testing sets to prevent information leakage). After removing the rows with missing values, we ended up with 7629 training samples and 3279 test samples (10908 samples total), only losing 37 samples.

Out of the 18 features we have, 8 of them are numerical features (such as TMB and mutation count...), while the other 10 are categorical features that need encoding. Before encoding, we checked the unique entries for these categorical features since one-hot encoding would add new features based on that. After investigation, we found that cancer type and primary tumor sites have many different unique entries which are relatively rare. Hence, we grouped rare entries into a different group called "Other" and represented only the top 20 cancer types and primary tumor sites. We also noticed that "Metastatic Site" is None for many of the entries, and each site that exists is relatively rare. Therefore, we turned "Metastatic Site" into a binary 1-0 feature to indicate whether or not there is a metastasis.

We then performed one-hot encoding on *Cancer Type, Primary Tumor Site, Smoking History, Specimen Preservation Type, Specimen Type* for having multiple unique entries, and binary encoding on *Sample Class, Sample Collection Source, Sample Type, Sex, Somatic Status* for having only two categorical unique entries. This resulted in our final dataset, which is fed into the models and encryption algorithms, containing 67 numerical features across 7629 observations. The models are then tested on 3279 observations. We also further standardized our features by removing the mean and scaling to unit variance before feeding them into our models.

5.2 Non-encrypted Base ML Models

5.2.1 Logistic Regression. We first trained and evaluated a logistic regression (LR) algorithm using *scikit-learn* without enforcing any regularization penalty. We feed additional class_weight parameters as balanced to make the model aware of our class imbalance problem. Then, we trained and tested on a separate validation test with a stratified KFold approach with 10 splits and reported accuracy, F-1, and ROC-AUC metrics both during the 10-KFold training validation set and one final time on the testing set.

5.2.2 Lasso Regression. After training an LR model without any regularization, we then trained another LR model enforcing the L-1 (Lasso) penalty and a lib-linear solver (https://www.csie.ntu.edu.tw/~cjlin/liblinear/) using *scikit-learn*. This time, we further did hyperparameter optimization for regularization strength with GridSearchCV using Stratified 10-KFold based on a weighted F-1 score. We then selected the model with our best validation metrics and reported the final performance on our testing set.

5.2.3 Ridge Regression. Similarly, we also trained an LR model with enforcing the L-2 (Ridge) penalty and a lib-linear solver using *scikit-learn*. We again did hyperparameter optimization for regularization strength with GridSearchCV using Stratified 10-KFold based on a weighted F-1 score, selected the model with our best validation metrics, and reported the final performance on our testing set.

5.2.4 Decision Trees. After training regression-based models, we further trained tree-based algorithms and started with a basic decision tree, again using *scikit-learn*. We did hyperparameter optimization with GridSearchCV using Stratified 10-KFold to optimize [max_depth, min_samples_split, min_samples_leaf], and report the metrics of our best estimators' performance during KFold training and on test set.

5.2.5 Random Forests. As another tree-based model, which is known to be a more complex model compared to a single decision tree, we trained a random forest using *scikit-learn*. We did hyperparameter optimization with GridSearchCV using Stratified 10-KFold to optimize [n_estimators, max_depth, min_samples_split, min_samples_leaf], and report the metrics of our best estimators' performance during KFold training and on test set.

5.2.6 XGBoost. In order to try to improve the performance metrics of our model tree-based model, we further trained an XGBoost model [Chen and Guestrin 2016]. We did hyperparameter optimization with RandomizedSearchCV using Stratified 10-KFold to optimize [max_depth, subsample, colsample_bytree, gamma, learning_rate, n_estimators], where we fixed our objective function as binary: logistic, our evaluation metric as logloss and scale_pos_weight as our class imbalance ratio. We then report the metrics of our best estimators' performance during KFold training and on the test set.

5.3 Encrypted Evaluation

5.3.1 XGBoost evaluation with ppxgboost. In order to have a privacy-preserving approach to test and evaluate our tree-based boosted XGBoost model, we used developed by Amazon AWS labs [Meng and Feigenbaum 2020] library which is developed by Amazon AWS labs [Meng and Feigenbaum 2020]. This library allows users to send encrypted queries to a remote ML service, receive the encrypted results, and decrypt them locally. It uses Paillier encryption for the homomorphic approach to let arithmetic operations on encrypted numbers [Paillier 1999]. Additionally, in order to make sure the orders of the multiple queries & that are getting encrypted are preserved, it utilizes the Boldyreva order-preserving encryption (OPE) scheme [Boldyreva et al. 2011].

To utilize and test the encrypted evaluation, we first trained an XGBoost model with using some of the hyper-parameters that

were found optimal by our base model runs on Fig.3 ('eta': 0.1,'subsample': 0.8, 'max_depth': 5, 'gamma': 0.1, 'colsample_bytree': 0.8, 'eval_metric': 'logloss', 'scale_pos_weight':3 - not every hyperparameter was available to use on ppxgboost). When the model is trained, we feed our non-encrypted training data and recorded metrics.

After establishing our encryption scheme and having a trained model, we homomorphically encrypted both the model's tree structure and our plain-test queries (unencrypted test data), ensuring the preservation of the tree leaf order, using ppxgboost's model and query encrypting functions. Once the tree structure and our test queries are encrypted, we obtain predictions from the encrypted tree by feeding encrypted test queries, allowing for the evaluation of our model with novel encrypted test data in a privacy-preserving manner. It is important to remember that since test queries are encrypted and all arithmetic operations are also done between encrypted numbers, the resulting predictions are also encrypted. So, we further decrypted these encrypted predictions using ppgxboost's Paillier decryption function and calculated the performance metrics to compare with the predictions coming from non-encrypted evaluation. Since encrypted arithmetic operations can introduce precision noise due to approximation errors, we split our dataset with 5-Fold and ran the same algorithm five times to average the results and minimize noise.

5.3.2 LR evaluation with TenSEAL. To evaluate our 1-layer-NN LR models with a privacy-preserving approach, we used TenSEAL [Benaissa et al. 2021]. TenSEAL is a library for doing homomorphic encryption operations on tensors and is built on top of Microsoft's popular SEAL package. It utilizes the Cheon-Kim-Kim-Song (CKKS) scheme to perform operations on vectors of complex values, leveraging the rich structure of integer polynomial rings [Benaissa et al. 2021; Huynh 2021].

To test the privacy-preserving evaluation approach, we first trained and evaluated a 1-layer-NN model using PyTorch. Model only had a single linear layer (therefore acts like a logistic regression), and uses sigmoid function as its forward function to do predictions. We used the Stochastic Gradient Descent as an optimizer with a learning rate of 0.5. We also used Binary Cross Entropy as our evaluation criteria. To train the model, we used an initial 70/30 split of the data for training and testing. We also split the dataset 10-fold and ran the same algorithm 10 times to average the results and minimize precision noise that can be added from arithmetic operations.

For the encrypted evaluation, we used TenSEAL to create a PyTorch-like model that computes the forward method as $X_{enc} \cdot W + B$ where X_{enc} is the encrypted test data, W corresponds to the weights and B the biases. The weights and biases are set to the ones learned from the LR trained on non-encrypted data. To encrypt the test queries, we used the CKKS encryption scheme with a polynomial modulus degree of 4096, coefficient bit sizes [40, 20, 20], and a scale of 2^{20}. We then decrypt the output using TenSEAL [Benaissa et al. 2021] and compare the ground truth test values. As a second additional privacy-preserving approach, we further encrypted the W and B, the weights and biases learned from LR trained on non-encrypted data that were originally not encrypted. Hence, it made the model's forward method as $X_{enc} \cdot W_{enc} + Benc$ where X_{enc} is the encrypted test data, W corresponds to the encrypted weights and B the encrypted biases. We again ran and evaluated these models 10-fold and compared them with ground truth test values to get metrics and examine encrypted evaluation's applicability.

Finally, since the performance metrics from our LR model with one linear layer were low, we further resampled from our combined training and test dataset to ensure an equal number of samples from the two classes in our target variable. We randomly sampled 6198 observations from our data (3099 alive and 3099 deceased labels), then performed a train-test split as 70/30. We then ran our plain evaluation, encrypted test evaluation, and encrypted test with encrypted weight evaluation 10 times to average the resulting metrics and eliminate the effect of precision noise when comparing performance metrics.

5.4 Encrypted Training

In order to train a model using encrypted data, we needed our own algorithm for the forward and backward propagation of the model. Part of our initial research was doing the mathematics required to define this for any sort of Multi-Layer Neural Network, although in the end, we only trained a single-layer network.

In the forward propagation phase, each layer computes the output based on the input from the previous layer using a linear transformation followed by a non-linear activation function. However, in the case of homomorphic encryption, there is only one effective activation layer that can be used, a polynomial approximation to a sigmoid function. The equations are given by:

$$z_l = W^{(l)} a_{(l-1)} + b^{(l)}, \quad \text{for } l = 1, 2, \ldots, L, \tag{1}$$

$$a_{(l)} = \sigma(z_{(l)}), \quad \text{for } l = 1, 2, \ldots, L, \tag{2}$$

$$\hat{y} = a_{(l)}, \tag{3}$$

$$\sigma(x) = 0.5 + 0.197x - 0.004x^3 \tag{4}$$

Where:

- $z^{(l)}$ is the linear combination at the l-th layer,
- $W^{(l)}$ is the weight matrix for the l-th layer,
- $a^{(l-1)}$ is the activation from the previous layer (with $a^{(0)}$ being the input to the network),
- $b^{(l)}$ is the bias vector for the l-th layer,
- σ is the polynomial approximation to the sigmoid function used as an activation function.

For the Loss function, we use the binary cross entropy loss function as shown:

$$\mathcal{L}(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right] \tag{5}$$

$$+ \frac{\lambda}{2m} \sum_{j=1}^{L} \left[||W^{(l)}||_{2,2}^2 + |b^{(l)}|_2^2 \right] \tag{6}$$

.

For the purposes of encrypted training, however, we never actually compute the loss function since log functions are not calculable using Homomorphically Encrypted values. We ended up using this loss function because its derivative is quite easy to calculate when combined with the sigmoid activation layer. Additionally, we set

λ to be equal to $0.05m$ as a regularization parameter. Furthermore, while we used an approximation to the sigmoid function in the forward propagation, we do not actually use the gradient of the approximation but instead use the gradient formula of the actual sigmoid function [Chen et al. 2018]. Thus, when using this, our backpropagation becomes:

$$\frac{\partial \mathcal{L}}{\partial W^{(L)}} = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(}i))a_{(L-1)}^{(i)} + 0.05W^{(L)}, \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial b^{(L)}} = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(}i)) + 0.05b^{(L)}, \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial a_{(L-1)}} = (y - \hat{y})W^{(L)\top}, \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial z_{(L-1)}} = \frac{\partial \mathcal{L}}{\partial a^{(L-1)}} \times \frac{\partial a_{(L-1)}}{\partial z_{(L-1)}}, \tag{10}$$

$$= \frac{\partial \mathcal{L}}{\partial a^{(L-1)}} \times (a_{(L-1)}(1 - a_{(L-1)})) \tag{11}$$

From here, you go back to your standard ML backpropagation algorithm as follows:

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_{(l)}}(a_{(l-1)})^{\top} + 0.05W^{(l)}, \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial b^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_{(l)}} + 0.05b^{(l)}, \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial a_{(l-1)}} = \frac{\partial \mathcal{L}}{\partial z_{(l)}}(W^{(l)})^{\top}, \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial z_{(l-1)}} = \frac{\partial \mathcal{L}}{\partial a^{(l-1)}} \times (a_{(l-1)}(1 - a_{(l-1)})) \tag{15}$$

Which you recurse over until you reach $l = 0$
Where:

- $\mathcal{L}$ is the loss function of the network,
- $g'^{(l)}$ is the derivative of the activation function at layer l,
- $(W^{(l)})^{\top}$ is the transpose of the weight matrix of the l-th layer.

Our code implemented this in two formats: The first is for a single-layer Neural Network, which only needs vector arithmetic as given in Equations 1-11. The second is for multi-layer neural networks, which require tensor arithmetic as shown above from equations 12-15. As discussed above we were not able to run that on our entire database due to the compute requirements for tensor arithmetic, but we were able to test its accuracy on smaller slices of the dataset. Beyond this, we chose to use a learning rate of 0.15 with stochastic gradient descent. Overall, this was the algorithm that we implemented using encrypted training data and models.

REFERENCES

Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. 2021. TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption. arXiv:2104.03152 [cs.CR]

Alexandra Boldyreva, Nathan Chenette, and Adam O'Neill. 2011. Order-Preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings (Lecture Notes in Computer Science, Vol. 6841)*, Phillip Rogaway (Ed.). Springer, 578–595. https://doi.org/10.1007/978-3-642-22792-9_33

Sergiu Carpov, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev. 2022. GenoPPML–a framework for genomic privacy-preserving machine learning. In *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*. IEEE, 532–542.

Hao Chen, Ran Gilad-Bachrach, Kyoohyung Han, Zhicong Huang, Amir Jalali, Kim Laine, and Kristin Lauter. 2018. Logistic regression over encrypted data from fully homomorphic encryption. *BMC medical genomics* 11 (2018), 3–12.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM. https://doi.org/10.1145/2939672.2939785

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. *Homomorphic Encryption for Arithmetic of Approximate Numbers*. Springer International Publishing, 409–437. https://doi.org/10.1007/978-3-319-70694-8_15

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. 2020. TFHE: fast fully homomorphic encryption over the torus. *Journal of Cryptology* 33, 1 (2020), 34–91.

Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. 2022. On the precision loss in approximate homomorphic encryption. *IACR Cryptol. ePrint Arch.* (2022), 162. https://eprint.iacr.org/2022/162

Thomas Davenport and Ravi Kalakota. 2019. The potential for artificial intelligence in healthcare. *Future Healthcare Journal* 6, 2 (June 2019), 94–98. https://doi.org/10.7861/futurehosp.6-2-94

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. 2019. A guide to deep learning in healthcare. *Nature Medicine* 25, 1 (Jan. 2019), 24–29. https://doi.org/10.1038/s41591-018-0316-z

David Froelicher, Juan R. Troncoso-Pastoriza, Jean Louis Raisaro, Michel A. Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. 2021. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature Communications* 12, 1 (Oct. 2021). https://doi.org/10.1038/s41467-021-25972-y

Craig Gentry. 2010. Computing arbitrary functions of encrypted data. *Commun. ACM* 53, 3 (March 2010), 97–105. https://doi.org/10.1145/1666420.1666444

Seungwan Hong, Jai Hyun Park, Wonhee Cho, Hyeongmin Choe, and Jung Hee Cheon. 2022. Secure tumor classification by shallow neural network using homomorphic encryption. *BMC Genomics* 23, 1 (April 2022). https://doi.org/10.1186/s12864-022-08469-w

Daniel Huynh. 2021. CKKS explained: Part 1, Vanilla Encoding and decoding. https://blog.openmined.org/ckks-explained-part-1-simple-encoding-and-decoding/

Konstantina Kourou, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V. Karamouzis, and Dimitrios I. Fotiadis. 2015. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal* 13 (2015), 8–17. https://doi.org/10.1016/j.csbj.2014.11.005

Minhyeok Lee. 2023. Deep Learning Techniques with Genomic Data in Cancer Prognosis: A Comprehensive Review of the 2021–2023 Literature. *Biology* 12, 7 (June 2023), 893. https://doi.org/10.3390/biology12070893

Xiaoxuan Liu, Ben Glocker, Melissa M McCradden, Marzyeh Ghassemi, Alastair K Denniston, and Lauren Oakden-Rayner. 2022. The medical algorithmic audit. *The Lancet Digital Health* 4, 5 (May 2022), e384–e397. https://doi.org/10.1016/s2589-7500(22)00003-6

Xianrui Meng and Joan Feigenbaum. 2020. Privacy-Preserving XGBoost Inference. https://doi.org/10.48550/ARXIV.2011.04789

Kundan Munjal and Rekha Bhatia. 2022. A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex Intelligent Systems* 9, 4 (May 2022), 3759–3786. https://doi.org/10.1007/s40747-022-00756-z

Blake Murdoch. 2021. Privacy and artificial intelligence: challenges for protecting health information in a new era. *BMC Medical Ethics* 22, 1 (Sept. 2021). https://doi.org/10.1186/s12910-021-00687-3

Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques (Lecture Notes in Computer Science, Vol. 1592)*. Springer, 223–238. https://doi.org/10.1007/3-540-48910-X_16

Roberta Pastorino, Corrado De Vito, Giuseppe Migliara, Katrin Glocker, Ilona Binenbaum, Walter Ricciardi, and Stefania Boccia. 2019. Benefits and challenges of Big Data in healthcare: an overview of the European initiatives. *European Journal of Public Health* 29 (October 2019), 23–27. https://doi.org/10.1093/eurpub/ckz168

Esha Sarkar, Eduardo Chielle, Gamze Gursoy, Leo Chen, Mark Gerstein, and Michail Maniatakos. 2023. Privacy-preserving cancer type prediction with homomorphic encryption. *Scientific Reports* 13, 1 (Jan. 2023). https://doi.org/10.1038/s41598-023-28481-8

James Scheibner, Marcello Ienca, and Effy Vayena. 2022. Health data privacy through homomorphic encryption and distributed ledger computing: an ethical-legal qualitative expert assessment study. *BMC Medical Ethics* 23, 1 (Dec. 2022). https://doi.org/10.1186/s12910-022-00852-2

James Scheibner, Jean Louis Raisaro, Juan Ramón Troncoso-Pastoriza, Marcello Ienca, Jacques Fellay, Effy Vayena, and Jean-Pierre Hubaux. 2021. Revolutionizing Medical Data Sharing Using Advanced Privacy-Enhancing Technologies: Technical, Legal,

and Ethical Synthesis. *Journal of Medical Internet Research* 23, 2 (Feb. 2021), e25120. https://doi.org/10.2196/25120

Dan Sha, Zhaohui Jin, Jan Budczies, Klaus Kluck, Albrecht Stenzinger, and Frank A. Sinicrope. 2020. Tumor Mutational Burden as a Predictive Biomarker in Solid Tumors. *Cancer Discovery* 10, 12 (Dec. 2020), 1808–1825. https://doi.org/10.1158/2159-8290.cd-20-0522

Chen Song and Xinghua Shi. 2024. ReActHE: A homomorphic encryption friendly deep neural network for privacy-preserving biomedical prediction. *Smart Health* 32 (2024), 100469.

Effy Vayena, Joan Dzenowagis, John S Brownstein, and Aziz Sheikh. 2017. Policy implications of big data in the health sector. *Bulletin of the World Health Organization* 96, 1 (Nov. 2017), 66–68. https://doi.org/10.2471/blt.17.197426

Alexander Wood, Kayvan Najarian, and Delaram Kahrobaei. 2020. Homomorphic Encryption for Machine Learning in Medicine and Bioinformatics. *Comput. Surveys* 53, 4 (Aug. 2020), 1–35. https://doi.org/10.1145/3394658

Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. 2020. Federated Learning for Healthcare Informatics. *Journal of Healthcare Informatics Research* 5, 1 (Nov. 2020), 1–19. https://doi.org/10.1007/s41666-020-00082-4

Neel Yadav, Saumya Pandey, Amit Gupta, Pankhuri Dudani, Somesh Gupta, and Krithika Rangarajan. 2023. Data Privacy in Healthcare: In the Era of Artificial Intelligence. *Indian Dermatology Online Journal* 14, 6 (2023), 788–792. https://doi.org/10.4103/idoj.idoj_543_23

Xun Yi, Russell Paulet, and Elisa Bertino. 2014. *Homomorphic Encryption.* Springer International Publishing, 27–46. https://doi.org/10.1007/978-3-319-12229-8_2

Ahmet Zehir, Ryma Benayed, Ronak H Shah, Aijazuddin Syed, Sumit Middha, Hyunjae R Kim, Preethi Srinivasan, Jianjiong Gao, Debyani Chakravarty, Sean M Devlin, Matthew D Hellmann, David A Barron, Alison M Schram, Meera Hameed, Snjezana Dogan, Dara S Ross, Jaclyn F Hechtman, Deborah F DeLair, JinJuan Yao, Diana L Mandelker, Donavan T Cheng, Raghu Chandramohan, Abhinita S Mohanty, Ryan N Ptashkin, Gowtham Jayakumaran, Meera Prasad, Mustafa H Syed, Anoop Balakrishnan Rema, Zhen Y Liu, Khedoudja Nafa, Laetitia Borsu, Justyna Sadowska, Jacklyn Casanova, Ruben Bacares, Iwona J Kiecka, Anna Razumova, Julie B Son, Lisa Stewart, Tessara Baldi, Kerry A Mullaney, Hikmat Al-Ahmadie, Efsevia Vakiani, Adam A Abeshouse, Alexander V Penson, Philip Jonsson, Niedzica Camacho, Matthew T Chang, Helen H Won, Benjamin E Gross, Ritika Kundra, Zachary J Heins, Hsiao-Wei Chen, Sarah Phillips, Hongxin Zhang, Jiaojiao Wang, Angelica Ochoa, Jonathan Wills, Michael Eubank, Stacy B Thomas, Stuart M Gardos, Dalicia N Reales, Jesse Galle, Robert Durany, Roy Cambria, Wassim Abida, Andrea Cercek, Darren R Feldman, Mrinal M Gounder, A Ari Hakimi, James J Harding, Gopa Iyer, Yelena Y Janjigian, Emmet J Jordan, Ciara M Kelly, Maeve A Lowery, Luc G T Morris, Antonio M Omuro, Nitya Raj, Pedram Razavi, Alexander N Shoushtari, Neerav Shukla, Tara E Soumerai, Anna M Varghese, Rona Yaeger, Jonathan Coleman, Bernard Bochner, Gregory J Riely, Leonard B Saltz, Howard I Scher, Paul J Sabbatini, Mark E Robson, David S Klimstra, Barry S Taylor, Jose Baselga, Nikolaus Schultz, David M Hyman, Maria E Arcila, David B Solit, Marc Ladanyi, and Michael F Berger. 2017. Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10, 000 patients. *Nature Medicine* 23, 6 (May 2017), 703–713. https://doi.org/10.1038/nm.4333

Jian Zhou, Francisco Sanchez-Vega, Raul Caso, Kay See Tan, Whitney S. Brandt, Gregory D. Jones, Shi Yan, Prasad S. Adusumilli, Matthew Bott, James Huang, James M. Isbell, Smita Sihag, Daniela Molena, Valerie W. Rusch, Walid K. Chatila, Natasha Rekhtman, Fan Yang, Marc Ladanyi, David B. Solit, Michael F. Berger, Nikolaus Schultz, and David R. Jones. 2019. Analysis of Tumor Genomic Pathway Alterations Using Broad-Panel Next-Generation Sequencing in Surgically Resected Lung Adenocarcinoma. *Clinical Cancer Research* 25, 24 (Dec. 2019), 7475–7484. https://doi.org/10.1158/1078-0432.ccr-19-1651

Wan Zhu, Longxiang Xie, Jianye Han, and Xiangqian Guo. 2020. The Application of Deep Learning in Cancer Prognosis Prediction. *Cancers* 12, 3 (March 2020), 603. https://doi.org/10.3390/cancers12030603