Privacy-Preserving Federated Inference for Genomic Analysis with Homomorphic Encryption

Anish Chakraborty and Nektarios Georgios Tsoutsos

University of Delaware
{anishch, tsoutsos}@udel.edu

Abstract. In recent years, federated learning has gained significant momentum as a collaborative machine learning approach, particularly in the field of medicine. While the decentralized nature of federated learning boasts greater security guarantees compared to traditional machine learning methods, it is still susceptible to myriad attacks. Moreover, as federated learning becomes increasingly ubiquitous in medicine, its use for classification tasks is expected to increase; however, maintaining patient data confidentiality remains a significant challenge, especially for genetic data. In this work, we introduce a novel framework for secure federated inference on nucleotide-based genotype data and provide a gateway to private inference through fully homomorphic encryption. A federated model with five local clients was created and trained before being encrypted with the TFHE cryptosystem and placed for inference. This framework successfully identified promoter sequences encoded within given DNA sequences, showing its potential applications in secure genomic data analysis in a federated context. Our work represents a crucial step in privacy-preserving federated inference on nucleotide-based data.

Keywords: Federated Learning · Fully Homomorphic Encryption · Genomics · Privacy-Preserving Machine Learning.

1 Introduction

Federated learning provides a massive advantage for collaborative model training and inference, while also maintaining data security and privacy [21]. While medical data is often abundant, it is often spread across multiple data centers. In these cases, a decentralized, federated model can outperform conventional, centralized machine learning models. Furthermore, this architecture boasts the ability to maintain the confidentiality of data, as local models are trained separately with their private data before the model's weights are aggregated to form a global model. This ensures that any confidential data is secure, even when being used to train a model with potential adversaries.

Although these models have numerous security benefits compared to a conventional centralized machine learning system, they remain vulnerable to security breaches. During inference, malicious adversaries may gain access to data as it is being transferred to the global model [19], severely hindering privacy guarantees with classified data. For instance, a man-in-the-middle attack (MITM), in which a malicious adversary secretly gains access to the data being sent for inference depicts a key issue with federated models. Alternatively, the global server itself could be corrupted, therefore, sidestepping the security which a decentralized architecture provides; consequently, the entire architecture would grow unusable, as adversaries could gain access to private, confidential data [3].

Recent advancements have demonstrated the feasibility of utilizing fully homomorphic encryption (FHE) as a solution to privacy breaches in federated learning. FHE schemes call for the user to encrypt the data before sending it to a server to perform computations; subsequently, the server can run its algorithm on this encrypted data, and then send the encrypted output back to the user, who can then decrypt it. Therefore, this scheme prevents adversarial attacks during data transfer, as well as attacks from a corrupt global server. Therefore, in the case of an MITM attack, the data and results would be encrypted under a secret key known only to the user, meaning an adversary would be unable to decrypt the data as it is being transferred for inference.

Many FHE-based federated inference works focus on imaging data and overwhelmingly utilize CKKS as their FHE scheme [29, 30]. However, while CKKS allows floating-point computations, it focuses on approximate computations [6], which reduces precision. While with images, reduced precision in image reconstruction can still lead to optimal results (allowing CKKS to be sufficient) [5, 14], this lack of precision must be mitigated for meaningful results with nucleotide-based data. Our work surrounds the application of FHE-based federated learning to these nucleotide sequences to detect the presence of a specific promoter sequence; consequently, TFHE was used to allow for complete precision, for inaccuracies in a DNA sequence prove detrimental to determining its message. Furthermore, other related literature utilizes numeric gene expression data [4, 24] or numeric encoding vectors of Single Nucleotide Polymorphisms (SNPs) [8, 10, 22, 26], but these data types do not contain the same requisite sequential precision that nucleotide data intrinsically does.

Despite the clear lack of pertinent research regarding FHE-based federated inference schemes in nucleotide data, this topic proves extremely important. As DNA sequencing becomes more prevalent, major data centers continue to gain access to DNA strings to use for predictive analytics. In this work, we introduce a novel framework[1] (depicted in Figure 1) to perform federated classification tasks on DNA sequence data accurately. This work directly paves the way for future works in DNA classification, which can potentially be applied for disease detection, as well as a fuller understanding of the relationship between seemingly meaningless nucleotides and noticeable changes in phenotype. Finally, using the

[1] Available as open-source software at `https://github.com/AnishC10/PPFI-DNA.git`

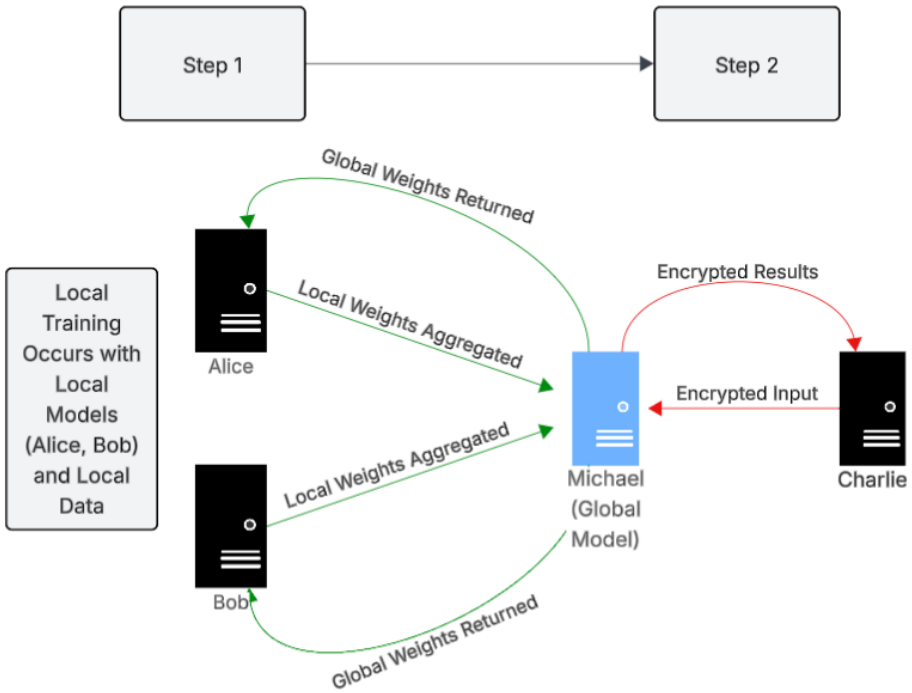

Fig. 1. Schematic of our proposed methodology. Step 1: Local models train on their respective local data, and perform aggregation with the global model. Step 2: Users can perform inference on this global model, and the outputs are protected through FHE.

TFHE encryption scheme, we ensure maximum precision with our data. Overall, our contributions can be summarized by the following:

- We reveal a novel federated framework which can accurately perform inference on individual nucleotide string data, allowing hospitals and medical data centers to utilize genotype data in predictive analytics;
- We ensure complete data security during private inference without significantly compromising on precision by utilizing the TFHE scheme.

Roadmap: The rest of the paper is organized as follows. Section 2 provides necessary preliminaries on homomorphic encryption, federated learning, and genetics. Section 3 discusses our proposed methodology. Section 4 presents our experimental results, as well as a brief discussion on the implications and limitations of our approach. Finally, Section 5 provides comparisons to related works, while Section 6 imparts our concluding statements.

2 Preliminaries

2.1 Fully Homomorphic Encryption (FHE)

Homomorphic Encryption (HE) is an encryption method that supports secure computation on encrypted data without first decrypting it. Specifically, modern HE systems can be split into three distinct subcategories: partial homomorphic encryption, leveled homomorphic encryption, and fully homomorphic encryption. Partial homomorphic encryption (PHE) refers to cryptosystems which allow for either multiplication or addition between two ciphertexts [16]. Cryptosystems such as RSA [25], and Paillier [23] comprise notable examples of PHE. However, despite being able to perform unbounded additions or multiplications, PHE's inability to do both severely hinders its potential usefulness. Moreover, leveled HE (LHE) supports addition *and* multiplication on certain ciphertexts; however, this system is inefficient with larger algorithm depths. To ensure security, a small noise component is added to encrypted ciphertexts; consequently, repeated computations result in an aggregation of noise, which prevents the correct decryption of the ciphertext [12]. Intuitively, homomorphic addition results in linear increases of noise, while homomorphic multiplication results in exponential increases in noise. In 2009, Charles Gentry proposed the first implementation of a fully homomorphic encryption (FHE) scheme, which notably included a *bootstrapping* mechanism that significantly reduces noise levels [11]. Since the noise component is the key facet to maintain security, bootstrapping helps ensure that noise is added without becoming overwhelming.

To elaborate, a bootstrapping mechanism calls for a decryption in the encrypted domain, which reduces the total amount of noise over a set of computations. However, the plaintext data is never exposed, as this computation happens completely in another level of encryption. Therefore, after applying the decryption algorithm homomorphically, the underlying message is not revealed, yet the amount of noise is reduced significantly. This creation of a bootstrapping mechanism intrinsically allows for FHE-based algorithms to run with unbounded depth; however, this bootstrapping method remains computationally intensive.
TFHE and CKKS: TFHE and CKKS remain two of the most commonly used cryptosystems in FHE, especially for neural network inference. The CKKS cryptosystem can perform floating-point computations, but this leads to approximate computations [6]. On the contrary, TFHE is another popular FHE scheme that solely performs integer arithmetic and does not utilize approximations in its calculations [7]. In our use case of specific nucleotide data, which is highly sensitive, we utilized the TFHE cryptosystem to ensure maximum precision. Our work, therefore, evaluates the feasibility of encrypted federated learning on DNA-based genomic data using TFHE.

2.2 Federated Learning

Federated learning constitutes a paradigm of computation and inference among multiple parties [21], and offers a decentralized approach to machine learning.

Unlike conventional "centralized" machine learning, which has one sole node for training and updating weights, FL calls for multiple nodes of local networks, all of which train on separate data. Then, the gradients of these local models are aggregated onto one global model, which, consequently, has "knowledge" from the entirety of the local data. This decentralized approach allows for medical companies to utilize potentially sensitive private patient data without disclosing it to rival companies or outsiders [9].

Despite federated learning offering a gateway to secure neural network computation with multiple parties, data must still be kept confidential when being sent to the models as well as during aggregation. Therefore, FHE has been deemed a valuable asset to ensure privacy in sensitive data during federated learning and inference [3], as this can prevent the prevalence of *data leakage* during model inference.

2.3 A Primer on DNA

DNA is the most fundamental building block in all living systems. Despite its ubiquity in all biological systems, its meaning proves elusive. Computational methods can be used to find patterns among DNA sequences. Given DNA's importance, documenting it is often spread across many different methods. Some methods utilize numeric data regarding the prevalence of a certain gene, while other methods highlight specific single-nucleotide combinations (called single-nucleotide polymorphisms) which are known to have some significance, and use this data to generate regression models. However, DNA nucleotide data remains more unexplored. This type of data results from DNA sequencing, which determines the exact combination of nucleotides that make up a given strand of DNA [15]. Our work leverages this data for decentralized predictive analytics.

2.4 Promoters and Their Importance

Promoters are a key part of DNA, as they help regulate transcription of RNA molecules. This concept is rooted in the central dogma of biology: for a protein to form, DNA instructions provide a blueprint for creating RNA sequences. This process is known as transcription. These RNA molecules can then undergo translation, which creates amino acids that later fold into proteins from an RNA blueprint. Therefore, this set of processes proves paramount for most biological organisms, describing the main transfer of information from DNA to build proteins. However, RNA polymerase, the molecule that creates RNA, needs to identify a promoter sequence within the DNA which is where it starts building the RNA. Finding these promoter sequences in a given strand of DNA paves the way towards understanding transcription. Furthermore, the processes used for classifying promoters can be further extended to classifying other parts of DNA, which can help with disease detection, as well as develop a fuller understanding of DNA.

2.5 Threat Model

Our methodology aims to protect user data privacy while creating a gateway for neural network inference on encrypted nucleotide data. We assume an *honest-but-curious (HBC)* cloud that performs proper operations on encrypted data sent for inference, but is incentivized to extract as much information as possible from these operations. In addition, we consider the case of leakages during data transfer, the prevalence of which proves severely detrimental for plaintext federated inference.

3 Our Proposed Methodology

In our proposed framework, we assume that there are multiple local clients, and they train a global model with federated learning. Then, a user intends to send a DNA sequence to be evaluated homomorphically by the model. The global model receives and performs computations on this encrypted data, and then returns the results to the user, who can then decrypt them. Privacy and precision are both ensured through the use of the TFHE cryptosystem.

Fundamentally, all DNA sequences contain four nucleotides: Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). A DNA sequence, therefore, can be expressed with a string with some combination of the letters [A,T,C,G], and this string serves as the elementary genetic code. However, these DNA sequences can be extraordinarily long; consequently, understanding the meaning behind a string of DNA proves extraordinarily difficult. Moreover, with regard to machine learning, data centers often have limited amounts of data, making it difficult to train a single centralized machine learning model; therefore, a federated framework is a more effective framework for genomic data.

DNA proves to be a more challenging than previously accepted forms of data. Related works mainly used either numeric data, or data from SNPs, which are short strands of known variation in DNA. Meanwhile, the methodology proposed in this work targets longer strands of DNA (with our dataset having over 50 nucleotides per strand of DNA). Moreover, as mentioned before, this work takes into account the sequential aspect which analyzing long strands of DNA entails. Moreover, while certain related works which use both numeric and SNP data use logistic or linear regression [4, 8, 10], a classification model was used in this framework to ensure non-linear pattern recognition.

3.1 Data Preprocessing

The DNA data[2] is given in the form of a string with some combination of [A,T,C,G]. We note that no synthetic data was generated to increase the size of the dataset, since given the cryptic nature of DNA, any synthetic generation could defy biological conventions. Furthermore, this limited dataset size highlights the low availability of public DNA data. Though there exist similar

[2] The UC Irvine Molecular Biology Dataset (n=106) was used in this work [13].

genomic datasets which are significantly larger, these generally target Single Nucleotide Polymorphisms or solely report known knowledge about a given strand of DNA. Conversely, our work requires multiple strands of labeled DNA sequenced data, from which we can discern the correlation between the nucleotides in the strand and the presence of a promoter sequence. To our knowledge, there exist no other larger or alternative datasets for this use case. To preprocess the data, one-hot encoding was used, in which each nucleotide correlated to a set of numbers; then, the encoded data was placed in an array. The data was then appropriately shuffled and split into training and testing sets. Then, the overall data was split randomly across five local clients to serve as local training data. We ensured that the local models solely had access to their local data to simulate an authentic use case for federated learning.

3.2 Federated Framework and Training

The local classification models were built using Concrete ML, and we used three total layers. The input layer contained 228 input neurons, the middle (hidden) layer contained 32 neurons, and the output layer included 2 neurons. The federated classification training phase is performed in two steps: first, the local models are trained for a specified amount of time on the local data. Then, the local models update the global model through weight aggregation by the federated average equation [21], which is expressed as follows:

$$w^{(t+1)} = \frac{1}{K} \sum_{k=1}^{K} w_k^{(t),} \tag{1}$$

where

- $w^{(t+1)}$: global model weights after iteration $t + 1$,
- $w_k^{(t)}$: model weights of client k after local training in iteration t,
- K: number of clients.

It should be noted that this equation is simplified on the assumption that the amount of data that each client receives is equal to the other clients; if the amount of data received was different, the formula would simply reflect a weighted average, rather than a normal average.

After the global model receives local weight updates, the parameters of the global model are redistributed to the local models, which then use those parameters in the next iteration. This continues for a specified number of global iterations. We note that the ReLU activation function [1] was specifically used due to its lower computational cost during the future inference phase compared to other common activation functions such as the Sigmoid or TanH functions.

3.3 Inference and Prediction

Fundamentally, inference occurs after the training and model weight compilation are finished. After a specified number of epochs, the training of the global

model is complete. As training continues (as mentioned before), the weights are compiled to the global model, and the local models are updated with the new compiled weights. Therefore, after training is complete, all local clients will have a fully trained global model from a diverse dataset, even if they contributed a small amount of data on their own. This paradigm addresses the issue of low amounts of data within local clients; however, when data is compiled within multiple clients, a strong federated machine learning model can be created without divulging the data.

FHE Implementation Details: For the model prediction phase, the Concrete ML library [32] (version 1.9.0) for the TFHE cryptosystem was used. This framework is a Python wrapper for the TFHE scheme [7], which was originally written in Rust. For our methodology, we built on the built-in neural networks that can predict on encrypted data from Concrete ML. Specifically, the model is trained in plaintext, then is compiled for FHE using the Concrete ML built-in function. The weights and biases were quantized at an 8 bit-width, the activation function was approximated using polynomials, and this allows the global model to be compiled and ready for inference on encrypted data.

The prediction phase of this framework was designed with the practical use case in mind: a client who intends to utilize our global model for classification, but wants to keep the DNA data private. Therefore, FHE is used to encrypt the data before it goes into the global model for inference. The model is specifically designed to perform calculations on encrypted data and return the output. Since the data is encrypted homomorphically, the global server will have no access to it, preserving the data's privacy and preventing data transfer attacks. The inference process is summarized further in Algorithm 1. First, the user submits their encrypted data (depicted as δ, with the *nth* data strand being denoted as D_E^n), which they encrypted using their secret key. Then, the global model (which can perform computations homomorphically), performs inference and returns the result to the user, who can then use their key to decrypt back to plaintext. We note that *FHEeval* refers to the process of sending encrypted data to the global model for inference.

Algorithm 1 Prediction with FHE

Require: Encrypted data $\delta = (D_E^1, D_E^2, D_E^3 \ldots, D_E^n)$, FHE Circuit C
Ensure: The decrypted results from homomorphic inference
1: Initialize list *results*
2: **for** each $D_E^i \in \delta$ **do**
3: $\quad$ *results*$[i] \leftarrow$ FHEeval(C, D_E^i)
4: **end for**
5: **if** permission is given **then** ▷ Done automatically during inference
6: $\quad$ User uses private keys K to decrypt *results*
7: **end if**

During the model inference phase, the decryption was done automatically; however, in a real-world use case, the final *if-statement* would apply, since the user would need to utilize their private key to decrypt *results*.

4 Results and Discussion

This section presents the experimental data collected, along with details regarding the hardware used in our methodology. First, all of our experiments were performed on an Amazon EC2 c7i.8xlarge instance, which contained 32 vCPUs and 64 GiB of memory. We utilized the Concrete ML framework for generating a FHE circuit, and utilized the default security parameters, ensuring 128 bits of security.

The scalability of our framework depends on the parameters of the framework, as noise growth of ciphertexts may present a larger issue as the number of clients or the length of data strings increase. While classical machine learning models result in a linear increase in complexity as input size increases, our framework contains a privacy-preserving facet, which causes noise to increase more quickly. As the input size of the data increases, the computational overhead increases at a significant rate, which presents a limitation of our work.

4.1 Experimental Evaluation

We developed a set of experiments that accurately captured the abilities of our methodology. We present the basic metrics (accuracy, F1 score, and runtime) in Table 1. Furthermore, we include the throughput of inference, which measures the amount of inputs our framework can process in a given unit time (a second), which is also shown in Table 1. These results display the clear accuracy and precision that our framework has, despite the large computational overhead that not only comes from encryption, but also the sequential awareness required to accurately make predictions on DNA nucleotide data. We note that although the throughput value appears low, this is justified by the significant computational overhead that stems from utilizing FHE during model inference. Moreover, we expect computation time to grow linearly as the number of input samples increases in a given dataset.

Table 1. Execution Times (s), Validation Accuracy(%), F1 Score, and Throughput (samples/second).

Execution Time	Validation Accuracy	F1 Score	Throughput (samples/s)
114.67	81.8	0.89	0.09

We compared our FHE-encrypted framework to its unencrypted counterpart to highlight the differences in accuracy and precision associated with additional

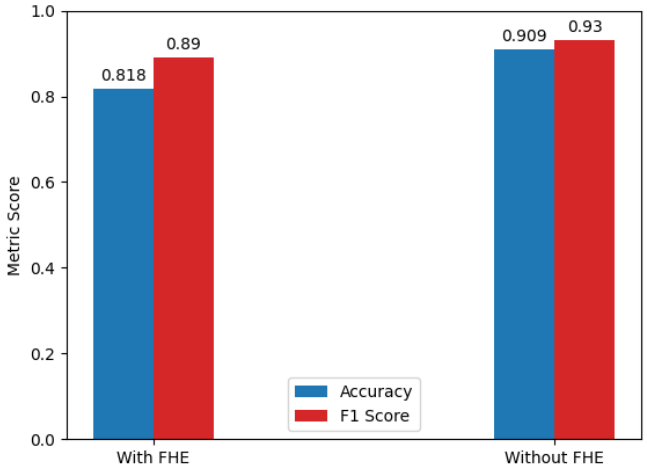

Fig. 2. Accuracy and F1 Score vs Model Type.

encryption. These results are illustrated in Figure 2, which shows that our encrypted framework achieves only slightly lower accuracy and precision,[3] despite the significant computational overhead added by FHE, demonstrating its distinct capability for deployment in real-world privacy-preserving federated classification.

Finally, we compare our findings to related works. To our knowledge, our framework is the first privacy-preserving decentralized framework for nucleotide data. The capabilities of our project compared to select related works are shown in Table 2. We note that given the novel nature of our framework, none of the related works that employ decentralized learning and fully homomorphic encryption can predict with nucleotide data.

4.2 Impact and Limitations

Our methodology demonstrates a novel approach to leveraging nucleotide data, which has become more ubiquitous in the field of biotechnology through DNA sequencing, in complex biomedical tasks. Our methodology not only utilizes the power of federated learning to generate a decentralized model that can be trained without sharing sensitive data but also capitalizes on the capabilities of fully homomorphic encryption to ensure data privacy during model inference. This approach allows for a truly privacy-preserving machine learning system,

[3] We remark that the reduced accuracy may come from the lower amount of testing data, leading to each sample holding a larger percentage value in the overall accuracy.

Table 2. Comparative Analysis of Privacy-Preserving Genomic Analysis Frameworks.

Proposed Framework	Federated Learning	Uses FHE	Predicts on DNA Nucleotides	Predicts on SNPs
This Work	✓	✓	✓	
Cho et al. [8]	✓	✓		✓
Froelicher et al. [10]	✓	✓		✓
Namazi et al. [22]	✓	✓		✓
Ananya et al. [2]			✓	
Umarov et al. [31]			✓	
Kapoor et al. [17]	✓		✓	

and to our knowledge, the first framework for predictions upon raw nucleotide data.

Our framework offers numerous advantages in the field of utilizing nucleotide data in machine learning contexts, as it creates an optimized pathway to train a decentralized model and perform encrypted computations without compromising on accuracy. However, some limitations of this project include its overall computational overhead. While our proposed framework offers promising accuracy, the use of FHE during prediction is impacting the inference throughput. Furthermore, in cases where leakage is possible during the training phase, our model may not be fully secure. Therefore, future extensions may focus on the use of differential privacy during the training phase to further reinforce the security of a privacy-preserving federated model. Another future direction can investigate utilizing an LSTM architecture for the local and global models, since these models are specifically calibrated to learn from sequential data [20]. However, the complex architecture of LSTMs may result in a significantly slower execution time compared to our methodology. Moreover, these additional computations may introduce noise, which can compromise the model's accuracy.

5 Related Works

In this section, we discuss previous works in the intersection between federated learning, fully homomorphic encryption, and medical data. These related works can be separated into three main classes: works focusing on imaging data, works focusing on numeric gene expression data, and works focusing on single-nucleotide polymorphisms. We discuss the capabilities of these works and compare them to our methodology.

While some related works have used DNA data for classification [2, 17, 31], these works do not incorporate fully homomorphic encryption during prediction, leaving privacy as a key concern. These works are omitted from the three prevailing classes that we focus on in this work.

5.1 Class 1: Imaging Data

Multiple previous works exist regarding using FHE and FL schemes on imaging data [18, 29, 28, 27, 30]. These works mainly utilize the CKKS FHE framework for their computations, which utilizes an approximation-based approach to encryption. This approach increases the number of potential computations; however, it leads to reduced precision when encrypting input data. However, while images can be accurately represented despite minor pixel inaccuracies [5, 14], DNA has an exceptionally low noise tolerance, since one small nucleotide change can result in a completely different output. Also, DNA sequences contain a sequential specificity that is intrinsic to determining their meaning, which images do not contain. This specificity requires added precision when encoding and performing operations on the data compared to images. Therefore, compared to these works, our methodology offers the same privacy guarantees during training (due to the decentralized nature of our framework) and inference (with 128 bits of security from FHE), but also (as shown by Table 1) provides accurate predictions without compromising the data.

5.2 Class 2 and 3: Clinical Data and Single Nucleotide Polymorphisms

Many previous works focus on using numeric data to develop robust classification and regression models. For instance, Raisaro *et al.* [24] developed a framework which can predict off of clinical patient data, as well as from numeric gene expression data regarding the position of a mutated allele in a specific chromosome. This allowed them to develop a cancer prediction framework on both numeric data and clinical data. Similarly, Carpov *et al.* [4] used numeric gene expression data as well to develop a regression model. However, neither of these data types require sequential specificity, which makes them inefficient for DNA nucleotide data. While these related frameworks prove more holistic, they fail to predict on data with as much sequential specificity as DNA nucleotide data, which is the data type targeted in our framework.

Moreover, multiple related works utilize SNP data in their privacy-preserving frameworks [8, 10, 22, 26]. Single-nucleotide polymorphisms are specific nucleotides that have been altered in a given DNA sample. However, the data is often stored as either the few nucleotides that have been altered or as a numeric vector of the former. These mutations allow for predictive analytics regarding certain diseases and prove extremely useful to find the link between a given mutation and predisposition to a disease. However, they also lack the emphasis on sequence that our model captures in nucleotide data. Although SNPs prove extremely useful, they are limited in scope; our work operates on DNA sequences that are much broader in scope and have more overall meaning to uncover.

6 Conclusion

In this work, we present a novel framework for privacy-preserving decentralized machine learning with nucleotide data. Previous works developed similar models

for imaging and numeric-based data; however, this ignores the rising prevalence of nucleotide-based data. In contrast to images and numeric data, nucleotide data is sequentially sensitive, as one misrepresentation can lead to a completely different output from the DNA. By utilizing the TFHE cryptosystem to ensure precise data encryption, we can create an accurate framework that can be deployed in real-world contexts. Furthermore, we demonstrate that despite a limited dataset size, we were able to train a highly fine-tuned model that can accurately perform encrypted inference on previously unseen data.

As DNA sequencing gains popularity in the field of biotechnology, predictive analytics on nucleotide data proves notably more important. Our framework depicts the ability to embrace this new method of capturing DNA and engage with it in a meaningful way. Our methodology depicts the ability to use DNA information to accurately decrypt its hidden code, which can in the future be applied to disease detection and further understand the meaning behind a given DNA strand.

References

[1] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE]. URL: https://arxiv.org/abs/1803.08375.

[2] Namburi Ananya et al. "DNA Classification For Finding E.coli". In: *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. 2024, pp. 1762–1768. DOI: 10.1109/ICAAIC60222.2024.10575339.

[3] Nathalie Baracaldo and Hayim Shaul. *Federated learning meets Homomorphic encryption*. Jan. 2023. URL: https://research.ibm.com/blog/federated-learning-homomorphic-encryption.

[4] Sergiu Carpov et al. "GenoPPML – A framework for genomic privacy-preserving machine learning". In: *IEEE 15th International Conference on Cloud Computing (CLOUD)* (July 2022), pp. 532–542. DOI: 10.1109/cloud55607.2022.00076.

[5] Xintao Chai et al. "Deep learning for irregularly and regularly missing data reconstruction". In: *Scientific reports* 10.1 (2020), p. 3302.

[6] Jung Hee Cheon et al. "Homomorphic encryption for arithmetic of approximate numbers". In: *Lecture Notes in Computer Science* (2017), pp. 409–437. DOI: 10.1007/978-3-319-70694-8_15.

[7] Ilaria Chillotti et al. *Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds*. Feb. 2017. URL: https://eprint.iacr.org/2016/870.

[8] Hyunghoon Cho et al. "Secure and federated genome-wide association studies for biobank-scale datasets". In: *Nature Genetics* (2025), pp. 1–6.

[9] Prerna Dogra. *Federated learning with flare: Nvidia brings collaborative AI to healthcare and beyond*. Nov. 2022. URL: https://blogs.nvidia.com/blog/federated-learning-ai-nvidia-flare/.

[10] David Froelicher et al. "Truly privacy-preserving Federated Analytics for precision medicine with multiparty homomorphic encryption". In: *Nature Communications* 12.1 (Oct. 2021). DOI: 10.1038/s41467-021-25972-y.

[11] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing* (May 2009), pp. 169–178. DOI: 10.1145/1536414.1536440.

[12] Charles Gouert and Nektarios Georgios Tsoutsos. "Data Privacy Made Easy: Enhancing applications with Homomorphic encryption". In: *ACM Transactions on Design Automation of Electronic Systems* 30.3 (Feb. 2025), pp. 1–31. DOI: 10.1145/3715877.

[13] Reynolds Harley and Noordewier. *Molecular Biology (Promoter Gene Sequences)*. UCI Machine Learning Repository. 1987. DOI: 10.24432/C5S01D.

[14] Hossein Hosseini, Sreeram Kannan, and Radha Poovendran. "Dropping Pixels for Adversarial Robustness". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 91–97. DOI: 10.1109/CVPRW.2019.00017.

[15] National Human Genome Research Institute. *DNA Sequencing Fact Sheet*. URL: https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Fact-Sheet.

[16] Marc Joye. "Homomorphic Encryption 101". In: *Zama.ai* (Dec. 2021). URL: https://www.zama.ai/post/homomorphic-encryption-101.

[17] Ayshika Kapoor et al. "iProLSTM-FL: A Federated Learning Framework for Promoter Identification using LSTM Networks". In: *International Conference on Signal Processing, Computing and Control (ISPCC)*. 2025, pp. 221–226. DOI: 10.1109/ISPCC66872.2025.11039561.

[18] Xavier Lessage et al. "Secure federated learning applied to medical imaging with fully homomorphic encryption". In: *IEEE International Conference on AI in Cybersecurity (ICAIC)*. 2024, pp. 1–12. DOI: 10.1109/ICAIC60265.2024.10433836.

[19] Pengrui Liu, Xiangrui Xu, and Wei Wang. "Threats, attacks and defenses to Federated Learning: Issues, taxonomy and Perspectives". In: *Cybersecurity* 5.1 (Feb. 2022). DOI: 10.1186/s42400-021-00105-6.

[20] *Long Short-Term Memory (LSTM)*. URL: https://developer.nvidia.com/discover/lstm.

[21] H. Brendan McMahan et al. "Communication-efficient learning of Deep Networks from Decentralized Data". In: *arXiv.org* (Jan. 2016). URL: https://arxiv.org/abs/1602.05629.

[22] Mina Namazi et al. "Privacy-preserving framework for genomic computations via multi-key homomorphic encryption". In: *Bioinformatics* 41.3 (Jan. 2025), btae754. ISSN: 1367-4811. URL: https://doi.org/10.1093/bioinformatics/btae754.

[23] Pascal Paillier. "Public-key cryptosystems based on composite degree residuosity classes". In: *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT'99. Prague, Czech Republic: Springer-Verlag, 1999, pp. 223–238.

[24] Jean Louis Raisaro et al. "MEDCO: Enabling secure and privacy-preserving exploration of distributed clinical and Genomic Data". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.4 (July 2019), pp. 1328–1341. DOI: 10.1109/tcbb.2018.2854776.

[25] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (Feb. 1978), pp. 120–126. DOI: 10.1145/359340.359342.

[26] Md Nazmus Sadat et al. "SAFETY: Secure gwAs in Federated Environment through a hYbrid Solution". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.1 (2019), pp. 93–102. DOI: 10.1109/TCBB.2018.2829760.

[27] Dimitris Stripelis et al. "A federated learning architecture for secure and private neuroimaging analysis". In: *Patterns* 5.8 (2024).

[28] Dimitris Stripelis et al. "Secure federated learning for neuroimaging". In: *arXiv preprint arXiv:2205.05249* 11 (2022).

[29] Dimitris Stripelis et al. "Secure neuroimaging analysis using federated learning with homomorphic encryption". In: *17th International Symposium on Medical Information Processing and Analysis*. Ed. by Letícia Rittner et al. Vol. 12088. International Society for Optics and Photonics. SPIE, 2021. URL: https://doi.org/10.1117/12.2606256.

[30] Daniel Truhn et al. "Encrypted federated learning for secure decentralized collaboration in cancer image analysis". In: *Medical Image Analysis* 92 (2024), p. 103059. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2023.103059.

[31] Ramzan Umarov et al. "Promoter analysis and prediction in the human genome using sequence-based deep learning models". In: *Bioinformatics* 35 (Jan. 2019). DOI: 10.1093/bioinformatics/bty1068.

[32] Zama.ai. URL: https://docs.zama.ai/concrete-ml.