SECURE PRIVACY USING FEDERATED LEARNING TECHNIQUES

[1] Velpuri Sai Teja, [2] P. Sujatha, [3] G. Rajasekharam

1. Department of Computer Science and Engineering, MIRACLE EDUCATIONAL SOCIETY GROUP OF INSTITUTIONS, Vizianagaram

2. Department of Artificial Intelligence and Data Science, MIRACLE EDUCATIONAL SOCIETY GROUP OF INSTITUTIONS, Vizianagaram

3. Department of Computer Science and Engineering, MIRACLE EDUCATIONAL SOCIETY GROUP OF INSTITUTIONS, Vizianagaram

ABSTRACT: With innovation based on available data, the creation of machine learning systems that ensures user privacy is now more pressing than ever. Organizations have been utilizing centralized datasets for model training purposes which raises serious concerns about potential data leakage breaches, misuse as well as regulatory compliance. The project "Secure Privacy Using Federated Learning Techniques" attempts to solve these issues by employing an approach based on decentralization and privacy-preserving techniques. Providing critical defense mechanisms on sensitive information at the data and model level while ensuring performance and scalability are preserved is the objective of this model.

This particular project uses the paradigm of Federated Learning (FL), which is a form of collaborative machine learning where model training happens on various devices or servers in a decentralized manner, and raw data stays local. FL helps reduce privacy risk as it eliminates the need to transfer private data to central repositories. To make this approach stronger, the system incorporates advanced cryptographic methods such as Homomorphic Encryption, Differential Privacy, and Secure Multi-Party Computation (SMPC). These approaches provide encrypted calculations, data mask through noise addition as well as secure collective computations thereby protecting each individual data point from exposure during training agnostic multi-dimensional space process known as computation graph directed acyclic graphs,. Besides privacy preservation, model failure reasons are also addressed by this project

Key words: Secure Multi-Party Computation, Differential Privacy, Homomorphic Encryption, Federated Learning.

1. INTRODUCTION:

As machine learning (ML) technologies are adopted in diverse industries such as healthcare, finance, and even security, serious concerns have arisen about user data privacy as well as the security of machine learning models. As systems become more automated, it is critical to protect data confidentiality, integrity, and compliance during the entire lifecycle of machine learning. This project seeks to apply privacy-preserving approaches and protective policies to mitigate risks for both the data and models associated with them.

Artificial intelligence has a branch called Machine Learning which trains algorithms on large troves of data so they can subsequently make forecasts or decisions without clear directives. With the help of sophisticated algorithms use today,

ML models tell patterns in data or transition towards more sophisticated states based on historical information. Each model can be grouped according to level of their training processes and type of utilized data Within this

project framework, privacy issues emerge due to the collection of data and authorization of confidentiality information in training and inference. To mitigate these challenges, the project applies federated learning, differential privacy, homomorphic encryption, along with other strategies designed for secure ML systems while ensuring they are scalable and compliant.

1.1 Important Aspects:

1.1.1 Homomorphic Encryption: The project uses homomorphic encryption to process encrypted data without decrypting it first. This helps maintain confidentiality during the entire machine learning pipeline involving sensitive data.

1.1.2. Differential Privacy: Differential privacy is also integrated in order to protect records within a dataset. For this purpose, controlled noise will be injected into the data so that privacy preserving features can be balanced with its usefulness for answering questions of interest.

1.1.3. Secure Multi-Party Computation (SMPC): SMPC protocols allow joint computation on distributed datasets without any party having access to the unprocessed data.In this way, proper multi-party contribution from different model training participants is ensured for security purposes.

1.1.4. Federated Learning: It enables raw datato remain stored in decentralized devices and only models are exchanged, thus achieving privacy disrespect alongside collaborative learning which lessens dependence on central data hubs.

1.1.5. Model Watermarking: Watermarking methods are used in order to strengthen model security by embedding identifiable markers into the parameters of the models which aids in identifying unauthorized reproductions or modifications of the models.

3. Related Works:

No.	Title	Technique / Approach	Contribution	Dataset / Application	Year	Reference
1	Deep Learning with Differential Privacy	Differential Privacy (DP-SGD)	First DP method for deep learning; used moments accountant	MNIST, CIFAR-10	2016	Abadi et al., 2016
2	PATE: Private Aggregation of Teacher Ensembles	Ensemble Learning + Differential Privacy	Improves privacy via teacher-student models	MNIST	2017	Papernot et al., 2017
3	Homomorphic Encryption for Machine Learning	Fully Homomorphic Encryption (FHE)	Enables ML on encrypted data without decryption	Encrypted MNIST	2018	Bost et al., 2015

4	Federated Learning: Collaborative ML without Centralized Data	Federated Learning (FL)	Edge devices train shared model while preserving data locality	Mobile keyboard prediction	2017	McMahan et al., 2017
5	CryptoNets: Applying Neural Networks to Encrypted Data	Homomorphic Encryption + CNN	First encrypted inference with neural networks	MNIST	2016	Gilad-Bachrach et al., 2016
No.	Title	Technique / Approach	Contribution	Dataset / Application	Year	Reference
6	Privacy-Preserving Deep Learning using Secure Multiparty Computation (MPC)	Secure MPC	Enables joint learning without data leakage between parties	Custom datasets	2017	Shokri & Shmatikov , 2015
7	Opacus: Differential Privacy Library for PyTorch	DP-SGD in PyTorch	Provides easy-to-use DP training for DL models	Various datasets	2020	Facebook AI
8	DP-FedAvg: Privacy-Preserving Federated Averaging	DP + FL (DP-FedAvg)	Combines DP with Federated Averaging	Healthcare data, image data	2020	Geyer et al., 2017
9	SecureML: System for Secure Machine Learning	MPC + Garbled Circuits	Provides secure training and prediction pipeline	Private datasets	2017	Mohassel & Zhang, 2017

10	Survey of Privacy-Preserving Techniques in ML	Survey (Various Techniques)	Overview of DP, HE, MPC, FL in ML applications	—	2021	Ayoade et al., 2021

Table 1:- Literature Survey on Secure ML Techniques

4. IMPLEMENTATION STUDY:

Current privacy-preserving and secure machine learning (ML) systems use a variety of strategies to safeguard private information and guarantee the safety of ML models. Among the most important methods and strategies employed in current systems are:

4.1 Differential privacy is a technique that ensures that individual data points cannot be differentiated by adding noise to the input data. In addition to enabling insightful analysis of the data, this helps safeguard people's privacy.

4.2 Federated Learning: This decentralized method of machine learning involves training the model on several servers or devices that store local data samples without sharing them. Since the raw data never leaves the local device, this helps to secure the privacy of the data.

4.3 Homomorphic Encryption: This type of encryption enables calculations to be made on encrypted material without the need to decrypt it. This guarantees the privacy of sensitive data by keeping it encrypted during the calculation process.

5 Proposed Methodology

To address the drawbacks of existing Methods, we propose a novel approach that combines multiple privacy-preserving and secure machine learning (ML) techniques to achieve a balance between privacy, security, and utility. Our proposed system includes the following key components:

5.1 Hybrid Differential Privacy: We propose a hybrid differential privacy approach that combines the strengths of local and centralized differential privacy. Local differential privacy is used to add noise to individual data points, ensuring privacy at the data source. Centralized differential privacy is then applied to aggregate the noisy data, preserving privacy while maintaining data utility.

5.2 Adaptive Federated Learning: Our system includes an adaptive federated learning framework that dynamically adjusts the learning process based on the data distribution and model performance at the local devices. This helps mitigate the challenges of data heterogeneity and improves the overall efficiency and scalability of federated learning.

5.3 Secure Model Aggregation: To address the security risks associated with federated learning, we propose a secure model aggregation technique that ensures that the model updates from different devices are combined in a secure and verifiable manner, protecting against model poisoning attacks.

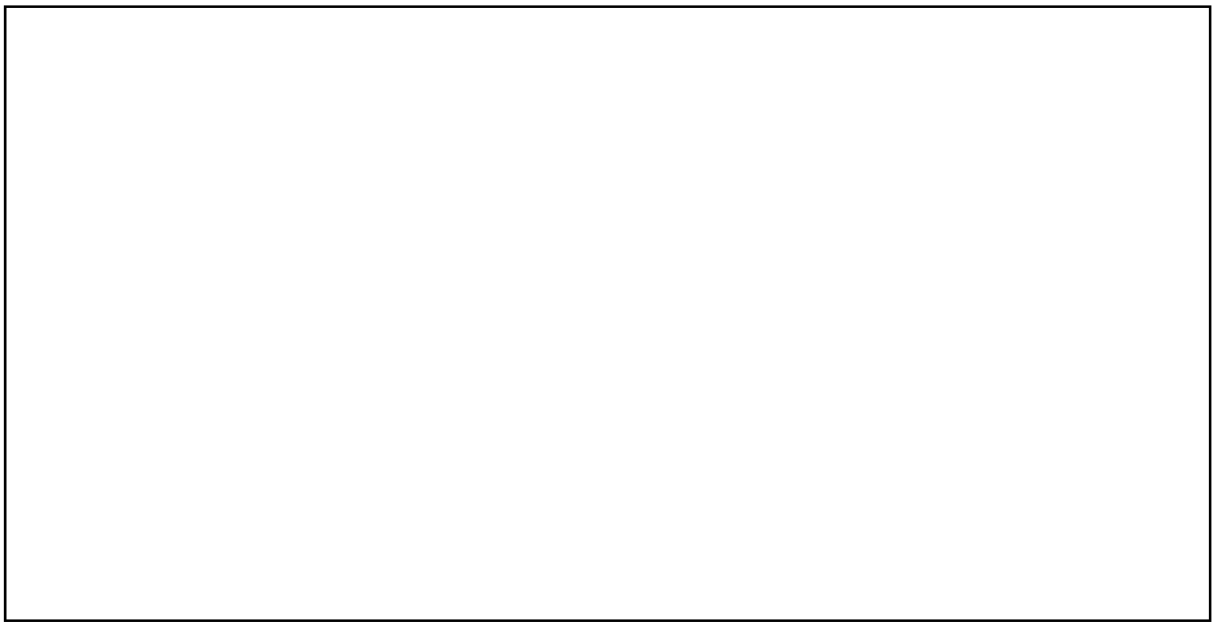

Fig 1:- this figure presents proposed Model

The architecture of the proposed system follows the **federated learning model with integrated privacy-preserving techniques**. As illustrated in *Figure 4.2.1*, the architecture consists of multiple client devices, each containing local data and a local machine learning model. These clients train the model independently on their private data and send encrypted model updates to a central server.

Key components of the system include:

- **Client Devices**: Each client contains:
  - Local dataset (e.g., healthcare, financial, or sensor data)
  - Local model which is trained using only the device's data
  - Encryption mechanism to protect model updates before transmission
- **Encrypt/Secure Aggregation Module**: This central module:
  - Receives encrypted model updates from clients
  - Applies homomorphic encryption or secure multiparty computation (SMPC)
  - Aggregates updates without accessing raw data
- **Central Server**: The server:
  - Hosts the aggregated model (global model)
  - Sends updated global model parameters back to clients
  - Does not access or store raw client data
- **Global Model**: The final outcome of the system that is collaboratively trained by all participating clients while preserving data privacy.

The architecture also supports differential privacy by injecting noise into the gradients or model updates before they are encrypted. This additional layer of defense ensures compliance with privacy regulations such as **GDPR** and **HIPAA**.

5.1 Algorithm

- K: total number of clients
- C: fraction of clients selected per round (e.g., 0.1 or 0.2)
- R: total number of communication rounds

- E: local epochs per client
- η: learning rate
- GlobalModel: initialized on the server
- Clients train locally on **private data**

steps

Initialize GlobalModel θ_0 on the Server
For round r in 1 to R:
  Server:
    m ← max(C * K, 1) # Number of clients to sample
    S_r ← random sample of m clients from total K clients
    Broadcast current GlobalModel θ_r to all selected clients in S_r
  Each selected Client i ∈ S_r (executed in parallel):
    θ_i ← θ_r # Receive global model
    Train θ_i on local data D_i for E epochs using learning rate η:
      For local epoch e in 1 to E:
        Update θ_i using local SGD on D_i
    Optional Privacy Step:
      - Clip model updates to bound sensitivity
      - Add Gaussian noise for Differential Privacy: θ_i = θ_i + Noise
    Send local model update Δθ_i = θ_i - θ_r back to server
  Server:
    Optional: Perform Secure Aggregation of all updates Δθ_i
    Aggregate updates to update global model:
      θ_r+1 = θ_r + (1/m) * Σ Δθ_i for all i ∈ S_r
Output: Final trained GlobalModel θ_R

6. RESULTS:

Out[17]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slo
0	56027.396936	55976.396936	55975.396936	56100.396936	56187.396936	55975.396936	55976.396936	56143.396936	55975.396936	55976.396936	55977.3969
1	56028.396936	55976.396936	55975.396936	56115.396936	56178.396936	55976.396936	55975.396936	56130.396936	55976.396936	55978.496936	55975.3969
2	56045.396936	55976.396936	55975.396936	56120.396936	56149.396936	55975.396936	55976.396936	56100.396938	55976.396936	55977.996936	55975.3969
3	56036.396936	55976.396936	55975.396936	56123.396936	56178.396936	55975.396936	55976.396936	56136.396936	55975.396936	55975.396936	55977.3969
4	56033.396936	55975.396936	55975.396936	56075.396936	56223.396936	55975.396936	55975.396936	56097.396936	55975.396936	55976.396936	55976.3969

Fig 2:-sample In the above screen we are applying the Differential Privacy algorithm on RELEVANT features dataset and after applying we can see entire dataset values get changed with noise data and this changed values you can see in the above table

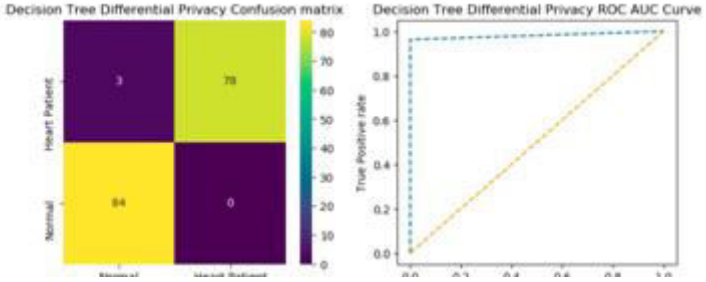

Fig 3:- In the above screen training Decision Tree algorithm on Differential Privacy values and after training we perform prediction on test data and then Decision tree got 98% accuracy on Differential privacy values which proves there is no effect on ML model after applying privacy. In confusion, the matrix graph x-axis represents Predicted Labels and y-axis represents True Labels where all blue boxes represent incorrect prediction count and yellow, green represent correct prediction count. In the ROC curve graph x-axis represents False Positive Rate and y-axis represents True Positive rate and if the blue line comes below the orange line, then all predictions are false and if goes above the orange line then all predictions are correct

Out[22]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	3680	416	352	8352	13920	352	416	11104	352	416	480	480	544
1	3744	416	352	9312	13344	416	352	10272	416	544	352	352	544
2	4832	416	352	9632	11488	352	416	8352	416	480	352	352	544
3	4256	416	352	9824	13344	352	416	10656	352	352	480	416	544
4	4064	352	352	6752	16224	352	352	8160	352	416	416	352	480
...	...	...	...	...	...	...	...	...	...	...	...	...	...
816	4128	416	416	9312	14496	352	416	10848	416	352	480	352	480
817	4192	416	352	8352	16864	352	352	9376	416	480	416	416	544
818	3360	416	352	7392	17952	352	352	7904	416	416	416	416	480
819	3552	352	352	7392	16608	352	352	10528	352	352	480	352	480
820	3808	416	352	8032	12384	352	416	7584	352	416	416	416	544

Fig 4:- Applying Homomorphic Encryption to training feature

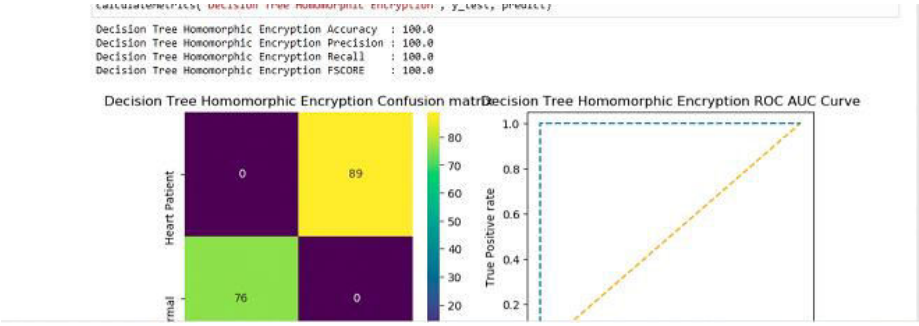

Fig 5:- In the above screen training decision tree on Homomorphic features and then decision tree got 100% accuracy and can see other metrics graph of trained model performance.

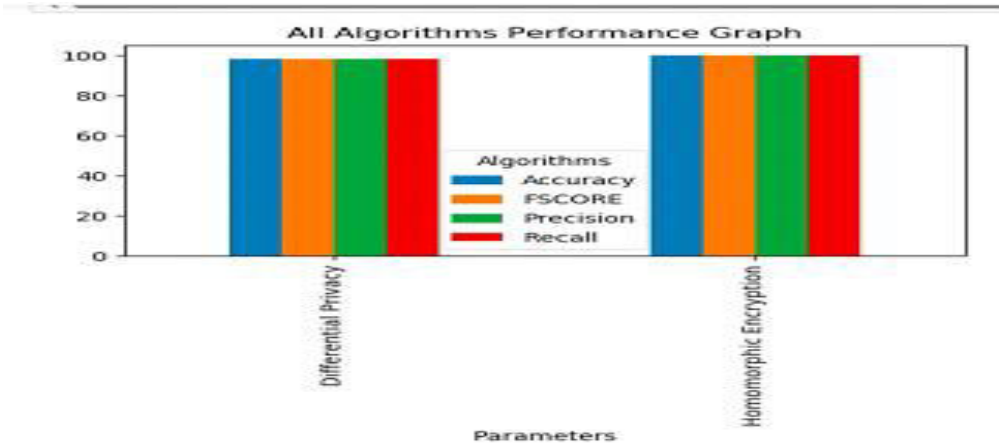

Fig 6:- In the above graph displaying Decision tree performance on both Differential Privacy and Homomorphic features where x-axis represents technique name and y-axis represents accuracy and other metrics in different color bars and from above graph we can say both techniques manage to give ML model accuracy more than 95%.

	Algorithm Name	Accuracy	Precision	Recall	FSCORE
0	Differential Privacy	98.181818	98.275862	98.148148	98.179411
1	Homomorphic Encryption	100.000000	100.000000	100.000000	100.000000

Fig 7:- In the above screen displaying both algorithm performance in tabular format. So, from above experiments we can see ML shows no change in performance even after model get privacy so by using this privacy, we can secure model features from attackers

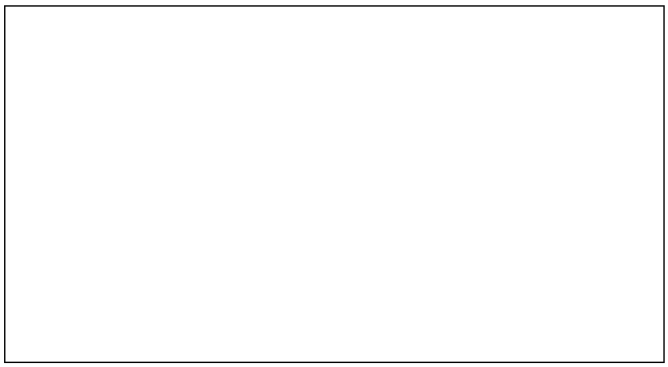

Fig 8:- In the above screen we are selecting the Differential Privacy mode from the Streamlit interface. The user can adjust the privacy budget (epsilon value), and the model displays its resulting accuracy after training with differential noise applied

7. CONCLUSION

As data privacy becomes an increasingly critical concern in the digital era, Federated Learning (FL) emerges as a promising paradigm for enabling collaborative machine learning without compromising individual privacy. By ensuring that raw data remains localized on user devices and only model updates are shared, FL significantly reduces the risk of data breaches and ensures greater compliance with privacy regulations such as GDPR and HIPAA.

However, to truly secure user privacy in FL, it is essential to address key challenges such as inference attacks, malicious participants, and communication inefficiencies. Integrating advanced

techniques like secure aggregation, differential privacy, and robust encryption enhances both the privacy and security of federated learning systems.

This project emphasizes the development of a secure and privacy-preserving federated learning framework that balances performance, scalability, and compliance. Through careful design of functional components and adherence to non-functional requirements such as security, reliability, and usability, the system can serve as a practical and ethical solution for privacy- preserving machine learning in real-world applications.

Ultimately, secure federated learning represents a critical step toward building trust in AI systems, empowering users with control over their data, and enabling innovation in a privacy- conscious manner.

7.1 FUTURE SCOPE

While this project successfully demonstrates the feasibility and effectiveness of secure and privacy-preserving federated learning, several avenues remain open for future enhancement and research:

1. **Scalability with Real-Time Data Streams:** The current implementation is based on static datasets. Future work can focus on integrating real-time data streaming capabilities to enable continuous learning from dynamic sources such as IoT devices or mobile apps.
2. **Integration of Blockchain for Auditing:** Blockchain technology can be introduced to create immutable logs of model updates and data access, ensuring traceability and increasing trust among participants in federated environments.
3. **Support for Multi-Model Learning:** Extending the framework to support simultaneous training of multiple models across heterogeneous data and devices can improve learning quality in complex use cases such as smart healthcare and autonomous systems.
4. **Enhanced Defense Against Adversarial Attacks:** Ongoing research into adversarial machine learning can be integrated to better defend against model inversion attacks, backdoor attacks, and poisoning of local updates.
5. **Cross-Device Optimization:** Optimization techniques such as model compression, quantization, and resource-aware scheduling can be incorporated to improve FL performance on low-power or mobile devices.
6. **Edge and Cloud Hybrid FL:** A future direction could include implementing a hybrid edge-cloud FL system, where computation is distributed intelligently based on network latency, energy consumption, and data sensitivity.
7. **User-Friendly Interface and Deployment:** The system can be enhanced with a graphical user interface (GUI) and containerized (using Docker or Kubernetes) for real- world deployment and testing at scale.

REFERENCE

1. **Shokri, R., & Shmatikov, V.** (2015). *Privacy-preserving deep learning*. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 1310–1321). https://doi.org/10.1145/2810103.2813687
2. **Gentry, C.** (2009). *A fully homomorphic encryption scheme*. *PhD thesis, Stanford University*. https://crypto.stanford.edu/craig
3. **Dwork, C., & Roth, A.** (2014). *The algorithmic foundations of differential privacy*. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4), 211–407. https://doi.org/10.1561/0400000042
4. **Lindell, Y., & Pinkas, B.** (2009). *Secure multiparty computation for privacy- preserving data mining*.

Journal of Privacy and Confidentiality, 1(1), 59–98. https://doi.org/10.29012/jpc.v1i1.455

5. **Kairouz, P., McMahan, H. B., et al.** (2021). *Advances and open problems in federated learning*. *Foundations and Trends® in Machine Learning*, 14(1–2), 1–210. https://doi.org/10.1561/2200000083
6. **Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L.** (2016). *Deep learning with differential privacy*. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 308–318). https://arxiv.org/abs/1607.00133
7. **Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I., & Talwar, K.** (2017). *Semi-supervised knowledge transfer for deep learning from private training data*. *arXiv preprint* arXiv:1610.05755. https://arxiv.org/abs/1610.05755
8. **Bost, R., Popa, R. A., Tu, S., & Goldwasser, S.** (2015). *Machine learning classification over encrypted data*. *In NDSS*. https://eprint.iacr.org/2014/785.pdf
9. **McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A.** (2017). *Communication-efficient learning of deep networks from decentralized data*. *In Artificial Intelligence and Statistics* (pp. 1273–1282). https://arxiv.org/abs/1602.05629
10. **Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing,J.** (2016). CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. Proceedings of the 33rd International Conference on Machine Learning (Vol. 48), 201–210. https://www.microsoft.com/en-us/research/publication/cryptonets-applying-neural-networks-to-encrypted-data-with- high-throughput-and-accuracy/