Explainability for FHE NN

Rupesh Raj Karn

New York University Abu Dhabi

ABSTRACT

Fully Homomorphic Encryption (FHE) offers a robust solution for maintaining data privacy by enabling computations on encrypted data without decryption. However, this encryption introduces significant challenges for explainability in neural networks (NN), as traditional interpretability methods require access to plaintext data and model parameters. This paper explores the adaptation of SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) for FHE-NN to provide insights into model decisions while preserving privacy.

We propose a novel approach where SHAP and LIME are adapted to operate within the encrypted domain. For SHAP, we pre-compute Shapley values on plaintext training data and develop homomorphic algorithms to map encrypted predictions to these pre-computed values. For LIME, we generate encrypted perturbations and compute encrypted predictions, mapping these to pre-computed local interpretable models. This ensures that the entire process, from training to inference, remains within the encrypted domain, thereby maintaining data privacy.

Our methods leverage homomorphic encryption to perform necessary computations without exposing sensitive information. This approach enhances trust and transparency in secure machine learning applications by providing meaningful and interpretable insights into model behavior. The advancements presented in this paper pave the way for more transparent and accountable AI systems, fostering broader adoption of privacy-preserving machine learning technologies.

Keywords: Fully Homomorphic Encryption, Explainability, SHAP, LIME

1 EXPLAINABILITY IN FULLY HOMOMORPHIC ENCRYPTED NEURAL NETWORKS (FHE-NN)

1.1 Objective

Fully Homomorphic Encryption (FHE) enables computations on encrypted data without needing to decrypt it, ensuring data privacy throughout the process. However, this encryption poses significant challenges for explainability in neural networks (NN), as traditional methods for interpreting model decisions rely on access to plaintext data and model parameters. This section explores how to provide insights into model decisions in FHE settings, focusing on adapting frameworks like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to work with encrypted models.

1.2 Adapting SHAP and LIME for FHE-NN

1.2.1 Mathematical Adjustments

- **SHAP**: SHAP values are derived from cooperative game theory, attributing the contribution of each feature to the model's output. In an FHE setting, the computation of SHAP values must be adapted to operate on encrypted data. This involves developing homomorphic algorithms that can approximate the Shapley values without decrypting the data.

- **LIME**: LIME explains individual predictions by approximating the model locally with an interpretable model. For FHE-NN, LIME's perturbation-based approach needs to be modified to generate encrypted perturbations and compute the corresponding encrypted predictions. Homomorphic encryption schemes must support these operations efficiently to maintain the practicality of LIME in FHE settings.

1.2.2 Feature Extraction

- **Interpretable Features**: Extracting interpretable features from encrypted models requires innovative approaches to ensure that the features remain meaningful while preserving privacy. This could involve designing encrypted feature extraction methods that leverage the structure of the neural network and the nature of the encrypted data.

- **Dimensionality Reduction**: Techniques such as Principal Component Analysis (PCA) or t-SNE, adapted for homomorphic encryption, can help reduce the complexity of the data, making it easier to interpret while maintaining encryption.

1.2.3 *Trust and Transparency*

- **Model Auditing**: Regular auditing of FHE-NN models using adapted SHAP and LIME can help build trust by providing stakeholders with insights into model behavior without compromising data privacy. This involves creating secure protocols for auditors to verify the explanations generated by the models.

- **User-Friendly Interfaces**: Developing interfaces that present the explanations in an understandable manner to non-experts is crucial. These interfaces should translate the complex mathematical operations into intuitive visualizations and narratives.

1.3 Detailed Adaptation of SHAP and LIME

1.3.1 *SHAP with FHE*

1. **Pre-computation of SHAP Values**:

    - For each training instance x_i in the plaintext training data, compute the SHAP values $\phi_{i,j}$ for each feature j. These values represent the contribution of feature j to the prediction for instance x_i.

2. **Encrypted Predictions**:

    - During inference, the model makes predictions on encrypted data. Let $\hat{y}_{enc}$ be the encrypted prediction for an encrypted input x_{enc}.

3. **Mapping to Pre-computed SHAP Values**:

    - To map the encrypted prediction $\hat{y}_{enc}$ to the pre-computed SHAP values, we need to securely compare $\hat{y}_{enc}$ with the predictions on the plaintext training data.

    - Let $\hat{y}_i$ be the plaintext prediction for training instance x_i. Encrypt these predictions to get $\hat{y}_{i,enc}$.

    - Compute the difference between $\hat{y}_{enc}$ and each $\hat{y}_{i,enc}$ using FHE:

    $$d_i = \hat{y}_{enc} - \hat{y}_{i,enc}$$

    - Identify the training instance x_i with the smallest d_i (in encrypted form). This instance is the closest match to the encrypted input x_{enc}.

    - Retrieve the pre-computed SHAP values $\phi_{i,j}$ for this instance x_i.

1.3.2 *LIME with FHE*

1. **Pre-computation of Local Interpretable Models**:

    - For each training instance x_i, generate perturbations $\{x_i^{\prime(k)}\}$ and fit a local interpretable model g_i (e.g., a linear model) to approximate the neural network's behavior around x_i.

2. **Encrypted Perturbations and Predictions**:

    - During inference, generate encrypted perturbations $\{x_{enc}^{\prime(k)}\}$ of the encrypted input x_{enc}.

    - Compute encrypted predictions $\{\hat{y}_{enc}^{\prime(k)}\}$ for these perturbations using the encrypted model.

3. **Mapping to Pre-computed Local Models**:

    - For each encrypted perturbation $x_{enc}^{\prime(k)}$, compute the difference between its encrypted prediction $\hat{y}_{enc}^{\prime(k)}$ and the encrypted predictions of the perturbations from the training data:

    $$d_i^{\prime(k)} = \hat{y}_{enc}^{\prime(k)} - \hat{y}_{i,enc}^{\prime(k)}$$

    - Identify the training instance x_i whose perturbations $\{x_i^{\prime(k)}\}$ have the smallest differences $d_i^{\prime(k)}$ with the encrypted perturbations $\{x_{enc}^{\prime(k)}\}$.

    - Retrieve the pre-computed local model g_i for this instance x_i.

1.4 Ensuring Privacy and Security

- **Homomorphic Operations**: All operations, including the computation of differences and identification of the closest matches, are performed using homomorphic encryption. This ensures that the data remains encrypted throughout the process.

- **Efficiency**: The efficiency of these operations depends on the homomorphic encryption scheme used. Optimizing these schemes for the specific requirements of SHAP and LIME can help maintain practicality.

1.5 Summary

Enhancing explainability in FHE-NN is essential for the broader adoption of privacy-preserving machine learning applications. By adapting existing interpretability frameworks like SHAP and LIME to work with encrypted data, we can provide meaningful insights into model decisions while maintaining data privacy. These advancements will not only enhance trust in secure machine learning applications but also pave the way for more transparent and accountable AI systems.

2 ENHANCING SECURITY OF SHAP AND LIME WITH ORDER-PRESERVING ENCRYPTION (OPE)

2.1 Using Order-Preserving Encryption (OPE) for SHAP/LIME Values

Order-Preserving Encryption (OPE) is a form of encryption where the order of the plaintext values is preserved in the ciphertext. This property allows for efficient range queries and comparisons on encrypted data without decryption. To enhance the security of SHAP and LIME values, we propose encrypting these values using OPE before providing them to the user.

Let $\phi_{i,j}$ represent the SHAP value for feature j of instance x_i. Instead of providing $\phi_{i,j}$ directly, we encrypt it using an OPE scheme:

$$\phi_{i,j}^{OPE} = \mathrm{OPE}(\phi_{i,j})$$

Similarly, for LIME, let $g_i(x)$ be the local interpretable model for instance x_i. The coefficients of this model, β_j, are encrypted using OPE:

$$\beta_j^{OPE} = \mathrm{OPE}(\beta_j)$$

This ensures that the encrypted SHAP/LIME values maintain their order, allowing for meaningful comparisons and interpretations without revealing the actual values.

2.2 Isolating FHE and OPE for Enhanced Security

By using two distinct encryption mechanisms—FHE for model parameters and inference data, and OPE for SHAP/LIME values—we create an additional layer of security. Each encryption mechanism uses a different key, providing a backup in case one key is compromised.

Let K_{FHE} be the key for FHE and K_{OPE} be the key for OPE. The model parameters θ and inference data x are encrypted using FHE:

$$\theta_{enc} = \text{FHE}(\theta, K_{FHE})$$

$$x_{enc} = \text{FHE}(x, K_{FHE})$$

The SHAP/LIME values are encrypted using OPE:

$$\phi_{i,j}^{OPE} = \text{OPE}(\phi_{i,j}, K_{OPE})$$

$$\beta_j^{OPE} = \text{OPE}(\beta_j, K_{OPE})$$

This separation ensures that even if K_{FHE} is compromised, the SHAP/LIME values remain secure under K_{OPE}, and vice versa.

2.3 Enhancing Security Against Attacks

Encrypting SHAP/LIME values with OPE enhances security by mitigating the risk of recovering original inference data from plaintext SHAP/LIME values. If an attacker intercepts the SHAP/LIME values (e.g., through a man-in-the-middle attack), they would only obtain the encrypted values, which are not directly useful without the decryption key.

Consider an attacker intercepting $\phi_{i,j}^{OPE}$. Without K_{OPE}, the attacker cannot decrypt $\phi_{i,j}^{OPE}$ to obtain $\phi_{i,j}$. This prevents the attacker from using the SHAP values to infer sensitive information about the input data x.

Mathematically, the security of OPE ensures that:

$$\Pr[\text{OPE}^{-1}(\phi_{i,j}^{OPE}) = \phi_{i,j}] \approx 0 \quad \text{without} \quad K_{OPE}$$

Thus, the combination of FHE and OPE provides a robust security framework, ensuring that even if one encryption scheme is compromised, the other remains intact, protecting the overall system.

2.4 Summary

By integrating OPE with FHE, we enhance the security of SHAP and LIME values in privacy-preserving neural networks. This dual encryption approach not only isolates the FHE and OPE parts, providing a backup in case of key compromise, but also protects against potential attacks that aim to recover original inference data from plaintext SHAP/LIME values. This method ensures that sensitive information remains secure, fostering trust in secure machine learning applications.

REFERENCES