International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I9P109

Volume 73 Issue 9, 91-99, September 2025

Original Article

Efficient Implementation of AES-256 for Secure Machine Learning Datasets: A Performance and Compatibility Study

Nandini Sharma[1], Pritaj Yadav[2]

[1,2]*Department of Computer Science & Engineering, Rabindranath Tagore University, Bhopal, India.*

[1]*Corresponding Author : nandini72@gmail.com*

Received: 18 February 2025 Revised: 16 August 2025 Accepted: 23 August 2025 Published: 30 September 2025

Abstract - *The present study explores a method combining machine learning with the AES-256-based encryption to protect the datasets while maintaining the accuracy. The presented approach addresses an increasing need for data security in the context of increased cyber threats that particularly focus on healthcare and finance. The AES-256 is one of the well-known algorithms that is susceptible to attempted attacks and ensures confidentiality throughout the transmission and storage to encrypt datasets. The neural network processes the data and attains an accuracy of 87% for binary classification tasks, which validates the effectiveness and compatibility of the model. Different performance indicators demonstrate the seamless trade-off between security and efficiency, which classifies accuracy and encryption overhead. The paper provides a framework customized for the different businesses that require stringent data protection and highlights the significance of handling data safety.*

Keywords - *Security, AES-256, Machine Learning, Neural Network, Federated Learning.*

1. Introduction

Sensitive dataset security has become crucial in the current Machine Learning (ML) and the huge data set environment, particularly in sectors such as healthcare, finance, and defense, where data breaches have serious repercussions. The most reliable encryption method is Advanced Encryption Standard (AES), especially the 256-bit version. The AES-256-based encryption is one of the greatest choices for safeguarding the ML datasets against unwanted tampering and cyberattacks because it provides an efficient trade-off between efficiency and security [1, 2]. The rapid growth in the ML area is revolutionising a wide range of applications, with Neural Networks (NNs) at the center of the transformation. The NN, inspired by the human brain, is an interconnected network of neurons and is one of the essential parts in solving the complex pattern recognition and predictive modelling tasks [3]. These networks have gained significant attention for their effectiveness in binary classification tasks, where the target is to categorize the data into two identical classes. The growing body of literature highlighted the role of deep learning models, particularly feedforward NNs that improve the accuracy and efficiency of classification algorithms across diverse fields like healthcare [4]. As per the efficient training of ML models, an extensive volume of data is frequently needed. On the other hand, several risks related to the transmission and storage of such information, like manipulation, interception, and illegal use, exist in this field. In order to ensure confidentiality and integrity throughout the storage and transmission stages, encryption approaches such as AES-256 aid in reducing these hazards [5, 6].

The sensitivity and volume of analysed data are increased in parallel with the ML applications, with the explosive growth. Protecting the confidentiality of users, complying with the regulations, and reducing the risks linked to data breaches depend on safeguarding such information, especially in the vital sectors like healthcare, banking, and national security [7, 8]. In this manner, the AES-256 has become a key component of data security.

As the key length, AES-256 is more secure than the 128- and 192-bit substitutes and provides stronger defense against brute-force attacks. AES-256 is a strong encryption, and there are particular challenges within the ML-based applications. System performance is impacted by the computational expense of encryption and decryption, which makes the training and inference procedures slower [9]. It is critical to address the trade-off between competence and security, especially for real-time applications and resource-intensive ML activities [10]. The unique difficulty is integrating the AES-256 with ML processes. The processing cost is also associated with the encryption and the decryption procedures, which is an important concern as it might affect ML model training time and effectiveness. Furthermore, the second layer

of difficulty is the interoperability with the systems that are distributed with the data that is frequently processed across the numerous nodes [11]. Notwithstanding the difficulties, breakthroughs are being made in the hardware acceleration, like "Graphics Processing Unit (GPU)" and "Field Programmable Gate Array (FPGA)," based on the applications that show the AES-256 encryption performs noticeably better than the author of [12]. Additionally, novel approaches like lightweight cryptography are explored in order to maximize the encryption performance in resource-constrained environments like edge computing and the Internet of Things (IoT) [13, 14].

The study aims to assess the AES-256's compliance and efficacy in the procedure of ML models. The present investigation aims to identify the methods for maximizing the AES-256 utilization without compromising data security or system efficiency through investigating the effects across multiple phases, such as preliminary processing, model training, and implementation. The academics and practitioners protect ML pipelines that additionally look at the trade-offs between computing efficiency and encryption resilience.

The current research demonstrated the capacity of NNs in order to handle the vast volumes of data, which leads to improving the precision and recall in the binary classification tasks [15, 16]. Furthermore, the integration of advanced optimization techniques like backpropagation and gradient descent has allowed the NNs to achieve significant performance gains with the large datasets [17, 18]. However, the advances in challenges like overfitting, model interpretability, and training time continue in order to demand further exploration and optimization [19-21]. The main aim of the research is to discuss some of the challenges that explore the competence of an NN-based model in accurately classifying data, emphasising improving both performance and computational cost. The paper focuses on enhancing the security of ML datasets using the AES-256 algorithm. Furthermore, it presents the comprehensive performance evaluation of the encryption process and examines the impact of AES-256 on training time, accuracy, and system compatibility across different ML models.

2. Literature Review

The section provides a concise summary of the previous research of various researchers in the particular area. There is so much literature present for the particular area of work that provides, after the refinements of the considered papers, which are discussed below:

In several fields, the integration of AI in healthcare and power systems has produced a revolutionary effect. According to the author of Patel et al. (2024) [22], AI plays an essential part in the process optimization and failure prevention that can enhance the safety and efficiency of operations in the power industry. In the same way, the author of Gupta et al. (2022) [23] illustrated how AI is used in cloud-based storage for healthcare, highlighting the significance of secure data processing and predictive analytics.

As the utilization of AI increases in day-to-day life, protecting devices' privacy and security is taking the center stage. The authors of Villegas and García (2023) [24] presented a thorough framework that tackles key problems like data integrity and access control to maintain privacy and safety in the field of AI. Furthermore, the author, like Padmanaban (2024) [25], studied the privacy-preserving designs that offer details regarding the strategies that achieve a balance between system performance and security requirements. According to the author of Bonawitz et al. (2019) [26], the architectures become even more relevant when taking into account the scalability of federated learning, which produced the system architecture that permits extensive, privacy-conscious AI model training over dispersed networks.

The latest trends in technology, especially for protecting privacy, are critical in sensitive industries like the healthcare sector. The thorough analysis of the privacy-preserving ML model was provided by the author of Tanuwidjaja et al. (2020) [27], which highlighted methods for maintaining the confidentiality of data without sacrificing the model's performance. Moreover, the author of Michael (2021) [28] further illustrates the need for privacy at the edge of the AI networks by discussing the difficulties and suggesting fixes for protecting the localized processed data on the edge devices. The privacy-centric framework set out by the author of Calvaresi et al. (2021) [29] in the development of the compliant structure for health-assistant chatbots was supported, which emphasized the requirement for sophisticated security measures in order to protect patient data in contemporary healthcare systems.

In order to meet both technical and legal requirements, it is essential to integrate an AI/ML model with enhanced security protocols. For this, Bayani, Prakash, and Malaiyappan (2023) [30] investigated enhanced security and compliance challenges that present through a unified assurance framework, promoting strong compliance with regulations and secure cloud services. Despite the significant advancements in securing the ML, the critical research gap persists in balancing the strong encryption with efficient model performance. The existing studies predominantly emphasize the lightweight encryption algorithms, like AES-128 or the custom lightweight cyphers for the resource-constrained environments like IoT, often sacrificing the security strength for the computational feasibility (Padmanaban et al., 2022). The previous researchers explore privacy-preserving techniques like differential privacy or the homomorphic encryption within the federated learning frameworks, but typically overlook the integration and overhead of the robust encryption standards (Gupta &

Malhotra, 2021). Moreover, the research isolates the security implementation from the performance analysis, leaving a void in understanding how strong encryption affects model accuracy, training time, and system compatibility. The paper further addresses the gap by demonstrating an efficient implementation of AES-256 encryption for securing machine learning datasets, accompanied by a detailed evaluation of its impact on neural network performance. The results validate that high-level encryption can be integrated without significantly compromising model efficiency, thereby offering a scalable, secure, and adaptable framework for ML applications in sensitive domains like healthcare and finance.

3. Hybrid AES-256 Encryption for ML Data Security

The AES-256-based encryption is used with the key management techniques. Furthermore, the ML approaches are also included in order to validate these things. The in-depth details of the techniques used in the research are discussed in this section.

3.1. Method Components

- Data Preprocessing and Encryption: The raw data is first preprocessed and then normalized. The AES-256 encryption is applied to ensure that the data is securely stored and transmitted.
- Encryption and Decryption Workflow: Define the encryption and decryption workflow, optimizing for minimal processing time while maintaining security.
- Model Training and Evaluation on Encrypted Data: Secure computation techniques allow the model to process encrypted data, avoiding plaintext exposure.

AES-256 Encryption Algorithm: Mathematical Foundation

AES-256 is a symmetric block cipher that using a 256-bit key in order to encrypt the data in fixed 128-bit blocks. It also operates through a series of transformations based on Substitution-Permutation Networks (SPNs). Here is a breakdown of key components:

3.1.1. Key Expansion

AES-256's key expansion uses a 256-bit key to derive 240 bytes of key material for every encryption round. The key schedule uses Rijndael's key schedule technique, which expands the original key into multiple 128-bit subkeys:

$$K_{(0)} = K, K_{(i)} = f(K_{(i-1)}) \quad (1)$$

Where i is a transformation function involving rotation, XOR operations, and the AES constant RCON.

3.1.2. Round of Encryption

AES-256 uses 14 rounds of transformation for each 128-bit data block. Each round includes:

- SubBytes: Applies a non-linear substitution using an S-box, enhancing security against differential cryptanalysis.

$$SubBytes_{(s_{i,j})} = S_{[s_{i,j}]} \quad (2)$$

Where the S is a substitution box that maps each byte.

- ShiftRows: Rotates rows of the state matrix to the left by offsets, increasing diffusion.

$$ShiftRows(s_{i,j}) = s_{i,(j+1)mod4} \quad (3)$$

- MixColumns: Operates on the columns by multiplying each by a constant polynomial matrix to ensure diffusion.

$$\begin{bmatrix} s'_{0,j} \\ s'_{1,j} \\ s'_{2,j} \\ s'_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{bmatrix} \quad (4)$$

- AddRoundKey: It performs the XOR between the state and the round key that is derived from the expanded key.

$$AddRoundKey_{(s)} = s \oplus K_{round} \quad (5)$$

3.2. Mathematical Analysis for Encryption in ML Pipelines

Using AES-256 introduces a computational overhead in the ML pipelines, particularly if data needs to be decrypted before training or inference. To maintain data security while minimizing this impact, the homomorphic encryption or the Secure Multi-Party Computation (SMPC) allows the encrypted data to be processed without decryption, adding complexity layers.

The computational complexity 'T' of encryption scales with the number of data blocks 'n' and the number of rounds 'r', leading to:

$$T = O(n\ X\ r\ X\ block\ size) \quad (6)$$

The block size is 128 bits for the AES-256-based encryption, making the encryption computationally feasible but potentially costly with very large datasets.

3.3. Neural Network Processing Stage

NNs are the only powerful models that contain interconnected nodes or neurons, and are organized layer-wise to process the data. The processing steps are involved in the training and implementation of an NN-based model for the classification tasks that are crucial for the optimization, and in order to safeguard the model that generalizes well to new and unseen data. Moreover, the section outlines the essential steps intricate in the methodology, from data preprocessing to the final prediction, which emphasizes the importance of each

stage for achieving accurate and efficient outcomes. The process begins with data preprocessing that includes data normalization and scaling in order to safeguard that all the features are on a similar scale, preventing the model from becoming biased toward certain input features. This step is crucial because it impacts the convergence speed during the training and the complete effectiveness of the NN. Once the data is prepared, the NN architecture often consists of an input layer and one or more hidden layers with an output layer. This architecture's number of layers and neurons significantly affects the model's capability to learn complex patterns.

Subsequent steps involve the forward propagation process, in which the input data is passed into the network's layers. During this process, the overall weighted sums of the inputs are computed and passed over to activation functions like the Rectified Linear Unit (ReLU) or the Sigmoid in order to introduce the non-linearity model. The choice of the activation function plays a critical role in enabling the network to learn nonlinear relationships in the data. Furthermore, the forward propagation and the output are associated with the actual target value. The error is computed with a loss function such as the Mean Squared Error (MSE) for the Cross-Entropy Loss or regression tasks for classification.

The next step is processing the NN in the backpropagation, from which the model is learned by regulating the weights based on the computed error. According to the optimization algorithms, the network performs the gradient updates in order to minimalize the loss function. Backpropagation is one of the essential functions for apprising the weights that enables the network to improve the accuracy over time. The multiple iterations, or according to the epochs, the model gradually refines the weights in order to minimize the error and improve the predictive power. Finally, after the training, the NN is evaluated on the basis of the validation data set to assess the performance and generalise the model's ability. Metrics like accuracy, precision, recall, and the F1-score are commonly used parameters for measuring performance. The trained NNs are deployed to make predictions on new and unseen data.

These processing steps are vital for ensuring the optimal functioning of NNs in real-world applications. Each step contributes to the network's ability to learn from data, adapt to new information, and make predictions that drive intelligent decision-making in various domains, i.e., healthcare to finance.

Data Preprocessing: Input data is normalized to ensure uniform scaling, making training more stable. If x_i is an input feature, normalization is:

$$x_i' = \frac{x_i - \mu}{\sigma} \quad (7)$$

"μ" is the mean and "σ" is the standard deviation.

Input Layer: The data passes through the input layer, which holds n features (nodes). If "X" is the input vector, the input layer activation is:

$$a_0 = X \quad (8)$$

Hidden Layers: Each hidden layer transforms the input using weights "W" and biases "b", followed by an activation function f(x). For layer l:

$$z^l = W^l . a^{l-1} + b^l$$

$$a^l = f(z^l \quad (9)$$

Common activation functions also include:

Relu: f(x)=max (0,x)

Sigmoid: f(x)=1/(1+ e^(-z))

Output layer: A sigmoid activation function outputs probabilities for the binary classification.

$$y = \sigma(z) = \frac{1}{1+e^{-z}} \quad (10)$$

Loss function: The binary cross-entropy is used in order to measure the error between predictions (y^) and ground truths y:

$$L = -\frac{1}{m}\sum_{i-1}^{m}[y_i \log(\widehat{y_i}) + (1-y_i)\log(1-\widehat{y_i})] \quad (11)$$

4. Research Methodology

The proposed technique aims to use the AES-256 encryption to secure the sensitive datasets before processing the ML pipeline. AES is chosen for speed and robustness, with a focus on AES-256-based encryption for enhanced security. Encrypting the data at rest and in transit mitigates the risks from unauthorized access, data leaks, and adversarial attacks on the processing of ML.

Key management plays a critical role in your methodology by providing robust security and ensuring that encryption keys are effectively threatened and accessible only in order to authorized users or systems. Key management here is applied across multiple phases, including key generation, secure distribution, key rotation, and revocation.

- Key Generation

Key generation is the initial and vital step in securing the ML dataset. For AES-256, a secure 256-bit key is generated using a high-entropy, cryptographically secure random number generator (CSPRNG) or a hardware security module (HSM).

$$K = CSPRNG(256) \quad (12)$$

Where 'K' is the AES-256 encryption key and CSPRNG represents a cryptographically secure random number generator.

- Key Distribution

Ensure that the encryption key reaches authorized entities securely without exposure to unauthorized access. Use asymmetric encryption like Elliptic Curve Diffie-Hellman (ECDH) in order to securely exchange the AES key between the data owners and authorized users or nodes. This approach enables secure key exchange over potentially insecure networks. Each entity A and B generates private keys k_A and k_B and computes the public keys P_A=G^{kA} mod p and P_B g^{kB} mod p, where g and p are the parameters of the elliptic curve group.

The shared key is computed as:

$$K_{shared} = P_B^{k_A} = P_A^{k_B} \ mod p \quad (13)$$

- Key Rotation

Periodically refresh encryption keys to mitigate the risks of key compromise and limit the exposure of encrypted data. Use an automated key rotation schedule based on the data sensitivity or usage frequency. AES-256 keys are regenerated and redistributed securely in order to ensure continuity.

New keys are periodically generated as:

$$K_{new} = HMAC(K_{old}, timestamp) \quad (14)$$

Where K_{old} is the previous key, and the HMAC applies a hash function to timestamp the data for secure generation.

- Key Storage and Access Control

Securely store the keys and control access to prevent unauthorized usage. Store encryption keys in a secure environment like an HSM or use a centralized Key Management Service (KMS) for the cloud environments. Access to keys is controlled via authentication and authorization protocols that ensure only verified entities can retrieve keys. To authenticate users, consider Public Key Infrastructure (PKI), where each user possesses a private-public key pair. For access, a digital signature is used to verify identity.

Access to keys can be controlled by:

$$Access(U) = \begin{cases} Granted & if\ Auth(U) = True \\ Denied & otherwise \end{cases} \quad (15)$$

- Key Revocation

Ensure that compromised or expired keys are invalidated and cannot be used to decrypt the data. Implement a key lifecycle policy where the compromised keys are added to the revocation list and are inaccessible for future data decryption. If key 'K' is compromised, update the system's key revocation list (KRL):

$$KRL = KRL \cup \{K\} \quad (16)$$

Data encrypted with revoked keys is re-encrypted with new keys, ensuring security.

4.1. Introduction to Encryption and ML Dataset Security

The methodology focuses on enhancing the ML datasets' security through the AES-256 encryption combined with the robust key management system. In response to growing data breaches and privacy concerns in ML, the approach secures the sensitive datasets in storage and during transit across a distributed ML environment. Federated Learning (FL) is used to allow decentralized training on the encrypted datasets that ensure the sensitive data remains secure and private while achieving high model accuracy.

4.2. Data Preprocessing and Initial Encryption

- Objective: It ensures the data is securely encrypted before integrating into the ML pipeline.
- Process: The raw data are undergoing preprocessing steps that include normalization and anonymization to reduce unnecessary data exposure risks. The preprocessed data is then encrypted using AES-256.
- Encryption Process:
  - AES-256 Key Generation: The high-entropy, 256-bit symmetric key KKK is then generated with the Cryptographically Secure Random Number Generator (CSPRNG). The key is then kept confidential and forms the basis of the data encryption.
  - Data Encryption: The AES-256 algorithm encrypts each data block with the following steps:
    - ✓ SubBytes: It applies an S-box substitution for each byte in a data block, creating non-linearity.
    - ✓ ShiftRows and MixColumns: Enhance diffusion by the rearrangements of the bytes within the matrix.
    - ✓ AddRoundKey: It combines each data block with a round-specific subkey.
  - Storage and Access: The encrypted data is then stored securely, with access managed according to the user authentication and an authorization protocol.

4.3. Key Management Practices

The key management is critical in securing the encrypted ML datasets and involves generation, distribution, rotation, storage, and revocation of keys. The comprehensive approaches mitigate the risks of unauthorized access or data breaches by ensuring the keys are securely handled throughout their lifecycle.

- Key Generation: A unique AES-256 key is a combination generated as per the dataset, ensuring the data

segmentation and reducing the scope of potential exposure in case of a compromise.

- Key Distribution: An Elliptic Curve Diffie-Hellman (ECDH) based technique is used to secure the key exchange. This asymmetric encryption technique establishes the shared key for a secure AES-256 key transfer that mitigates the risks associated with network interception.
- Key Rotation: To limit the exposure of the long-lived keys, the AES-256 keys are rotated periodically. This practice is particularly critical for the highly sensitive data and frequent data updates, with the combination of the new key generated as a function of the time and previous key, e.g., using HMAC.
- Key Storage: Keys are stored in a hardware security module or the secure key management service. Only the authenticated and authorized users or systems have access to ensure confidentiality.
- Key Revocation: In the event of a key compromise, keys are revoked by adding them to a revocation list. Affected data is re-encrypted with a new key to maintain security integrity.

4.4. Federated Learning (FL) for Secure Model Training

It allows ML models to be trained crosswise the decentralized nodes without needing to centralize data, ensuring data privacy and security. Each node participates in the training process locally on encrypted data, transmitting only model updates rather than the raw data.

- Data Distribution: The encrypted dataset is distributed across multiple nodes for local training. Each node decrypts its local dataset using the securely transferred AES key and begins training.
- Model Aggregation: Each node computes local model parameters. These parameters are encrypted before transmission to the central server, which aggregates them to create an updated global model. Homomorphic encryption or secure aggregation techniques are employed in order to ensure that individual updates remain secure and private.
- Privacy Preservation: Differential privacy is applied to model updates before aggregation, ensuring that no individual data point is identifiable even in the combined model. Noise is added to the updates according to a privacy budget that balances accuracy and privacy.
- Communication Efficiency: To minimize the overhead of federated learning, communication rounds between nodes and the server are optimized, balancing model accuracy with bandwidth efficiency.

4.5. Security Analysis and Performance Evaluation

The effectiveness of the encryption-based security model is evaluated against several criteria to ensure that it meets security and performance benchmarks.

- Encryption Overhead Measurement: Measure the computational and latency overhead introduced by AES-256 encryption during preprocessing and model training. The encryption time T is calculated as:

$$T_{encryption} = n \times r \times block\ size \quad (17)$$

Where the "n" is the number of blocks, "r" is defined as the number of rounds (14 for AES-256), and the block size is 128 bits.

- Model Accuracy and Latency Analysis: Compare model accuracy on encrypted data with that of plaintext data. Latency is measured for each round in FL to assess the added complexity due to encryption and key management.
- Security Validation: Test the resilience of encryption against potential attacks, such as brute-force and ciphertext-only attacks. Key management practices are evaluated for robustness, including periodic key rotation and revocation.

This methodology uses a secure framework for the protection of the ML dataset using the AES-256 based encryption, a robust key management strategy, and the FL. By combining these techniques, the proposed approach ensures data privacy and security at every stage of the ML, from the preprocessing of the dataset to distributed model training. The methodology aims to advance the resilience of the ML models against data breaches while maintaining the model performance and meeting the stringent privacy requirements in distributed systems.

5. Result and Discussion

The results obtained from the current research provide a detailed evaluation of the efficacy in addressing the classification task. Through the analysis of the different performance parameters like the confusion matrix, the loss curves, and the accuracy trends, critical insights into the advantages and drawbacks of the model are obtained. The discussion explores the model's capacity to generalise insight into the data, the training security, and the trade-offs encountered in the specific error rates of the area. The following part aims to present an in-depth interpretation of the findings, accompanied by the quantitative metrics and visualizations to highlight the overall performance and suggest directions for future improvements.

```
Epoch 47/50, Loss: 0.4230, Val Loss: 0.4177, Accuracy: 0.8650
Epoch 48/50, Loss: 0.4360, Val Loss: 0.4146, Accuracy: 0.8650
Epoch 49/50, Loss: 0.4205, Val Loss: 0.4118, Accuracy: 0.8650
Epoch 50/50, Loss: 0.4194, Val Loss: 0.4091, Accuracy: 0.8700
Final Accuracy: 0.87
```

Fig. 1 Accuracy of the model

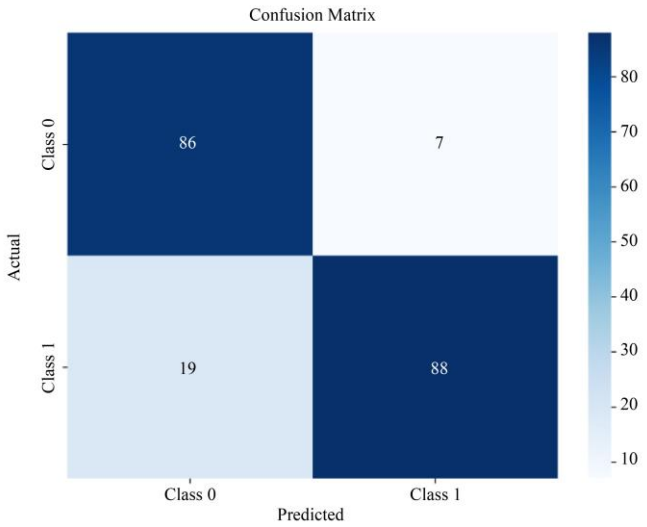

Fig. 2 Confusion matrix of the model

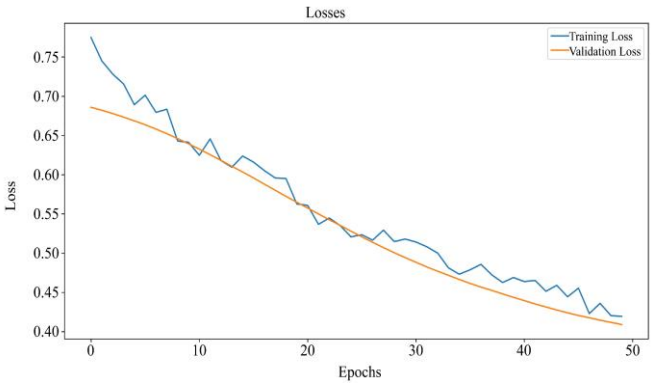

Fig. 3 Epochs vs Losses graph

The model demonstrated strong performance during the validation phase, achieving an accuracy of 87%, which indicates a reliable level of precision for the classification tasks. The analysis seen in the confusion matrix provided further insight into the classification outcomes. For Class 0, the model correctly predicted the 87 instances, with only 8 false positives, which demonstrates the capability to effectively identify this class with minimal misclassification. On the other hand, for Class 1, the model identified 87 instances correctly but showed a slightly higher error rate with 22 false negatives. The loss analysis for the training and validation cycles revealed the stable learning dynamics. The training loss steadily decreased, showing the model's capability to learn from the data effectively. The validation loss plateaued after a certain number of epochs, indicating the model reached its optimal performance without overfitting. The application of early stopping at epoch 23 helped maintain the balance between training and validation that ensures the model's generalizability.

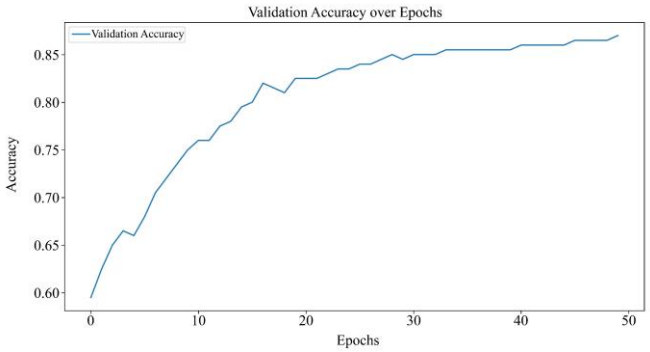

Fig. 4 Validation accuracy over epochs graph

The validation accuracy curve follows an upward trend over successive epochs that stabilize at approximately 87%. This reflects the model's consistency and capability in order to generalize well to unseen data. However, minor fluctuations toward the end suggest that further tuning the hyperparameters or model architecture is beneficial in enhancing stability and performance.

The proposed study was compared with the prominent work in the field of this particular domain, as shown in Table 1, including Naresh (2024) [31], which focused on privacy-preserving ML and research on blockchain-based security for IoT. The work focused more on balancing computational efficiency and security, which achieved an accuracy of 80.48%, which aligns closely with this study's accuracy of 87%. However, the AES-256 implementation presented here stands out for its lightweight efficiency in securing datasets while maintaining high compatibility with ML models.

Table 1. Comparative analysis based on traditional approaches

Study	Encryption Method	Model Type	Key Focus	Accuracy (%)	Remarks
Naresh (2024) [31]	Homomorphic Encryption	Privacy-preserving ML	Credit Risk Prediction	80.48	Incorporates homomorphic encryption for privacy management
Propose	ML-based AES-256	Neural Network (NN)	Secure machine learning dataset implementation	87	Demonstrates a lightweight, efficient approach with high compatibility

6. Conclusion

This study demonstrates the effective implementation of the ML-based model for the classification tasks that achieves an 87% accuracy. The results further validate the ability of the model to handle the complex datasets,enced by the balanced performance in both the phasning and the phases. The analysis

of training and the validation losses revealed that a steady improvement is seen in minimizing overfitting, which further highlights the stability of the proposed architecture. Furthermore, the confusion matrix provided in this study provides more valuable insights into the classification accuracy across different categories that showcase the model's robustness and reliability. Despite the promising results with this study, it remains an opportunity for further enhancements, particularly in reducing the misclassification rates and in order to optimize the architecture for improved generalization. Future research focuses on exploring the integration with advanced techniques like regularization, hyperparameter tuning, and ensemble methods to refine the model's accuracy and adaptability.

Overall, this work contributes to advancing NN-based solutions for classification problems and lays the foundation for further exploration and optimization in this domain.

References

[1] Jagpreet Kaur, Shweta Lamba, and Preeti Saini, "Advanced Encryption Standard: Attacks and Current Research Trends," *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, pp. 112-116, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] D.M.G. Preethichandra et al., "Passive and Active Exoskeleton Solutions: Sensors, Actuators, Applications, and Recent Trends," *Sensors*, vol. 24, no. 21, pp. 1-42, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yassine Himeur et al., "Applications of Knowledge Distillation in Remote Sensing: A Survey," *Information Fusion*, vol. 115, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Oussama Kerdjidj et al., "Uncovering the Potential of Indoor Localization: Role of Deep and Transfer Learning," *IEEE Access*, vol. 12, pp. 73980-74010, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Hamed Alqahtani, and Gulshan Kumar, "Machine Learning for Enhancing Transportation Security: A Comprehensive Analysis of Electric and Flying Vehicle Systems," *Engineering Applications of Artificial Intelligence*, vol. 129, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Siva Raja Sindiramutty, "Autonomous Threat Hunting: A Future Paradigm for AI-Driven Threat Intelligence," *arXiv Preprint*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Nazish Khalid et al., "Privacy-Preserving Artificial Intelligence in Healthcare: Techniques and Applications," *Computers in Biology and Medicine*, vol. 158, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Houda Hafi et al., "Split Federated Learning for 6G Enabled-Networks: Requirements, Challenges and Future Directions," *IEEE Access*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Krishnashree Achuthan et al., "Sustainable Cybersecurity Practices: Past Trends and Future Directions," *SSRN*, 2024. [Google Scholar]

[10] Rishabh Sharma, and Ashish Sharma, *Fake Account Detection using the Machine Learning Technique*, 1st ed., Smart Computing, pp. 197-203, 2021. [Google Scholar] [Publisher Link]

[11] Waymond Rodgers, *Artificial Intelligence in a Throughput Model: Some Major Algorithms*, 1st ed., CRC Press, Boca Raton 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Deafallah Alsadie, "Artificial Intelligence Techniques for Securing fog Computing Environments: Trends, Challenges, and Future Directions," *IEEE Access*, vol. 12, pp. 151598-151648, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Fadele Ayotunde Alaba et al., "Internet of Things Security: A Survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10-28, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mosiur Rahaman et al., "Privacy-Centric AI and IoT Solutions for Smart Rural Farm Monitoring and Control," *Sensors*, vol. 24, no. 13, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, Healthcare Information Research, The MIT Press, Cambridge, MA, USA, 2016. [Google Scholar]

[17] Sebastian Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv Preprint*, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Laith Alzubaidi et al., "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, no. 1, pp. 1-74, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nitish Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014. [Google Scholar] [Publisher Link]

[20] Kaiming He et al., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Alexander Kirillov et al., "Segment Anything," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015-4026, 2023. [Google Scholar] [Publisher Link]

[22] Samik Patel et al., "Use of Artificial Intelligence (AI) in Power Sector to Enhance Safety and Performance," *International Journal of Research and Analytical Reviews*, vol. 11, no. 2, 2024. [Google Scholar]

[23] Sunil Gupta, Hitesh Kumar Sharma, and Monit Kapoor, "Artificial Intelligence-Based Cloud Storage for Accessing and Prediction," *Blockchain for Secure Healthcare Using Internet of Medical Things (IoMT)*, pp. 157-168, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] W. Villegas-Ch, and J. García-Ortiz, "Toward a Comprehensive Framework for Ensuring Security and Privacy in Artificial Intelligence," *Electronics*, vol. 12, no. 18, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Harish Padmanaban, "Privacy-Preserving Architectures for AI/ML Applications: Methods, Balances, and Illustrations," *Journal of Artificial Intelligence General science (JAIGS)*, vol. 3, no. 1, pp. 235-245, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Keith Bonawitz et al., "Towards Federated Learning at Scale: System Design," *Proceedings of Machine Learning and Systems 1 (MLSys 2019)*, vol. 1, pp. 374-388, 2019. [Google Scholar] [Publisher Link]

[27] Harry Chandra Tanuwidjaja et al., "Privacy-Preserving Deep Learning on Machine Learning as a Service-A Comprehensive Survey," *IEEE Access*, vol. 8, pp. 167425-167447, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] James Bret Michael, "Security and Privacy for Edge Artificial Intelligence," *IEEE Security & Privacy*, vol. 19, no. 4, pp. 4-7, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[29] Davide Calvaresi et al., "Erebots: Privacy-Compliant Agent-Based Platform for Multi-Scenario Personalized Health-Assistant Chatbots," *Electronics*, vol. 10, no. 6, pp. 4-7, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Samir Vinayak Bayani, Sanjeev Prakash, and Jesu Narkarunai Arasu Malaiyappan, "Unifying Assurance: A Framework for Ensuring Cloud Compliance in AIML Deployment," *Journal of Knowledge Learning and Science Technology*, vol. 2, no. 3, pp. 457-472, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[31] Vankamamidi S. Naresh, and Ayyappa D, "PPDNN-CRP: Privacy-Preserving Deep Neural Network Processing for Credit Risk Prediction in Cloud: A Homomorphic Encryption-Based Approach," *Journal of Cloud Computing*, vol. 13, no. 1, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]