Research on the Application of Homomorphic Encryption and Federated Learning in the Internet of Vehicles Environment

Haowen Zheng

Academy of intelligent manufacturing, Jianghan University, Wuhan, China

Xpl1234567890@tzc.edu.cn

Abstract. The rapid growth of the Internet of Things has significantly increased data volumes, leading to heightened concerns over security risks such as data theft and leakage. As machine learning becomes increasingly integral to various applications, data security in training processes has emerged as a critical issue. The Internet of Vehicles (IoV), as a crucial branch of the IoT, faces particular challenges in securely and efficiently training data. While current machine learning frameworks enable fast and efficient data training in IoV environments, security risks remain a pressing concern. This study explores the use of a federated learning framework enhanced with homomorphic encryption to address these issues. The research involves simulating real-world environments to test the basic performance and feasibility of the selected framework in IoV applications. Additionally, the impact of homomorphic encryption on the framework's effectiveness is assessed. Finally, a comparative analysis with traditional machine learning frameworks demonstrates that the chosen federated learning framework, when combined with homomorphic encryption, offers superior efficiency and security in IoV scenarios. This study underscores the potential of integrating advanced encryption techniques in machine learning frameworks to enhance data security in the IoV.

Keywords: Internet of Things, Internet of Vehicles, Technology, Federated Learning.

1. Introduction

The Internet of Things (IoT) has revolutionized the way objects interact by connecting them and processing data through sensors, leading to advanced applications across various fields. In the medical industry, for instance, IoT enables automatic control of treatment environments, enhancing patient care. In circular business models, IoT’s tracking and monitoring capabilities significantly reduce energy consumption, while in smart cities, IoT data optimizes resource allocation and other urban functions [1]. The Internet of Vehicles (IoV) represents another crucial application area, linking data from sensors between vehicles, drivers, and the cloud to optimize driving routes, monitor driver behavior, and enhance vehicle safety [3].

Despite the advancements brought by IoV, significant security risks accompany its growth, making security a critical concern. Since 2010, more than 900 public security incidents related to connected vehicles have been reported, with attacks increasing in scale, frequency, and complexity [4]. Frequent issues like data leakage and tampering have severely hampered the development and trust in connected vehicles, underscoring the urgent need for secure and efficient machine learning solutions that can handle discrete data in IoV environments. Addressing these challenges is essential for the future of connected vehicle technology.

This study focuses on demonstrating the applicability of a selected machine learning scheme tailored for the secure, rapid processing of discrete data in the IoV context. By conducting experiments, this research aims to validate the feasibility of the chosen machine learning approach, ensuring that it can meet the demanding security and efficiency requirements of connected vehicle systems. The findings will contribute to advancing IoV technologies by offering a robust solution to current security challenges, thereby fostering safer and smarter vehicle networks.

2. Related Work

The data of the Internet of Vehicles includes the location information of connected vehicles themselves, as well as the location of non connected vehicles and other obstacles that affect vehicle movement detected by roadside devices; Including owner information, vehicle body information, vehicle speed information, and vehicle control information contained in the in vehicle network end; This includes information about connected cars and their surrounding environment, including images of the exterior and interior of the vehicle captured by visual sensors, examples of the distance between the vehicle and the roadside detected by radar, as well as information about traffic signals, and so on. These pieces of information have the characteristics of discretization, partially satisfying independent and identically distributed, privacy, and requiring high-speed processing, time-varying, etc. Therefore, the processing difficulty of these data is relatively high. Encryption technology is a technique that transforms plaintext data into a meaningless ciphertext by processing it. The recipient decrypts the ciphertext using the key provided by the encryptor to obtain the plaintext. Homomorphic encryption technology is a special encryption technique, and its uniqueness lies in the fact that the result obtained from processing encrypted data is consistent with the result obtained from processing decrypted original data. This means that processing encrypted data is equivalent to processing the original text. By utilizing this feature, user data can be processed under encryption, greatly improving the security of data processing. The current federated learning technology has made significant contributions to the path planning and resource management of the Internet of Vehicles. The improved method of traditional federated learning FedAVG has been applied in fields such as traffic flow prediction and weather prediction. The resource management of vehicles can rely on the combination of federated learning and deep reinforcement learning (DRL) to achieve intelligent traffic resource allocation. Therefore, federated learning currently has certain application examples, but its data security issues are still threatened. Through existing research, it has been found that some attackers can even infer the user's raw data from the parameters uploaded by the user. Moreover, because federated learning requires each edge device to train personal data and upload parameters to the terminal, the training time of each user also affects the update time of the final model. However, due to the different computing capabilities of each user, computing may require a large amount of resources. Once a certain resource is insufficient, it will seriously slow down the overall model training time, so the lightweighting of the model is also a major problem [5].

3. Methodology

The selected method is a new federated learning method based on a single-layer feedforward neural network. The key steps of the selected scheme will be briefly introduced next [6].

3.1. Data Preprocessing and Model Design

The optimal weight is usually obtained by iteratively minimizing the cost function in neural networks, and mean square error (MSE) is the most widely used cost function. The most intuitive and commonly used approach is to calculate the MSE at the network output by comparing the expected and actual outputs. Another approach is to minimize the measured MSE before the activation function which is before $\mathbf{X}^T\mathbf{b}$ and $\mathbf{Out} = f^{-1}(Out)$. (where X represents the dataset input through the input matrix, b represents the weight vector of the neural network parameters, where the weight values are biased, and Out represents the output), and then based on the L2 norm regularization term, to avoid overfitting of the model, the cost function is finally generated.

$$J(\mathbf{b}) = \frac{1}{2}\left[\left(\mathbf{F}(\overline{\mathbf{Out}} - \mathbf{X}^T\mathbf{b})\right)^T\left(\mathbf{F}(\overline{\mathbf{Out}} - \mathbf{X}^T\mathbf{b})\right) + k\mathbf{b}^T\mathbf{b}\right] \tag{1}$$

Among them, k is called the hyperparameter, and if k=0, the basic MSE is obtained. However, the cost function in (1) also has its drawbacks. Although the activation function is nonlinear, it is convex and its global optimum can be obtained through the closed solutions of X and Out. His solution b has a high computational complexity and is related to the number of input samples. Once the sample size is high, even if it is a non iterative method to determine b, it requires high computational requirements. To solve this problem, using singular value decomposition (SVD) for transformation can rewrite the conditional expression that d satisfies

$$(USVFX + kI)b = XFF\ Out \tag{2}$$

Among them, U is a rare orthogonal matrix, and S is a diagonal matrix with non-zero elements, called the singular value of S. The optimal solution of this equation is to provide the solution with the minimum error given the training set, as shown in equation (3)

$$b = U(SS + kI)UXFF\ Out \tag{3}$$

However, the scheme proposed in equation (3) is only applicable to centralized learning scenarios. In the original algorithm, matrices U and S were calculated by XF's centralized SVD. Research has found that SVD can also be calculated in an incremental and distributed manner. Therefore, by applying the incremental method, we can use the partial SVD calculated on all clients in the joint scenario to calculate XF's SVD.

$$\begin{aligned} \mathrm{SVD}(\mathbf{A}) &= \mathrm{SVD}([\mathbf{A}_1|\mathbf{A}_2|\ldots|\mathbf{A}_P]) \\ &= \mathrm{SVD}([\mathbf{U}_1\mathbf{S}_1|\mathbf{U}_2\mathbf{S}_2|\ldots|\mathbf{U}_P\mathbf{S}_P]) \end{aligned} \tag{4}$$

This SVD calculation scheme is robust to rounding errors and data corruption errors.

Finally, the factor is calculated using equation (5)

$$\begin{aligned} \mathbf{n} = \mathbf{XFF}\bar{\mathbf{d}} &= [\mathbf{X}_1|\mathbf{X}_2|\ldots|\mathbf{X}_P]\begin{bmatrix}\mathbf{F}_1\\ \mathbf{F}_2\\ \vdots\\ \mathbf{F}_P\end{bmatrix}\begin{bmatrix}\mathbf{F}_1\\ \mathbf{F}_2\\ \vdots\\ \mathbf{F}_P\end{bmatrix}\begin{bmatrix}\overline{\mathbf{Out}}_1\\ \overline{\mathbf{Out}}_2\\ \vdots\\ \overline{\mathbf{Out}}_p\end{bmatrix} \\ &= \mathbf{X}_1\mathbf{F}_1\mathbf{F}_1\overline{\mathbf{Out}}_1 + \cdots + \mathbf{X}_p\mathbf{F}_p\mathbf{F}_p\overline{\mathbf{Out}}_p \end{aligned} \tag{5}$$

Due to the lack of a process of aggregating client information through network transmission, only each user's own US and n are used, so this solution is currently private. Therefore, each client sends it to the coordinator mvector, who then aggregates it. Due to the aforementioned data security issues, homomorphic encryption is required before the mvector sends it to the coordinator to further enhance data security.

3.2. Application of Homomorphic Encryption Technology

Next, a brief introduction will be given to the homomorphic encryption technology of the selected scheme [7].

$$[[\mathbf{n}]] = [[\mathbf{n}]] + [[\mathbf{n}_p]] \tag{6}$$

[[.]] is a homomorphic operator that directly calls the encapsulation tool using the CKKS CH scheme. Since this scheme operates using matrices, it has a higher level of security.

3.3. Implementation and Optimization of Federated Learning

Briefly introduce the implementation of the selected federated learning scheme, which consists of m participants and an aggregation server (coordinator) to train an ML model. After each participant performs FL on the client to execute the above formulas (1) to (5), the basic parameters are obtained. Finally, the information is encrypted through (6) and sent to the coordinator. The coordinator then uses equations to gradually merge all the information and sends it back to each participant, completing the model update.

Next, provide pseudocode for the client and coordinator to facilitate readers in reproducing the results. As show in the table 1 and table 2.

Table 1. Algorithm 1 Client-side pseudocode

```
Algorithm 1 Client-side pseudocode
Client input section:
$\mathbf{X}_p \in \mathbb{R}^{m \times n_p}$
$\mathbf{d}_p \in \mathbb{R}^{n_p \times 1}$
f
Client output section:
$[[\mathbf{m}_p]]$
$\mathbf{US}_p$
function $\mathbf{FEDHEONN_{CLIENT}}(\mathbf{X}_p, \mathbf{d}_p f)$
    $\mathbf{X}_p = [ones(1, n_p); \mathbf{X}_p];$
    $\mathbf{d}_p = f^{-1}(\mathbf{d}_p);$
    $\mathbf{f}_p = f'(\mathbf{d}_p);$
    $\mathbf{F}_p = diag(\mathbf{f}_p);$
    $[\mathbf{U}_p, \mathbf{S}_p, \sim] = SVD(\mathbf{X}_p * \mathbf{F}_p);$
    $\mathbf{US}_p = \mathbf{U}_p * \mathrm{diag}\ (\mathbf{S}_p)$
    $\mathbf{m}_p = \mathbf{X}_p * (\mathbf{f}_p .* \mathbf{f}_p .* \mathbf{d}_p);$
    $[[\mathbf{m}_p]] = \mathrm{ckks\_encryption}(\mathbf{m}_p)$
    return$[[\mathbf{m}_p]], \mathbf{US}_p$
endfunction
```

Table 2. Algorithm2 Coordinator pseudocode

```
Algorithm2 Coordinator pseudocode
Input:
$\mathbf{M}$_list
client
US_list
$\lambda$
Output:
$[[\mathbf{w}]] \in \mathbb{R}^{m \times 1}$
$functionFEDHEON$N $COORDINATOR(M_l ist, US_l ist, \lambda)$
    if previous[[m]], USmatricesareavailable:
        $[[\mathbf{m}]] = \mathrm{Stored}[[\mathbf{m}]]$
        $US = Stored\ US$
    Else:
        $[[\mathbf{m}]] = \mathbf{0}$
        $\mathbf{US} = [\,]$
    for$[[\mathbf{m}_p]], \mathbf{US}_p$in$(\mathbf{M}_{\mathrm{list}}, \mathrm{US}_{\mathrm{list}})$:
        $[[\mathbf{m}]] = [[\mathbf{m}]] + [[\mathbf{m}_p]]$
        $[\mathbf{U}, \mathbf{S}, \sim] = \mathrm{SVD}([\mathbf{US} \mid \mathbf{US}_p]);$
        $\mathbf{US} = \mathbf{U} * \mathrm{diag}(\mathbf{S})$
    $[[\mathbf{w}]] = \mathbf{U} * inv(\mathbf{S} * \mathbf{S} + \lambda\mathbf{I}) * (\mathbf{U}^T * [[\mathbf{m}]])$
    Save [[m]].US
    return [[W]]
end function
```

4. Experiment and Results Analysis

The experiment is divided into three phased objectives. Firstly, basic performance testing is conducted on the selected model, analyzing accuracy and error rate to test the basic performance of the model; In addition, the homomorphic encryption module is used to test the performance of the model and assess the impact of homomorphic encryption on its performance; Finally, the model will be compared with traditional federated learning models to determine if it outperforms most models [8].

During the experiment, in order to observe the robustness of the model to the data, both independent and identically distributed (IID) data and non IDD data were used simultaneously. In order to study the impact of encryption on the model, the data was also divided into encrypted and unencrypted data. The number of clients in this experiment ranges from 200 to 20000, increasing in increments of 200 each time. Different experiments are conducted to observe whether the number of clients has a significant impact on the model. The training dataset used six large datasets with logistic function as the activation function for neurons. Due to limitations in conditions, these experiments were only simulated on one computer and did not run on 20000 independent computers (specifically, the training time refers to the time from the start of training to the last client data being uploaded to the coordinator and completed, as this is an experiment conducted on the same computer and not on multiple joint computers; CPU usage is reflected in the CPU training time and energy consumption (unit: Wh) during the experiment).

4.1. Experimental Setup and Parameter Configuration

The experiment was conducted in the Python 3.9 environment, equipped with the third-party library numpy for homomorphic encryption and federated learning, scipy,tenseal,pandas,scikit_ learn.

Experimental hardware environment: Intel (R) Core (TM) i7-1260P 2.10 GHz processor, 16GB RAM.

4.2. Performance Testing and Data Analysis

Tge expirmental shows the model is very good in accuracy and MSE.

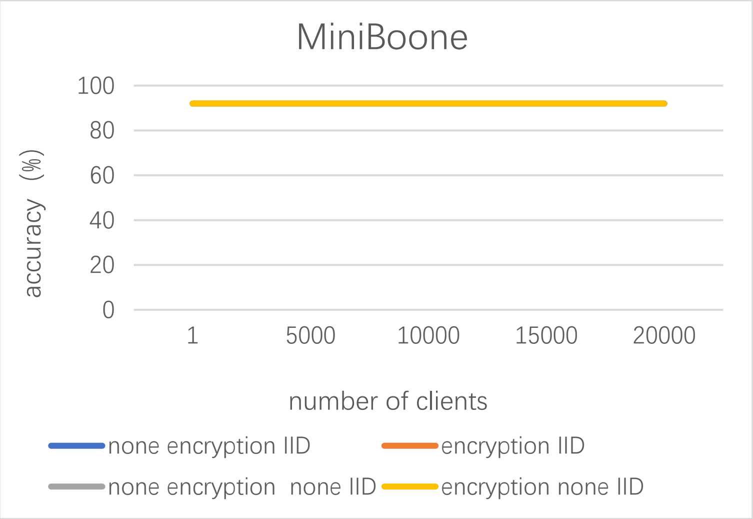

Figure 1. MiniBoone (Photo credit: Original)

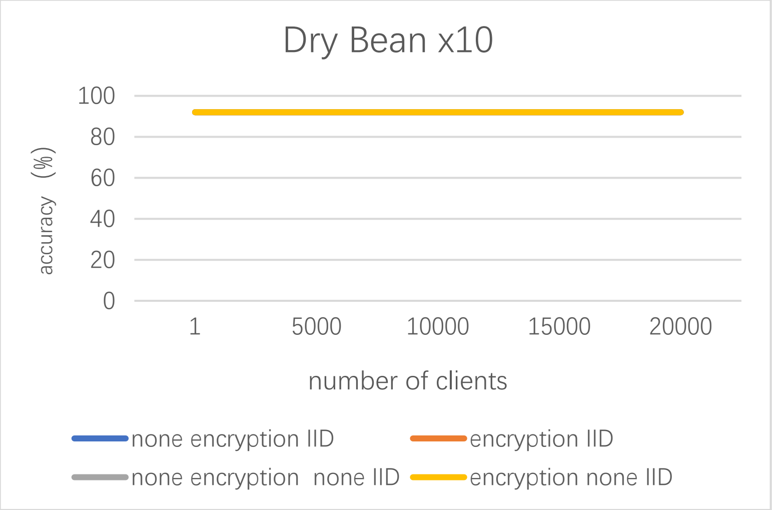

Figure 2. Dry Bean x10 (Photo credit: Original)

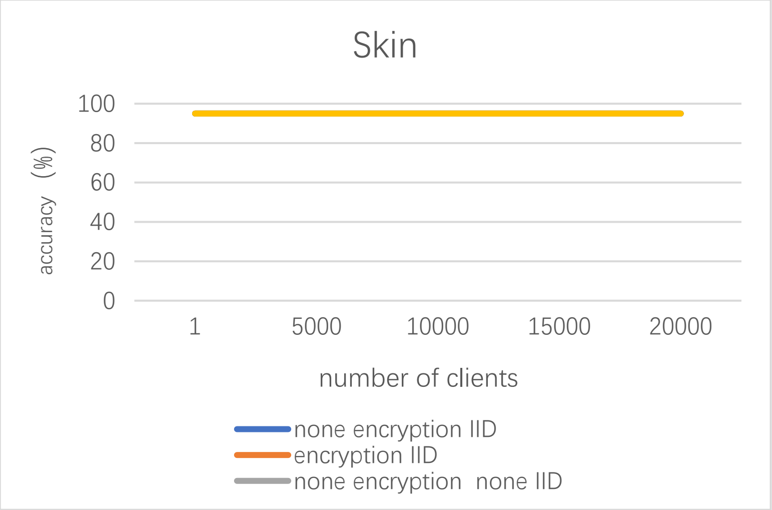

Figure 3. Skin (Photo credit: Original)

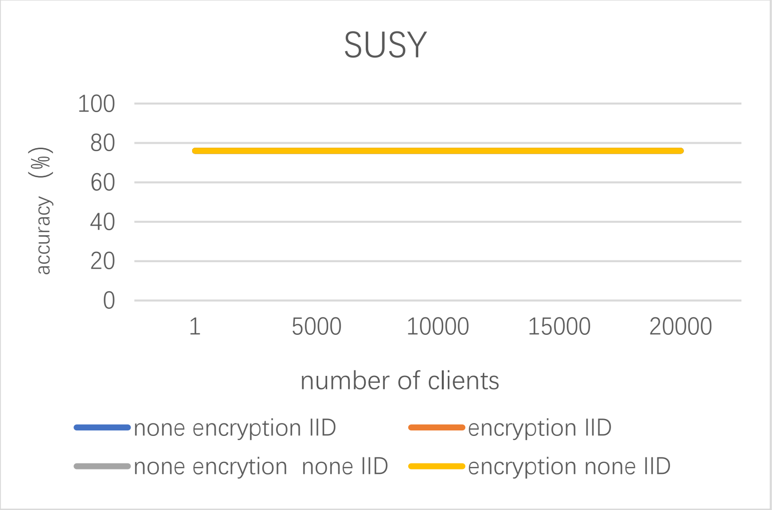

Figure 4. SUSY (Photo credit: Original)

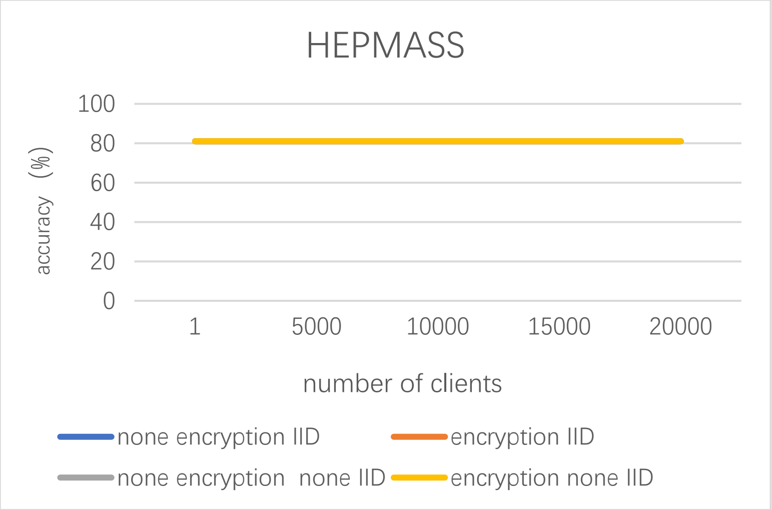

Figure 5. HEPMASS (Photo credit: Original)

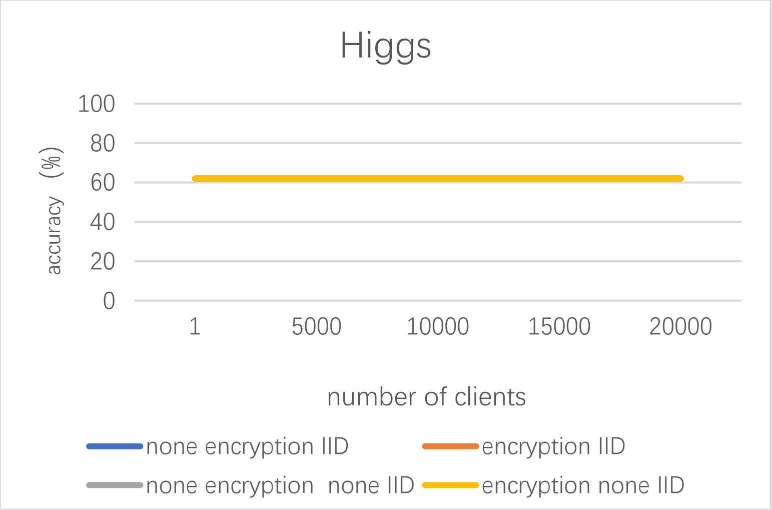

Figure 6. Higgs (Photo credit: Original)

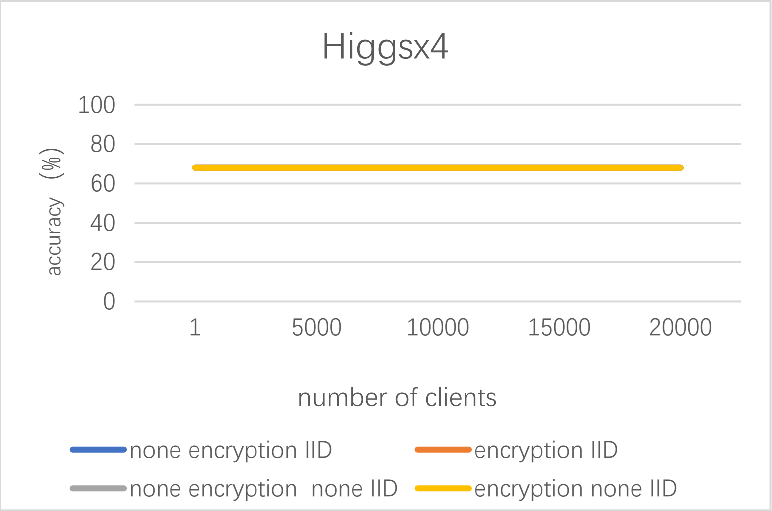

Figure 7. Higgsx4 (Photo credit: Original)

As show in thefigure 1 to the figure 7. Figure 1 shows the results of IID and non IID, encrypted and non encrypted data. Through experimental results, it was found that the accuracy of the four lines in these four cases (data encrypted and satisfying IID, data encrypted but not satisfying IID, data not encrypted and satisfying IID, data not encrypted but not satisfying IID) was very stable on the client side from 200 to 20000, and was basically not affected by changes in the number of client ends. Only one colored line could be observed on the graph because of the overlapping relationship of the four lines. This also means that regardless of whether the data is IID or non IID, the accuracy of model training will not be affected by data encryption and non encryption. Comparing the number of different clients in each large dataset before and after, it was found that the result is very smooth, indicating that the model Has good robustness to the number of clients.

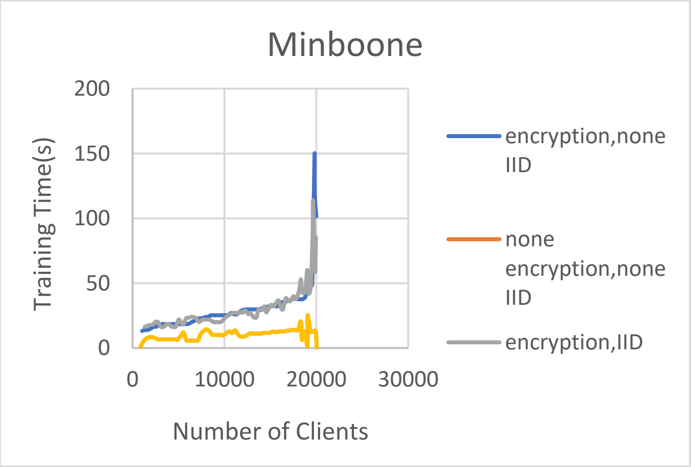

Figure 8. Minboone (Photo credit: Original)

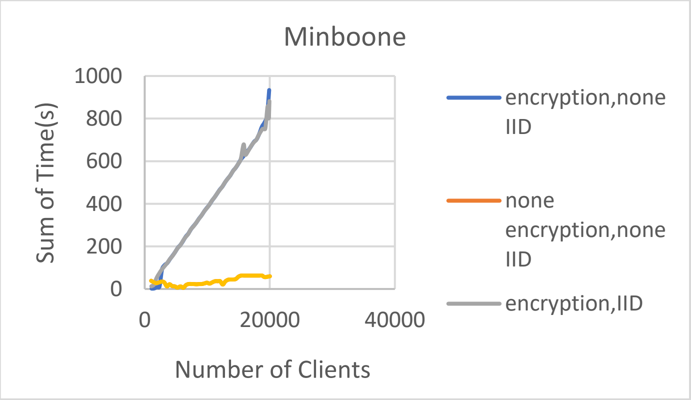

Figure 9. Experimental results chart (Photo credit: Original)

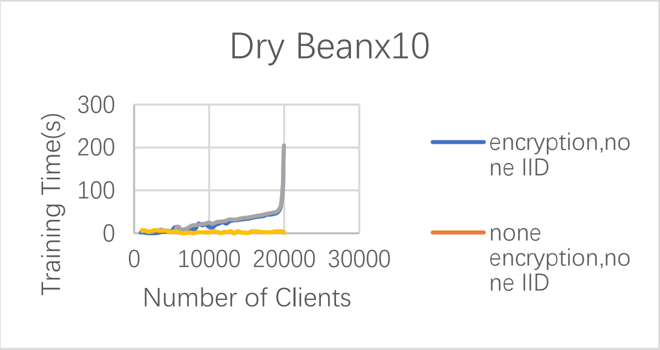

Figure 10. Experimental results chart 10 (Photo credit: Original)

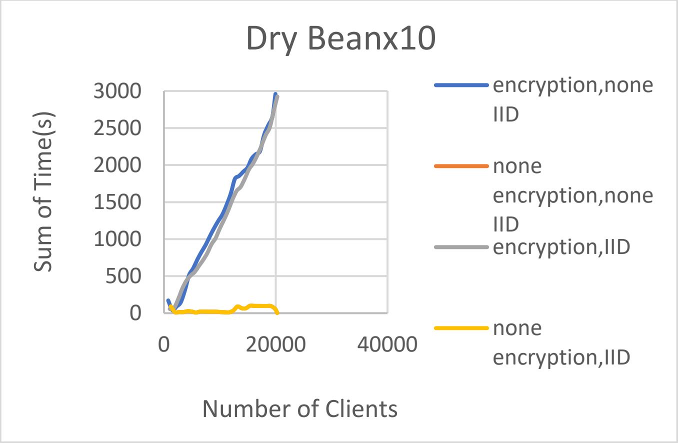

Figure 11. Experimental results chart 11 (Photo credit: Original)

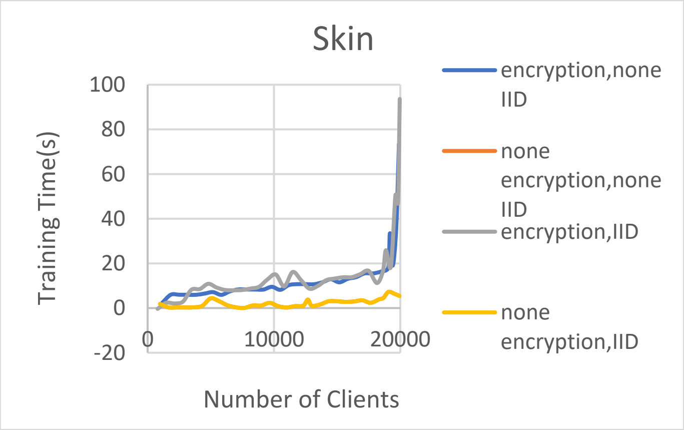

Figure 12. Experimental results chart 12 (Photo credit: Original)

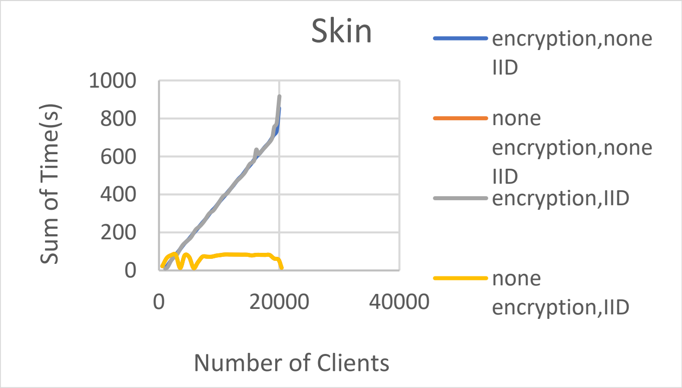

Figure 13. Experimental results chart 13 (Photo credit: Original)

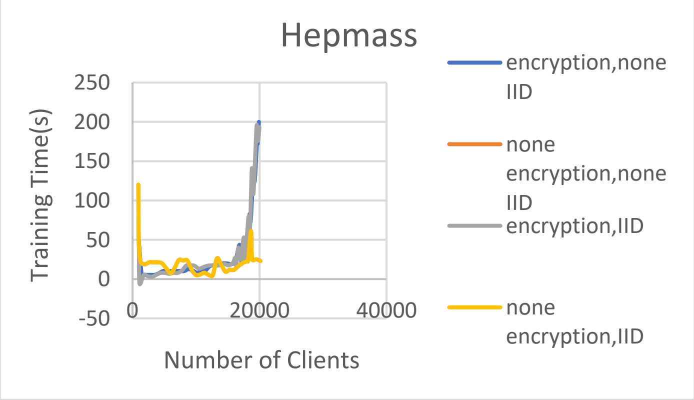

Figure 14. Experimental results chart 14 (Photo credit: Original)

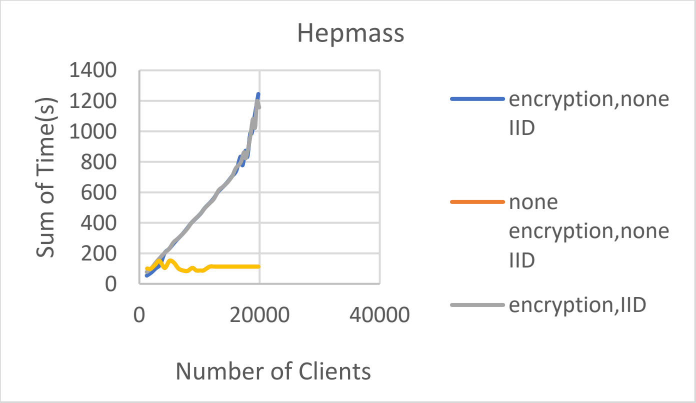

Figure 15. Experimental results chart 15 (Photo credit: Original)

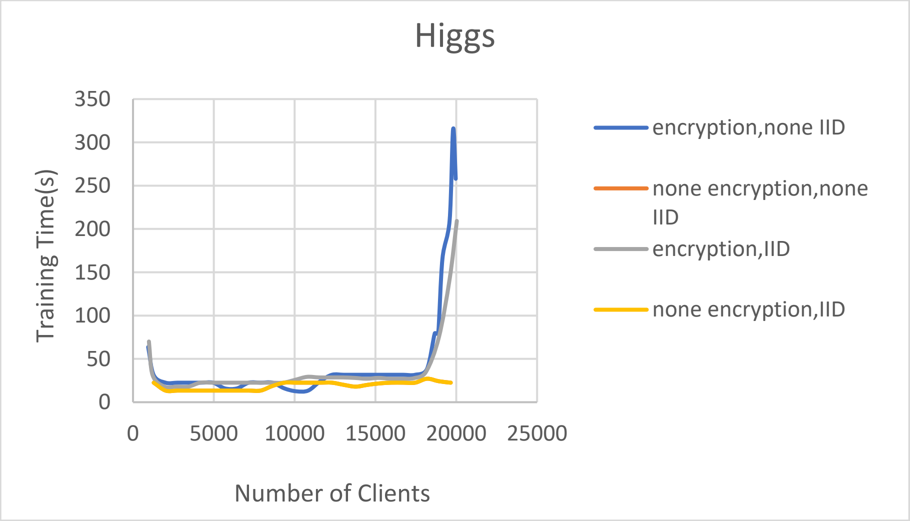

Figure 16. Experimental results chart 16 (Photo credit: Original)

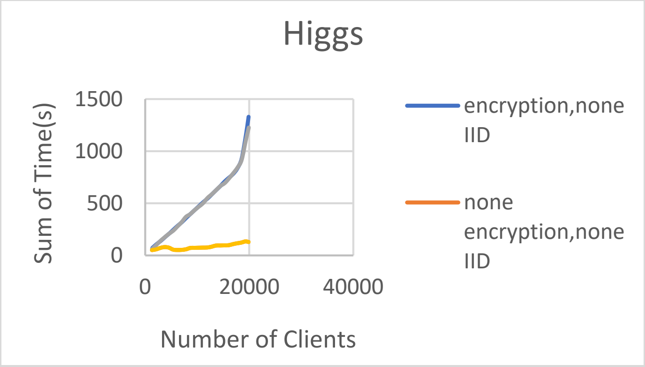

Figure 17. Experimental results chart 17 (Photo credit: Original)

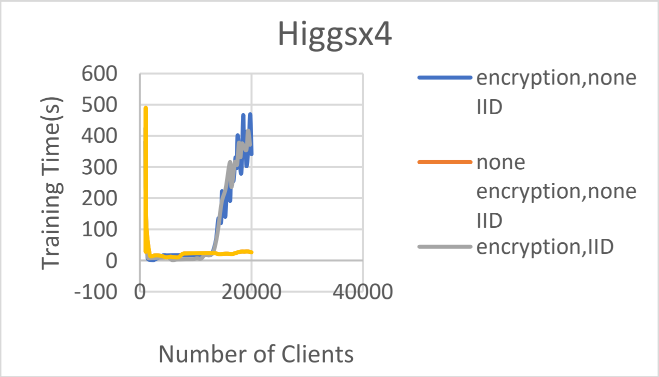

Figure 18. Experimental results chart 18 (Photo credit: Original)

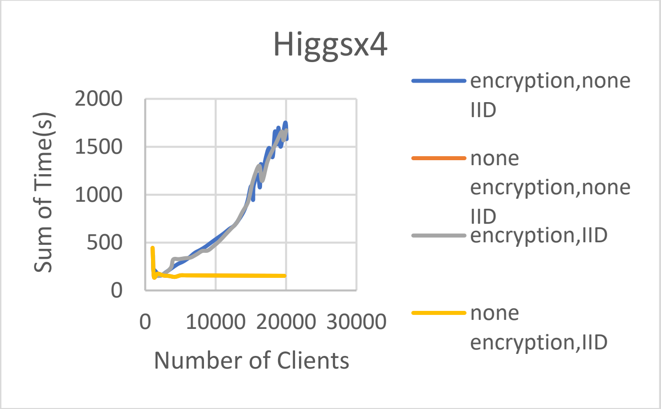

Figure 19. Experimental results chart 19 (Photo credit: Original)

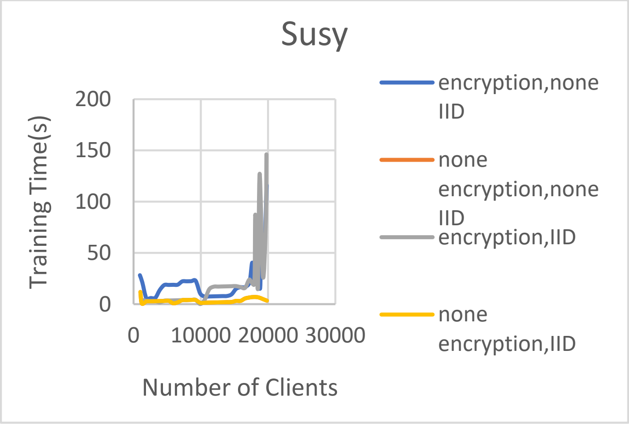

Figure 20. Experimental results chart 20 (Photo credit: Original)

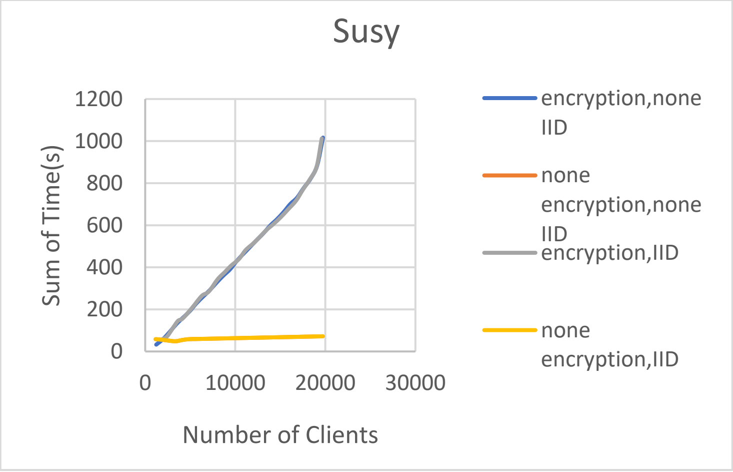

Figure 21. Experimental results chart 21 (Photo credit: Original)

As show in the figure 8 to figure 21. Figure 21 shows the time it takes for the model to train on various large datasets. It can be observed that the blue and purple curves are always below the yellow and green lines, which is due to the increased CPU energy consumption caused by the encryption step and the extra time spent encrypting data. However, it can also be observed that when the number of clients is less than 10000, the time required to train on these large datasets is relatively short, resulting in a faster training speed that can meet certain training needs in the context of the Internet of Vehicles.

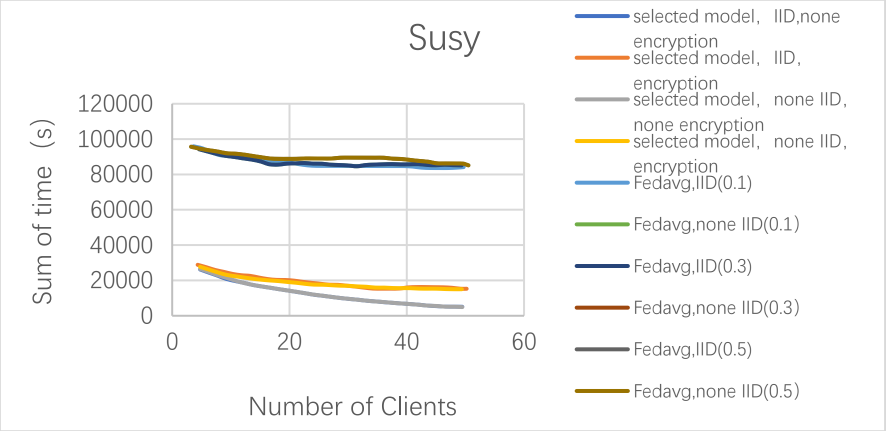

Figure 22. Experimental results chart 22 (Photo credit: Original)

As show in the table 3 and figure 22. And experiment also compared the training time of four data scenarios using the large dataset SUSY and the traditional federated learning method FedAVG, as shown in Figure 7. It was found that the selected model had better training speed than the traditional FedAVG in terms of training time, regardless of whether it faced IID or non IID types of data [9]. This indicates that the model can train faster than the traditional federated learning scheme FedAVG in the context of connected vehicles.

Table 3. Experiment comparison chart

Model	Higgs	Susy	HEPMASS
Selected Model	64.05	75.76	83.50
Single Layer neural network	64.17	78.80	83.64
Logistic Regression	64.21	-	-
Logistic Regression	-	78.84	
Spark K-means	48.34	50.04	50.66
Spark Generalized Linear Model	63.51	75.01	83.40
Decision Tree	63.57	-	-
Decision Tree	-	75.46	-
Random Forest	67.64	-	-
Random Forest	-	77.40	-
Random Forest	67.67	77.67	82.21
Spark.Random Forest	59.65	76.81	82.43
Rotation Forest	68.80	78.59	84.44
Gradient Boosted Tree	70.62	-	-
Gradient Boosted Tree	-	79.30	-
Spark Gradient Boosted Tree	59.49	75.11	81.83
PANFIS	63.94	75.42	83.32
PANFIS MapReduce	63.48	76.80	83.35
Scalable PANFIS Merging	63.66	76.70	83.47
Scalable PANFIS Voting	63.70	76.22	84.18
Scalable PANFIS AL Merging	63.72	76.79	83.45
Scalable PANFIS AL Voting	63.92	76.20	84.15
PDMS Genetic Algorithm	63.00	-	83.67
Max Mean Discrepancy	57.90	-	-
eTS	64.69	77.05	82.32
Simpl_eTS	60.17	70.93	81.22
MapReduce MRAC	62.96	74.57	-
PCA Random Discretization Ensemble	58.33	72.64	81.33
median	63.66	76.70	83.35
mean	62.92	75.06	81.12

As show in the table 3. Finally, comparing the accuracy of the selected model with other authors' published models trained on large datasets, it can be found that the accuracy is not significantly different from other models. However, in the test results of the three large datasets, the overall accuracy of the three tests is still more stable than that of the general model.

4.3. Discussion and Model Evaluation

Therefore, overall, through testing the model itself and comparing it with the traditional federated model FedAVG and other models published by authors, it is found that the selected model is relatively stable in accuracy, has high precision, and has a fast computing speed. It also has a homomorphic encryption module, and has strong robustness to the impact of the number of clients and data types on whether they meet IID requirements. Therefore, it is believed that the model can handle dispersed, fast computing, and high security data in the Internet of Vehicles well, and is a high-quality model that can handle Internet of Vehicles data well.

5. Discussion

Federated learning can train models without uploading user raw data, only uploading parameters and aggregating the results through a demodulator for model training. According to existing findings, attackers can tamper with user uploaded data to affect the results of federated learning, obtain the user's original data by uploading parameters, and so on [10]. Therefore, encrypting parameters through homomorphic encryption can make data more secure. Enhance the anti attack capability of federated learning. Of course, the trade-off between data security and computational efficiency is also important. Enhancing data security by encrypting data inevitably results in a loss of computational efficiency. However, the selected model, even after encryption, is still faster and more accurate than traditional federated learning FedAVG. Therefore, it can be considered that this model has an excellent trade-off between security and computational efficiency. So it can be basically considered that this model is an excellent model that can run in the Internet of Vehicles environment. However, through the above research, it can also be found that although the training speed of the model is fast when the number of clients is less than 10000, when the number of clients is greater than about 15000, the computation time of the model will significantly increase, which is also a problem that needs to be further solved.

6. Conclusion

This paper provides a comprehensive review of the current state of Wireless Sensor Networks (WSNs) in smart home environments, highlighting their significant potential and identifying the key challenges that hinder their widespread adoption. The study delves into the various applications of WSNs in smart homes, such as remote control, monitoring, and security, while also examining the integration of advanced technologies like artificial intelligence. Despite the promising developments, the research underscores the ongoing challenges in energy efficiency, protocol interoperability, and data privacy that need to be addressed to fully realize the potential of WSNs in smart homes. Looking forward, future research should focus on developing more energy-efficient algorithms, enhancing the security and privacy of data in WSNs, and establishing universal standards for protocol interoperability. Additionally, there is a need for interdisciplinary collaboration to tackle ethical concerns and ensure that the deployment of WSNs in smart homes is both technically robust and ethically sound. By addressing these challenges, the smart home industry can move closer to creating more secure, efficient, and user-friendly environments that fully leverage the capabilities of WSN technology.

References

[1] R. Salama, F. Al-Turjman, P. Chaudhary, S.P. Yadav, "Benefits of Internet of Things (IoT) Applications in Health care-An Overview," in 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), 2023, pp. 778-784.

[2] H. Rehan, "Internet of Things (IoT) in Smart Cities: Enhancing Urban Living Through Technology," J. Eng. Technol., vol. 5, no. 1, pp. 1-16, 2023.

[3] F. Chen, Z. Dong, J. Dong, M. Xu, "Review of security protection technologies for Internet of Vehicles," Telecommun. Sci., vol. 39, no. 3, pp. 1-15, 2023.

[4] O. Fontenla-Romero, B. Guijarro-Berdiñas, E. Hernández-Pereira, B. Pérez-Sánchez, "FedHEONN: Federated and homomorphically encrypted learning method for one-layer neural networks," Future Gener. Comput. Syst., vol. 149, pp. 200-211, 2023.

[5] X. Zhu, H. Xu, Z. Zhao, X. Wang, X. Wei, Y. Zhang, J. Zuo, "An environmental intrusion detection technology based on WiFi," Wireless Pers. Commun., vol. 119, no. 2, pp. 1425-1436, 2021.

[6] Y. Peng, Z. Chen, Z. Chen, W. Ou, W. Han, J. Ma, "Bflp: An adaptive federated learning framework for internet of vehicles," Mobile Inf. Syst., vol. 2021, no. 1, art. 6633332, 2021.

[7] N.M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni, F. Karray, "Secure federated learning with fully homomorphic encryption for IoT communications," IEEE Internet Things J., 2023.

[8] H. Mun, K. Han, E. Damiani, T.Y. Kim, H.K. Yeun, D. Puthal, C.Y. Yeun, "Privacy enhanced data aggregation based on federated learning in Internet of Vehicles (IoV)," Comput. Commun., vol. 223, pp. 15-25, 2024.

[9] M. Han, K. Xu, S. Ma, A. Li, H. Jiang, "Federated learning-based trajectory prediction model with privacy preserving for intelligent vehicle," Int. J. Intell. Syst., vol. 37, no. 12, pp. 10861-10879, 2022.

[10] Z. Du, C. Wu, T. Yoshinaga, K.L.A. Yau, Y. Ji, J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," IEEE Open J. Comput. Soc., vol. 1, pp. 45-61, 2020.