A Paillier Homomorphic Encryption-Based Lightweight Privacy Protection Model for Mobile Crowd Sensing Networks

Shijie Feng[1*], Yanyan Qin[2,] Zhixiang Zeng[1], Bing Wen[1], Weijie Zhong[1], Ning Wang[2], Wei Guo[2], Yu'nan Zhang[3]
[1]Hainan Power Grid Co., Ltd., Haikou 570203, China
[2]Information and Telecommunication Branch of Hainan Power Grid, Haikou 570203, China
[3]CSG Electric Power Research Institute Co., Ltd., Guangdong Provincial Key La-boratory of Power System Network Security, Guangzhou 510663, China
Email: fengshijie198811@163.com, qinyy@hn.csg.cn, zengzx2@hn.csg.cn, wb@hn.csg.cn, zhongwj@hn.csg.cn, wangn3@hn.csg.cn, guow@hn.csg.cn, zhangyn2@csg.cn
*Corresponding author

Keywords: MCSN, homomorphic encryption, paillier, DNN, network security, privacy preservation

Received: July 16, 2024

The objective of this study is to enhance the privacy protection capability of mobile crowd perception networks and improve the detection accuracy of security threat data. To this end, a three-level collaborative distributed architecture is designed, which combines the concept of zero trust and proposes a lightweight security threat detection model based on the Paillier homomorphic encryption, deep neural networks, and box graph methods. Firstly, by optimizing the mobile crowd sensing networks framework and introducing three different network structures, a collaborative distributed system of terminal edge cloud was constructed. Secondly, the Paillier homomorphic encryption algorithm was designed to protect data privacy. The experimental results showed that the designed model achieved a detection accuracy of 98.84%, a mean square error of 0.03, and an average detection time of 0.15 seconds for A-class threats. In terms of processing efficiency, this model significantly improved data transmission efficiency, had lower computational overhead, and was suitable for various types of security threat detection. Therefore, the security threat detection model proposed in this study provides effective privacy protection technology for mobile crowd-sensing networks, significantly improving network security.

Povzetek: Razvit je nov model zaščite zasebnosti za mobilna senzorska omrežja, ki združuje Paillierjevo homomorfno šifriranje in globoke nevronske mreže.

1 Introduction

With the popularization of smart devices and the advancement of IoT technology, mobile crowd sensing networks (MCSN) have been widely used in the fields of environmental monitoring, public safety, and intelligent transportation [1]. The current MCSN collects data through mobile terminal devices and processes and analyzes it through a cloud platform, which greatly improves the efficiency of data collection and processing [2]. However, with the continuous growth of data size, data privacy preservation (PP) becomes an important problem to be solved. Traditional PP methods, such as data anonymization and encrypted transmission, have been difficult to meet the increasingly complex security requirements [3-4]. In this context, homomorphic encryption (HE) techniques have gradually attracted the attention of researchers due to their ability to directly compute data in the encrypted state. Among them, Paillier homomorphic encryption (PHE) algorithm is widely used in the field of PP for its excellent additive homomorphic property. However, it is difficult for a single encryption method to cope with complex and changing security threats. In order to better protect data privacy and improve the accuracy of threat detection at the same time, researchers have proposed comprehensive schemes combining multiple techniques. Zheng et al. proposed an HE-based grid matching scheme. The scheme first encrypted the grid used for task assignment so that the task matching process took place in an encrypted environment. Second, the location information of the applicant and the publisher kept secret from each other, thus protecting location privacy. Finally, the applicant fed back the results of the tasks in the grid, and the publisher received these results to complete the whole crowd sensing process. Research results demonstrated that this scheme outperformed other similar schemes in terms of performance and security [5]. With the rapid proliferation of mobile applications, mobile crowd sensing became an increasingly important topic, where the anonymity of the participants was crucial for network security. Ganjavi and Sharafat proposed an efficient edge-assisted mobile crowd sensing scheme that protected the privacy and anonymity of participants while

guarding against adversaries and verifying the authenticity of aggregation results. In this scheme, the joining and leaving problems were transparent and the computational cost and communication overhead were fixed and independent of the crowd size. The results of the study indicated that the scheme was able to identify and block malicious adversaries while providing anonymity to ordinary participants [6]. With the rapid increase in the amount of data generated by industrial devices in the Internet of Things (IoT), Jia et al. designed a blockchain-supported federated learning application model for industrial IoT and proposed a data protection aggregation scheme. The study introduced multiple data clustering methods based on differential privacy and HE to achieve multiple protections in data and model sharing. The final experimental results demonstrated that the proposed scheme performed well on various metrics [7]. Currently, encryption-based PP collaborative filtering is widely used for generating recommendation tasks. However, the existing solutions are slow and not scalable. To solve this problem, Jumonji et al. proposed the privacy-preserving collaborative filtering protocol based on BGV fully homomorphic encryption (BGV-CF). The results indicated that BGV-CF significantly simplified the recommendation process and improves the recommendation speed by reducing the interaction and communication traffic between users and recommendation servers [8].

In summary, although existing research has made some progress in data processing and PP, there are still some shortcomings. First, a single PP technique is difficult to cope with complex and changing security threats, and the existing methods are inefficient in processing large-scale data and have a large computational overhead in the encryption and decryption process. Second, the existing threat detection models still need to be improved in terms of real-time and accuracy. Especially in the face of large-scale mobile crowd sensing data, the existing scheme has a bottleneck in detection performance. To address the above shortcomings, the research proposes a lightweight security threat detection model (STDM) combining PHE, deep neural network (DNN), and box plot method (BPM). There are two innovations in the research. First, the PHE algorithm is used, which is able to maintain high encryption and decryption efficiency while protecting data privacy. Second, through the terminal-edge-cloud (TEC) three-tier collaborative architecture, it aims to improve the real-time and accuracy of threat detection. It is hoped that this research will provide new techniques and methods for MCSN PP. The summary of related work is shown in Table 1.

Table 1: Summary of related work

Researcher	Methodologies	Key performance metric	Limitation
Zheng et al. [5]	HE-based grid matching scheme; encrypted task assignment; location PP	Superior performance and security	Potential computational overhead due to encryption
Ganjavi et al. [6]	Efficient edge-assisted MCSN; privacy and anonymity of participants	Ability to identify and block malicious adversaries; fixed computational cost and overhead	Scalability with varying crowd sizes not addressed
Jia et al. [7]	Blockchain-supported federated learning for industrial IoT; differential privacy & HE	Performs well on various metrics, multiple protections in data and model sharing	Potential complexity in implementation
Jumonji et al. [8]	BGV-CF	Simplified recommendation process; improved speed by reducing interaction/communication	Scalability issues

2 Methods and materials

To improve the PP in MCSN, the research designs the MCSN framework in conjunction with the zero-trust concept (ZTC) that considers the secure access of mobile users. To address the threat data security detection in MCSN framework, the research further builds STDM jointly with lightweight HE algorithms Paillier, DNN, and BPM.

2.1 Design of MCSN framework based on zero trust concept

Crowd sensing network (CSN) is a network that utilizes a large number of users to collect data and share data through various devices, such as smartphones and wearable devices [9]. CSN focuses on collecting data through voluntary or organized user participation, which can be either static or dynamic. However, with the development of various intelligent technologies, the data

collection method is gradually changing from static collection to dynamic collection, which consequently also causes the problem of data explosion. In the field of network privacy and security protection, MCSN, as a subset of CSN, has been widely used in more practical scenarios because of its efficient data dynamic collection capability [10-11]. The network framework of MCSN is shown in Figure 1.

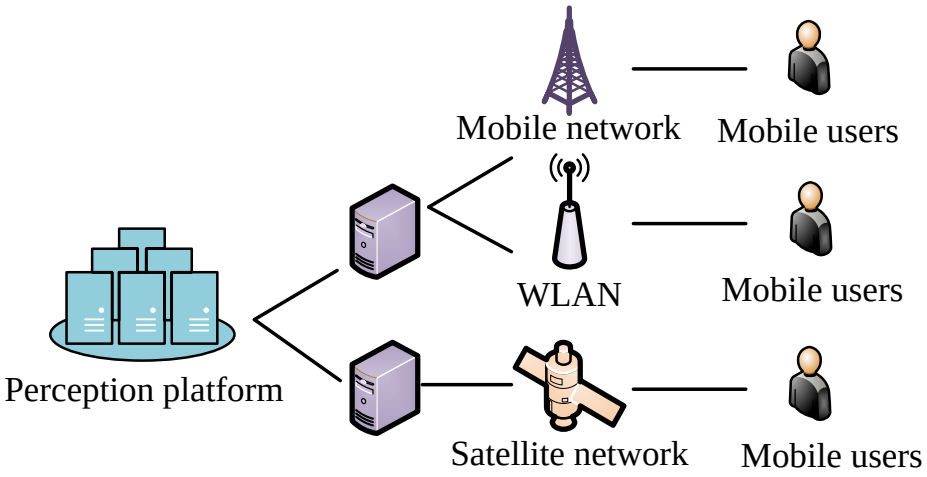

Figure 1: Network structure diagram of MCSN

The MCSN illustrated in Figure 1 utilizes mobile terminals as a source of data acquisition to construct a perception network. Task allocation and data aggregation are realized through cooperation between different communication networks. In the MCSN architecture, there are two main parts: the perception cloud platform with cloud server clusters and mobile users. The sensing cloud platform is responsible for the processing and analysis of the collected data, while the mobile users are tasked with the collection of data such as geographic information, speed, and pressure through the sensors of their devices. These devices then connect to the sensing cloud platform through various forms of wireless connectivity, including Wi-Fi, 4G, and 5G, and ultimately upload the sensing data. The access of a large number of mobile users fuels the development of MCSNs, but also introduces new security risks. On the one hand, users may fraudulently obtain rewards through Sybil attacks or duplicate submission of reports, which reduces the quality of reports and affects the effectiveness of the platform. On the other hand, users who are data consumers may engage in overstepping access or data corruption when accessing resources, threatening the security of platform resources [12]. In addition, users may use vulnerable devices to access the network, further exacerbating security risks. The common types of security threats in MCSN are shown in Table 2.

In Table 2, the types of security threats are mainly six categories: system attack, illegal intrusion, false identity, Sybil, malware, and various types of network Trojans, which are numbered as A, B, C, D, E, and F. In order to detect these security threats more effectively, the study introduces the ZTC to build the brand new MCSN. The structure of the security model using the ZTC is shown in Figure 2.

Table 2: Security threat types in MCSN

Types of security threats	Explaination	Threat type number
System attack	Malicious users can access system resources through remote control and eavesdropping	A
Trespassing	The user accesses an unauthorized resource	B
False identity	The user accesses an unauthorized resource	C
Sybil	Users declare multiple illegal identities in the sense cloud platform	D
Malware	Scripts or programs that control software processes exist on the access device	E
Various network trojans	The access device contains various network Trojans	F

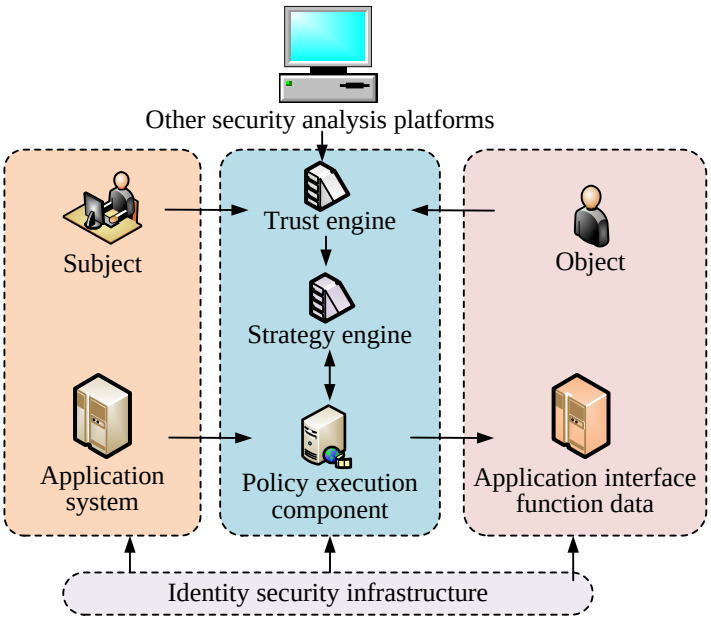

Figure 2: Frame diagram of zero trust concept

In Figure 2, the zero-trust architecture consists of four main components, namely the identity security infrastructure, the policy enforcement component, the policy engine, and the trust engine. The identity security infrastructure manages the identity and privileges of entities, including identity digitization, privilege auditing, and verifying the validity of authentication information. For users, it provides authentication techniques based on multiple factors, and for devices, only devices with installed identity credentials can access system resources, and those without will be denied [13-14]. The policy enforcement component decides the release or denial of user access requests based on the real-time authorization policy of the policy engine. The policy engine develops access control policies to assess the legitimacy of resource requests. The trust engine evaluates the user's trust level, based on behavioral logs and identity information, and improves the trustworthiness of trust metrics using big data and artificial intelligence techniques. The MCSN incorporating ZTC is notated as ZT-MCSN and its framework diagram is shown in Figure 3.

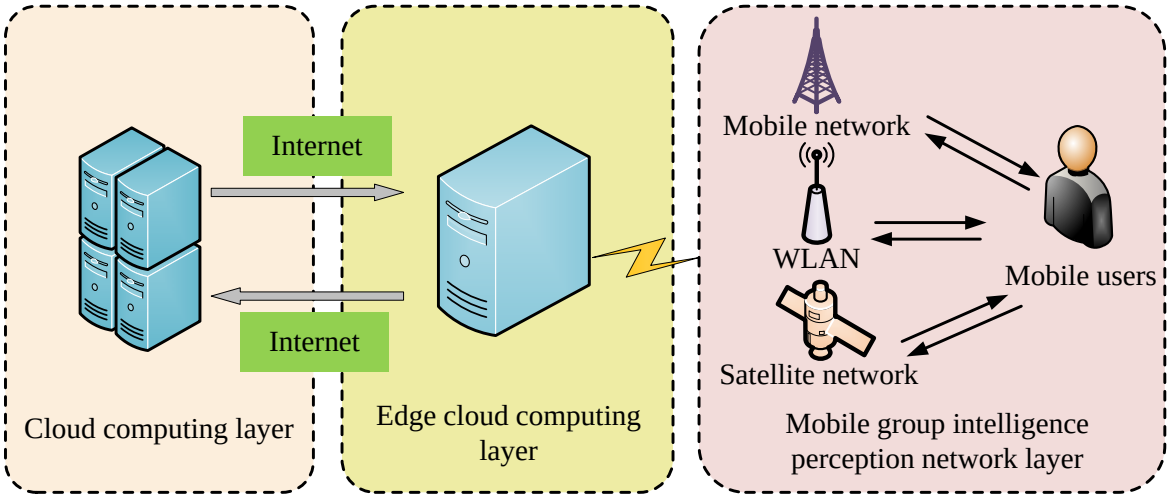

Figure 3: ZT-MCSN frame structure diagram

The ZT-MCSN network architecture in Figure 3 is divided into the MCSN network layer, edge cloud computing layer, and cloud computing layer from bottom to top. The MCSN network layer includes a large number of mobile users and a variety of communication networks, which are responsible for performing sensing tasks, local security model training and inference. In addition, the MCSN network layer has a variety of sensing nodes, including smart terminals and various types of mobile devices, and the mobile users collect data

through built-in sensors and upload data through communication technologies such as Wi-Fi and 5G. The edge cloud computing layer is located at the edge of the perceptual network and provides perceptual task processing and zero-trust security services by utilizing the computing and storage capabilities of edge servers. This layer is mainly responsible for the management of sensing tasks and secure access control of mobile users, ensuring network security through authentication and dynamic access control. The cloud computing layer consists of distributed server clusters that handle high-complexity services such as perceptual application management and resource scheduling to optimize perceptual activities and enhance user services.

2.2 Security threat detection model construction based on improved PHE

In the ZT-MCSN network, the distributed architecture of "TEC" is generally adopted. This architecture can distribute the training and inference tasks of STDM to each mobile device. By executing model training and reasoning locally, it effectively reduces the data transmission in the network and improves the data PP. The three-level collaborative distributed architecture is shown in Figure 4. The specific components of the "TEC" three-level collaborative distributed architecture are given in Figure 4. Although this architecture can ensure data privacy to a certain extent, distributed machine learning itself cannot completely ensure data privacy, and the insufficient generalization of the model may also lead to reverse attacks, which may leak device data [15]. In ZT-MCSN, device data privacy leakage mainly occurs in two links. One is during the transmission of model parameters; the attacker may obtain the parameters through eavesdropping and utilize them for reverse attacks. The second is that in the process of model parameter aggregation, there may be servers accessing device data through reverse analysis of the aggregated model. For this reason, the research combines PHE algorithm, DNN, and BPM in ZT-MCSN to design a new distributed training technique to enhance the protection capability of STDM for device data. The final constructed STDM is denoted as STDM with improved PHE (Paillier-deep neural network-box plot method, Paillier-DNN-BPM), and the structure diagram of its distributed training framework is shown in Figure 5.

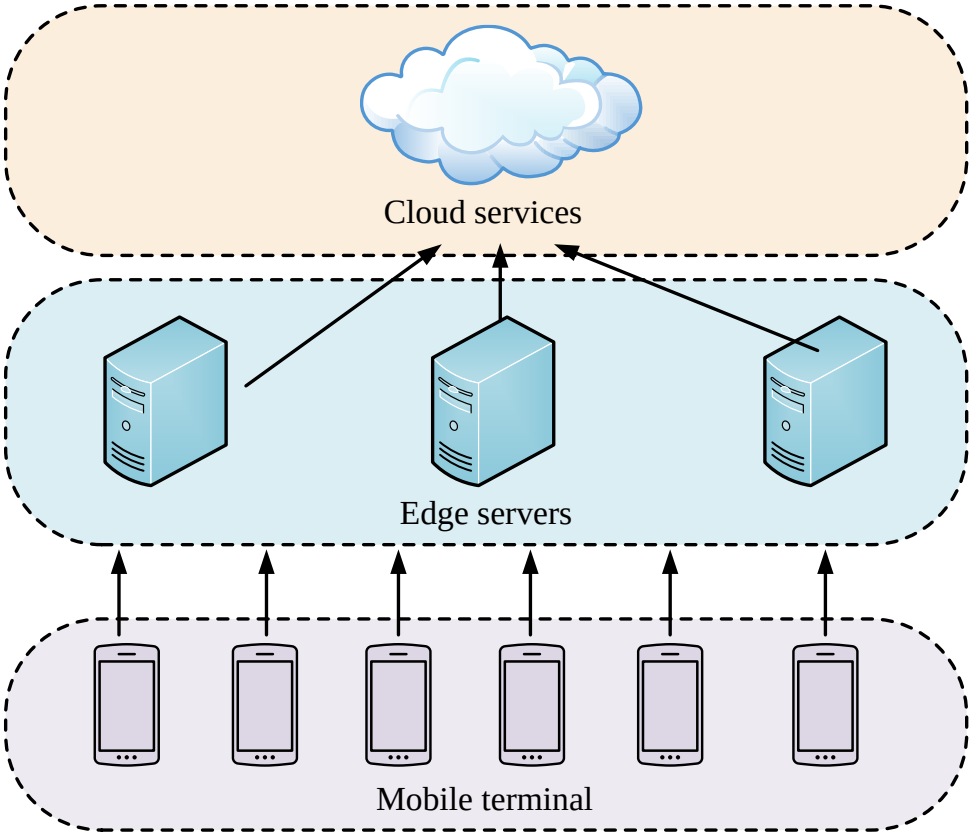

Figure 4: Three-level collaborative distributed architecture diagram

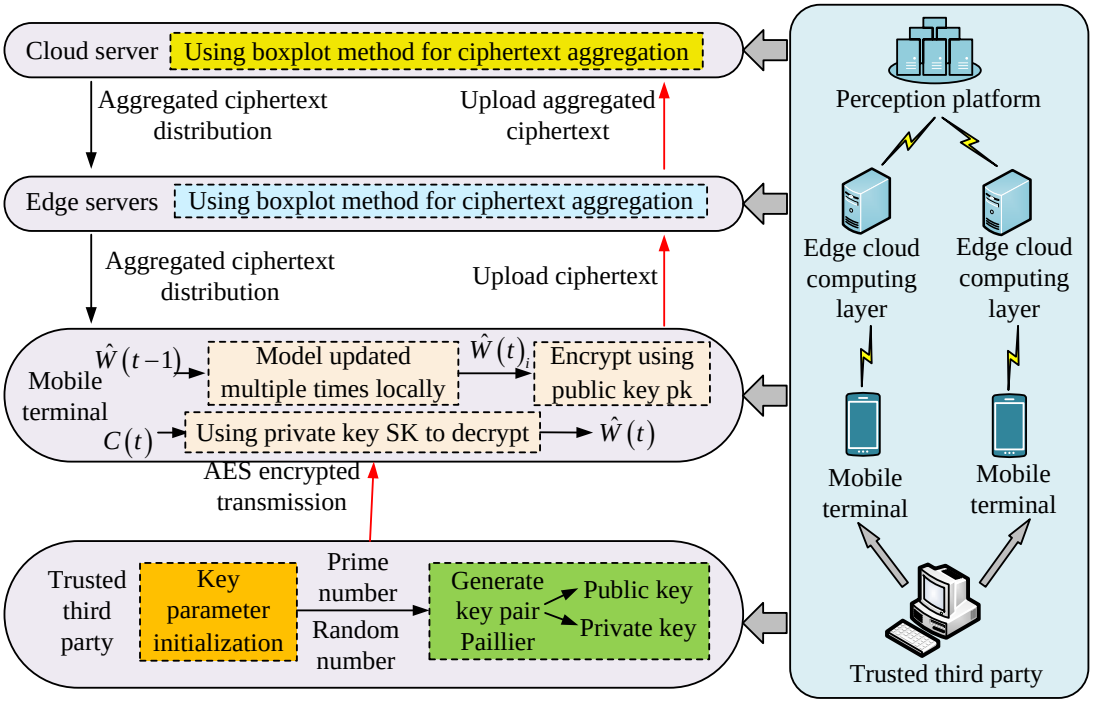

Figure 5: Paillier-DNN-BPM structure diagram

In Figure 5, the distributed training architecture of STDM Paillier-DNN-BPM, which combines PHE algorithm, DNN, and BPM, mainly consists of four parts: trusted third party, mobile terminal, edge server, and cloud server. The mobile terminal acts as a working node to decrypt the model parameters distributed by the server using the private key of the Paillier algorithm, and utilizes the local data for model training and parameter updating. The updated model parameters are encrypted by the public key of Paillier algorithm and uploaded to the edge server. The edge server, as a local parameter server in the distributed model training architecture, is responsible for filtering and aggregating the locally updated ciphertexts submitted by the mobile terminals, and uploading the local aggregation results to the cloud server for further computation. As the global parameter server in the distributed model training architecture, the cloud server is mainly responsible for initializing the model parameters and filtering and aggregating the local aggregated ciphertexts submitted by the edge servers to generate the global model parameter ciphertexts. The generated parameter ciphertexts are distributed to mobile terminals through the scheduling edge servers. In addition, to ensure that the encryption keys are only shared among mobile devices, the model architecture introduces a trusted third party. The trusted third party securely shares the key pairs of the Paillier algorithm among mobile devices via the advanced encryption standard (AES) algorithm. The Paillier algorithm is a packet cipher based HE algorithm. The algorithm is additively homomorphic, which means that encrypted data can be added without first being decrypted, and the result of the operation remains encrypted. In Paillier's algorithm, it is assumed that p and q are two different large prime numbers. Based on these two different large prime numbers, the formulae for the modulus n and the parameter λ can be obtained as shown in Equation (1) [16-17].

$$\begin{cases} n = p \cdot q \\ \lambda = lcm(p-1, q-1) \end{cases} \tag{1}$$

In Equation (1), $lcm(\cdot)$ denotes the least common multiple of $p-1$ and $q-1$. Setting a certain random number as g, $g \in Z_{n^2}^*$, the formula for the parameter u is obtained as shown in Equation (2).

$$u = \left(L\left(g^{\lambda} mod n^2\right)\right)^{-1} \tag{2}$$

In Equation (2), $L(\cdot)$ is a function used to calculate the intermediate value in the decryption process. n^2 denotes the modulus square, which is used to define the modulus space of the encryption operation. mod denotes modulo operation. In the Paillier encryption algorithm, the modulus operation is used to keep the value within a

specific range, preventing the value from being too large, while ensuring the correctness and consistency of the calculation results. The expressions for the public and private keys of Paillier algorithm are shown in Equation (3).

$$\begin{cases} p_k = (n, g) \\ s_k = (\lambda, u) \end{cases} \quad (3)$$

In Equation (3), p_k and s_k denote the public key and private key respectively. Setting x and y as two different plaintexts, combining Equation (1) ~Equation (3) can get the encryption process of Paillier's algorithm as shown in Equation (4).

$$\begin{cases} x' = E(x) = g^x r^n \bmod n^2 \\ y' = E(y) = g^y r^n \bmod n^2 \end{cases} \quad (4)$$

In Equation (4), x' and y' denote the plaintext x and y encrypted ciphertexts, respectively. $E(x)$ and $E(y)$ denote the encryption function respectively. r is some random number and $r < n$. Further the decryption process of the ciphertext is obtained as shown in Equation (5).

$$\begin{cases} x = D(x') = L\left(x'^{\lambda} \bmod n^2\right) * u \bmod n \\ y = D(y') = L\left(y'^{\lambda} \bmod n^2\right) * u \bmod n \\ L(a) = a - 1/n \end{cases} \quad (5)$$

In Equation (5), $D(x')$ and $D(y')$ denote the decryption function, respectively. $L(a)$ denotes a function in the decryption process. a denotes the variable of that function.

In Paillier's algorithm, its additive homogeneity is manifested as a multiplication computation on the ciphertexts of plaintexts x and y. The result obtained from the computation is decrypted using a key and the decrypted plaintext is obtained. The former encrypted and decrypted content is the same as the result of the addition operation performed directly on the plaintext. The expression for additive homogeneity is shown in Equation (6).

$$E(x) \times E(y) = E(x + y) \quad (6)$$

In Equation (6), $E(x+y)$ denotes the addition operation directly on the plaintext. Paillier has addition homogeneity in addition to number multiplication homomorphism. Number multiplication homomorphism means that the z-power computation is performed on the ciphertext of x, and the result of the computation is the same as that of the number multiplication z computation on the plaintext x after decryption using the key. The expression for the number multiplication homomorphism is shown in Equation (7).

$$E(x \times z) = E(x)^z \quad (7)$$

In Equation (7), z denotes the number of times a certain power square is calculated and is a constant. Based on the generated key pairs, the study utilizes trusted third-party AES encryption algorithms to achieve secure sharing of data across devices.

In DNN, it is assumed that l denotes the neuron of layer l in the DNN, $2 \le l \le 4$. In this way, the weights and biases of the neurons of layer l can be obtained as $W^{(l)}$ and $BS^{(l)}$, respectively. Let the output of the previous layer be $OP^{(l-1)}$ to get the output of layer l as shown in Equation (8) [18].

$$OP^{(l)} = f\left(W^{(l)} OP^{(l-1)} + BS^{(l)}\right) \quad (8)$$

In Equation (8), $f(\cdot)$ denotes the activation function of the neuron, and the study chooses Relu as the activation function as a way to save the computation time of the network and prevent the gradient from disappearing. In order to simplify the formula, the study expresses the weight values and bias values of all neuron layers in the DNN in the form of vectors, which is denoted as $\hat{W} = (W; BS)$. If the input layer data is $IN = (in_1, in_2, \cdots, in_m)$, its output layer data can be denoted as $OP = f\left(\hat{W}, IN\right)$. The mean-square error (MSE) calculation formula in the regression model is used to determine whether the to-be-detected data of the

DNN is threatening data. The calculation formula of MSE is shown in Equation (9).

$$MSE = \frac{\sum_{s=1}^{no}\left(o\hat{p}_s - op_s\right)^2}{no} \quad (9)$$

In Equation (9), s and no denote the number and total number of training samples, respectively. $o\hat{p}_s$ and op_s denote the true and detected values of the s th training sample, respectively. The updating of parameters is accomplished using gradient descent method as shown in Equation (10) and Equation (11).

$$\Delta\hat{W}(t)_i = \eta\nabla_{\hat{W}}\left\{\frac{\left[O\hat{P}_i - f\left(\hat{W}(t-1), IN_i\right)\right]^2}{no}\right\} \quad (10)$$

In Equation (10), $\hat{W}(t)_i$ denotes the updated model parameters of the mobile terminal i locally. $\hat{W}(t-1)$ denotes the global model parameters before updating. $\Delta\hat{W}(t)_i$ denotes the adjusted value of the update. η denotes the learning step parameter. $O\hat{P}_i$ denotes the real output data of mobile terminal i. IN_i denotes the input data of mobile terminal i. The specific expansion of $\hat{W}(t)_i$ is shown in Equation (11).

$$\hat{W}(t)_i = \hat{W}(t-1) - \Delta\hat{W}(t)_i \quad (11)$$

For $\hat{W}(t)_i$, the mobile terminal i will enter the information of p_k in the Paillier algorithm and compute the ciphertext $c(t)_i$ waiting to be transmitted. The transmission process of $c(t)_i$ is shown in Equation (12).

$$c(t)_i = g^{\hat{W}(t)_i} r^n \left(mod n^2\right) \quad (12)$$

According to Equation (8) to Equation (12), DNN is chosen as the detection model in the study, which is utilized to complete the task of security threat detection for mobile devices. The mobile terminals participating in the DNN model training will train and update the initial parameter set obtained in the local data before encrypting the model parameters for transmission using the p_k public key in Paillier's algorithm. The encrypted transmitted parameters will enter into the edge-end server, and in this part the parameter ciphertexts are then selected and aggregated using the BPM, so as to obtain the global parameter ciphertexts. The global model parameter ciphertext obtained by the mobile terminal will be decrypted again by the private key s_k in the Paillier algorithm.

3 Results

In order to demonstrate the superior benchmark performance and practical applicability of the proposed Paillier-DNN-BPM model, this study introduces AES, HE, and secure hash algorithm (SHA) as comparison algorithms and employs the following criteria for the comparison test: detection precision, MSE, key sensitivity, and data encryption relevance.

3.1 Algorithm benchmark performance testing

Since the final designed Paillier-DNN-BPM model is composed of three parts, Paillier, DNN and BPM, the study starts with an ablation test of the Paillier-DNN-BPM model as a way of proving the performance of each part of the model. The results of the ablation test are shown in Table 3.

In Table 3, the various benchmark performances of Paillier+DNN+BPM, i.e., Paillier-DNN-BPM, have the best performance in the seven sets of ablation tests. The detection precision, recall, and F1 value of Paillier-DNN-BPM are as high as 0.982, 0.989, and 0.988, respectively. The value are much higher than those of the single BPM of 0.831 precision, 0.842 recall, and 0.836 F1 value. The performance of all models decreases after introducing noise and adversarial attacks. However, the Paillier DNN-BPM model still maintains high accuracy, recall, and F1 score. In adversarial attack scenarios, its F1 value is 0.931, indicating the robustness of the model in the face of noise and attacks. It can be concluded that the various components used in the study in the Paillier-DNN-BPM model all have important roles. The sensitivity of the four algorithms HE, AES, SHA, and Paillier-DNN-BPM is further tested and the results are shown in Figure 6. Figure 6(a) and Figure 6(b) show the sensitivity values of HE, AES, SHA, and Paillier-DNN-BPM in the training set and test set, respectively. In Figure 6(a), the fluctuation ranges of the sensitivity values of HE, AES, SHA, and Paillier-DNN-BPM in the training set are -0.19-0.46, -0.21-0.28, -0.13-0.25, and -0.05-0.15, respectively. Compared to HE, AES, and SHA, the sensitivity values of Paillier-DNN-BPM fluctuation range is minimized. Similarly, it can be found that Paillier-DNN-BPM has the smallest fluctuation range of sensitivity values in the test set of Figure 6(b), which is only -0.15~0.09. Thus, it can

be concluded that Paillier-DNN-BPM has the best sensitivity and is able to detect the existence of threat data in time. The MSE values of the four algorithms are shown in Figure 7.

Table 3: Ablation test results of Paillier-DNN-BPM model

Model	Precision	Recall	F1 value	F1 value after introducing Gaussian noise	F1 value after introducing adversarial attacks
Paillier	0.862	0.875	0.871	0.842	0.815
DNN	0.885	0.892	0.890	0.861	0.832
Box diagram method	0.831	0.842	0.836	0.806	0.781
Paillier+Box diagram method	0.905	0.912	0.910	0.876	0.852
DNN+Box diagram method	0.921	0.926	0.922	0.889	0.865
Paillier+DNN	0.943	0.956	0.951	0.917	0.893
Paillier+DNN+Box diagram method (Paillier-DNN-BPM)	0.982	0.989	0.988	0.954	0.931

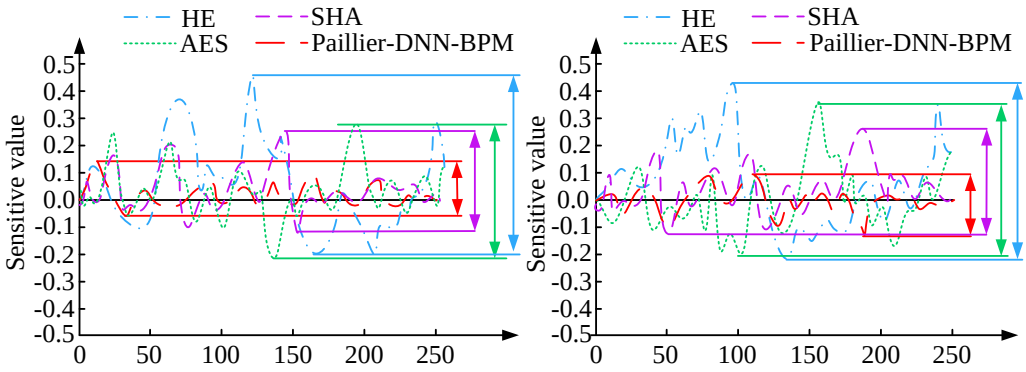

Figure 6: Sensitivity of different algorithms

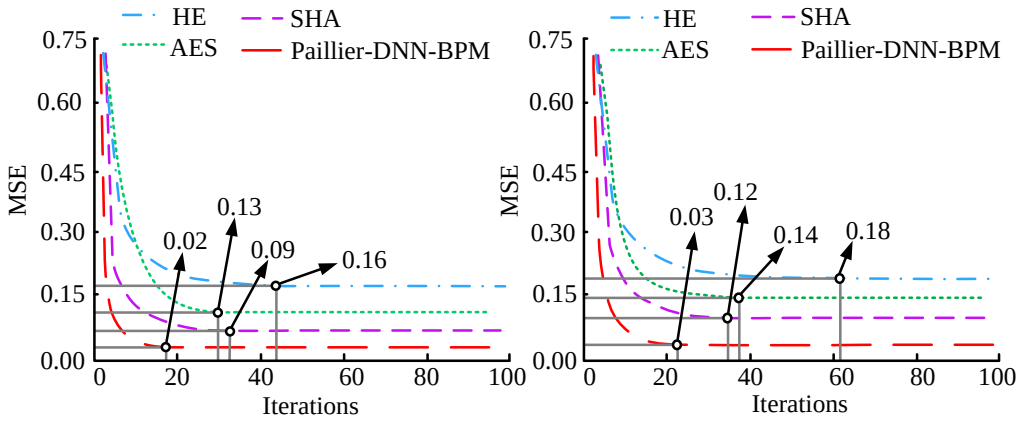

Figure 7: MSE of different algorithms

In Figure 7, the MSE values of the four algorithms in the training and test sets are given. In Figure 7(a), the MSE values of HE, AES, SHA, and Paillier-DNN-BPM are 0.16, 0.13, 0.09, and 0.02, respectively, when they reach the stability in the training set. In Figure 7(b), Paillier-DNN-BPM reaches the stability in the test set

first, and at this time, the MSE value is also the smallest, which is only 0.03. Taken together, the Paillier-DNN-BPM has better error performance during both training and testing.

3.2 Effect of practical application of the model

Four algorithms, HE, AES, SHA, and Paillier-DNN-BPM, are applied to the MCSN to build the corresponding four types of STDMs, respectively. In order to test the effectiveness of the four models in practical applications, the study chooses the intelligent transportation system of a first-tier city as the object. The MCSN is applied to this intelligent transportation system to test the detection precision of the four types of STDM for the six types of threat data in the intelligent transportation MCSN, as shown in Table 4.

In Table 4, when MCSN is applied to the intelligent transportation system, at this time, the detection rates of STDMs built using the four algorithms of HE, AES, SHA, and Paillier-DNN-BPM are all above 80%. Among them, the detection precision of Paillier-DNN-BPM model for all six types of threats is higher, up to 98.84%. While HE has generally lower detection precision for the six categories of threats, as low as 81.25%. The average detection time of the four models for the six categories of threats is tested and shown in Figure 8. In Figure 8, the average detection time of Paillier-DNN-BPM model is lower than that of HE, AES, and SHA models for all six classes of threats. Taking class, A threat as an example, the average detection time of HE, AES, SHA, and Paillier-DNN-BPM for this class of threat is 0.62, 0.56, 0.19, and 0.15, respectively, which shows that the detection time of Paillier-DNN-BPM is shorter and its detection efficiency is higher in practical applications. Comparing the encryption relevance and decryption relevance of the four types of models for data, as shown in Table 5.

Table 4: Detection precision of different models for six types of threats

Threat type	HE	AES	SHA	Paillier-DNN-BPM
A	81.25%	89.24%	92.36%	96.52%
B	84.06%	85.74%	90.58%	98.10%
C	85.97%	87.89%	89.17%	98.84%
D	82.23%	86.30%	91.25%	96.78%
E	85.79%	88.15%	90.21%	96.29%
F	83.41%	87.96%	89.73%	97.06%

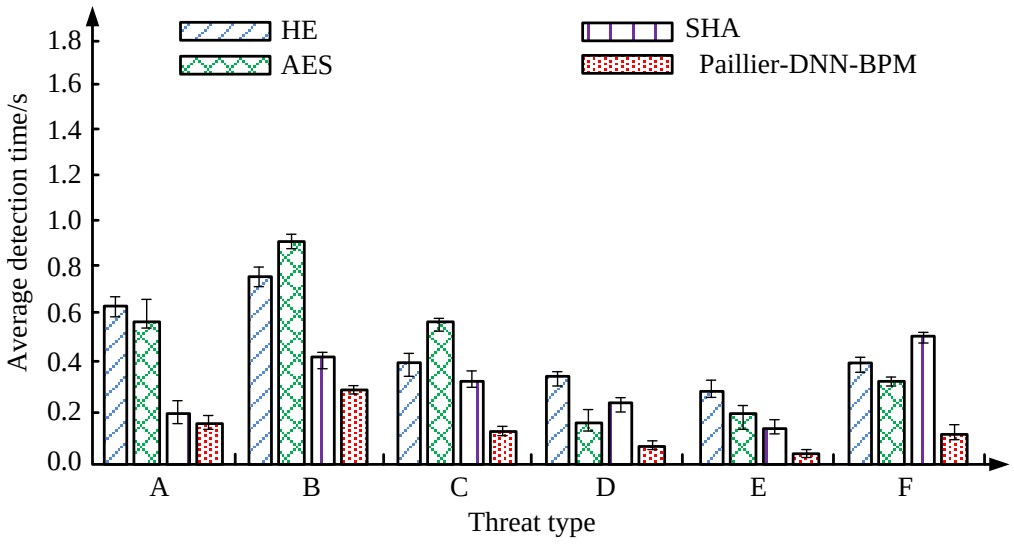

Figure 8: Average detection time of different models for six types of threats

Table 5: Correlation of data encryption and decryption of different models

Threat type	Cryptographic correlation				Decryption correlation			
	HE	AES	SHA	Paillier-DNN-BPM	HE	AES	SHA	Paillier-DNN-BPM
A	0.853	0.876	0.921	0.975	0.862	0.855	0.914	0.971
B	0.846	0.872	0.918	0.973	0.858	0.886	0.902	0.975
C	0.860	0.869	0.911	0.981	0.869	0.873	0.919	0.979
D	0.859	0.852	0.925	0.969	0.832	0.859	0.913	0.965
E	0.855	0.861	0.908	0.986	0.821	0.864	0.901	0.982
F	0.842	0.878	0.913	0.992	0.835	0.859	0.921	0.987

Combining all the correlation values in Table 5, it can be observed that the maximum correlation differences of HE, AES, SHA, and Paillier-DNN-BPM are 0.026, 0.021, 0.016, and 0.005, respectively, which shows that Paillier-DNN-BPM has the smallest loss during encryption and decryption, and the final detected threat data is more accurate. To investigate the universality of the proposed model, the study applies it to healthcare and smart city environments. The performance indicators obtained are shown in Figure 9.

In Figure 9, the proposed model still maintains high accuracy, recall rate, and F1 value in both healthcare and smart city environments, further verifying the generality and practical application potential of the model.

4 Discussion

In order to improve the PP capability of MCSN and the accuracy of security threat detection, a Paillier DNN BPM model was designed, and its performance was verified in the study. The PP joint learning method based on multi-bond HE proposed by Ma J et al. in reference [19] mainly focused on the application of multi-bond HE in joint learning. Although the Ma J method performed well in protecting data privacy, it lacked real-time threat detection and adaptability in complex network environments. To comprehensively evaluate the superiority of the proposed model, its performance in detection accuracy, computational efficiency, and scalability was compared with other common methods, including HE, AES, and SHA. The experimental results showed that the proposed model achieved detection accuracy, recall rate, and F1 score of 0.982, 0.989, and 0.988, respectively, which were significantly better than single encryption algorithms or traditional detection models. Especially in the application of intelligent transportation systems, the detection accuracy of the proposed model was as high as 98.84%, while the detection accuracy of HE, AES, and SHA were 94.32%, 91.76%, and 88.45%, respectively. In terms of computational efficiency, the proposed model demonstrated higher efficiency in encryption and decryption operations by combining PHE and DNN technologies. Compared with the HE

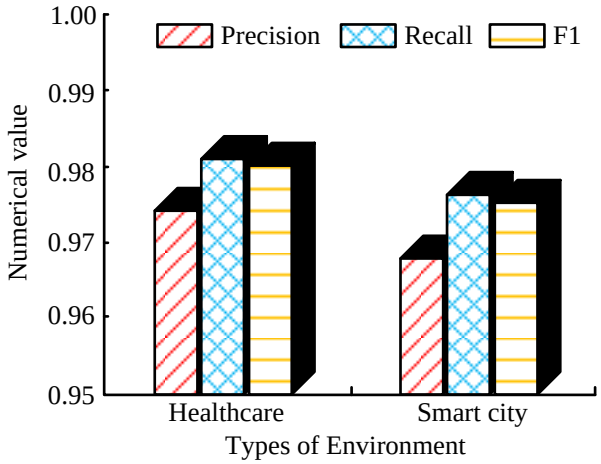

Figure 9: Performance of Paillier DNN-BPM model in different environments

model, the proposed model could significantly reduce computation time in large-scale data processing and maintain lower latency at different network scales. In terms of scalability, the proposed model adopted a distributed architecture of TEC three-level collaboration, which enabled the model to better adapt to complex network environments when processing large-scale data, demonstrating higher practicality and flexibility.

Although FHE can theoretically achieve homomorphic computation of any function, its computational complexity is extremely high, making it difficult to meet real-time requirements in practical applications. In contrast, Paillier encryption is a partially HE technique that supports only additive homomorphisms, which makes its computational complexity much lower than FHE, thus showing higher computational efficiency in practical applications. Federated learning models have privacy advantages, but are susceptible to network latency and synchronization issues due to the need for model synchronization and updates among multiple participants. A comparison of the HE-based PP deep learning method proposed by Falcetta A et al. in the literature [20] with other methods indicates that, although the method had high accuracy in PP of deep learning models, its complex structure may result in lower computational efficiency in large-scale MCSNs. The proposed model reduced the dependence on network synchronization on the basis of data privacy protection by adopting the TEC architecture, while improving the flexibility and adaptability of the overall system through distributed computing. The proposed model combined DNN for threat detection, leveraging the advantages of DNN in handling complex pattern recognition tasks to improve the real-time and accuracy of threat detection. In contrast, FHE models were difficult to achieve the same effect in real-time applications due to their high complexity, while federated learning models were limited by data dispersion and network synchronization issues, resulting in poor real-time performance. The key to the superiority of the proposed model over other alternative methods in multiple performance metrics lied in its architecture selection. The TEC three-level collaborative structure provided high flexibility and scalability for the model, enabling it to maintain efficient performance at different network sizes and complexities. Through the TEC architecture, the proposed model could improve system reliability while reducing network latency. The Paillier encryption algorithm's principal advantage lies in its low computational complexity, which enables the system to maintain efficient encryption and decryption processes at various stages of data transmission and processing. This results in efficient PP.

In summary, the study proposes an efficient and accurate lightweight STDM by combining PHE, DNN and BPM: Paillier-DNN-BPM. This not only provides new ideas and technical means for future PP of MCSN, but also positively impacts on improving the overall security of the network.

5 Conclusion

The results demonstrated that the Paillier-DNN-BPM model had better ablation test results and showed the best performance in detection precision, recall and F1 value, which was significantly higher than a single encryption algorithm or a traditional detection model. In addition, the Paillier-DNN-BPM model also had better performance in sensitivity and error detection, with smaller sensitivity fluctuation ranges and error values. Finally, the performance of the model was tested in a real operating environment, and compared with the HE, AES, and SHA models, Paillier-DNN-BPM demonstrated high detection accuracy and low system response time in intelligent transportation system application scenarios. In summary, the Paillier-DNN-BPM model not only performs well in ablation tests, but also achieves excellent detection results in practical applications. Further optimization of the computational and communication overhead of the model may be achieved through the implementation of follow-up studies. Additionally, the threat detection effect of different combinations of Paillier-DNN-BPM models in other complex network environments can be evaluated, thereby enhancing the generalizability and practicality of the model.

The equation symbols and their explanations are shown in Table 6.

Table 6: Equation symbols and their explanations

Equation Symbol	Explanation
n	Modulus
p, q, u	Large prime numbers
λ	Parameter
$lcm(\cdot)$	Least common multiple
$L(\cdot)$	Function used to calculate the intermediate value in the decryption process
g, r	Random number

n^2	Modulus square
p_k, s_k	Public key, private key
x, y	Plaintext
x', y'	Encrypted ciphertext
$E(x)$, $E(y)$	Encryption function
$D(x')$, $D(y')$	Decryption function
a	Variable of that function
$E(x+y)$	Addition operation directly on the plaintext
z	Number of times a certain power square is calculated and is a constant
l	Neuron of layer l in the DNN
$f(\cdot)$	Activation function of the neuron
$OP^{(l-1)}$	Output of the previous laye
$W^{(l)}$	Weights of the neurons of layer l
$BS^{(l)}$	Biases of the neurons of layer l
i	Mobile terminal
$\hat{W}(t)_i$	Updated model parameters of the mobile terminal i locally.
$\Delta\hat{W}(t)_i$	Adjusted value of the update
η	Learning step parameter.
$\hat{OP}_i$	Real output data of mobile terminal i
IN_i	Input data of mobile terminal i

References

[1] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. Fitzek, and N. Aaraj, "Survey on fully homomorphic encryption, theory, and applications," Proceedings of the IEEE, vol. 110, no. 10, pp. 1572-1609, 2022. https://doi.org/10.1109/JPROC.2022.3205665

[2] S. Saminu, G. Xu, S. Zhang, I. A. E. Kader, H. A. Aliyu, A. H. Jabire, Y. K. Ahmed, and M. J. Adamu, "Applications of artificial intelligence in automatic detection of epileptic seizures using EEG Signals: A review," Artificial Intelligence and Applications, vol. 1, no. 1, pp. 11-25, 2023. https://doi.org/10.47852/bonviewAIA2202297

[3] M. Dai, Z. Su, Q. Xu, Y. Wang, and N. Lu, "A trust-driven contract incentive scheme for mobile crowd-sensing networks," IEEE Transactions on Vehicular Technology, vol. 71, no. 2, pp. 1794-1806, 2021. https://doi.org/10.1109/TVT.2021.3117696

[4] D. Asprone, S. Di Martino, P. Festa, and L. L. L. Starace, "Vehicular crowd-sensing: a parametric routing algorithm to increase spatio-temporal road network coverage," International Journal of Geographical Information Science, vol. 35, no. 9, pp. 1876-1904, 2021. https://doi.org/10.1080/13658816.2021.1893737

[5] X. Zheng, Q. Yuan, B. Wang, and L. Zhang, "A homomorphic encryption-based location privacy preservation scheme for crowdsensing tasks allocation," Wireless Personal Communications, vol. 126, no. 1, pp. 719-740, 2022. https://doi.org/10.1007/s11277-022-09767-y

[6] R. Ganjavi, and A. R. Sharafat, "Edge-assisted public key homomorphic encryption for preserving privacy in mobile crowdsensing," IEEE Transactions on Services Computing, vol. 16, no. 2, pp. 1107-1117, 2022. https://doi.org/10.1109/TSC.2022.3172136

[7] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT," IEEE Transactions on Industrial Informatics, vol. 18, no. 6, pp. 4049-4058, 2021. https://doi.org/10.1109/TII.2021.3085960

[8] S. Jumonji, K. Sakai, M. T. Sun, and W. S. Ku, "Privacy-preserving collaborative filtering using fully homomorphic encryption," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 3, pp. 2961-2974, 2021. https://doi.org/10.1109/TKDE.2021.3115776

[9] Y. Ren, T. Wang, S. Zhang, and J. Zhang, "An intelligent big data collection technology based on micro mobile data centers for crowdsensing vehicular sensor network," Personal and Ubiquitous Computing, vol. 27, no. 3, pp. 563-579, 2023. https://doi.org/10.1007/s00779-020-01440-0

[10] T. N. Nguyen, and S. Zeadally, "Mobile crowd-sensing applications: Data redundancies, challenges, and solutions," ACM Transactions on Internet Technology (TOIT), vol. 22, no. 2, pp. 1-15, 2021. https://doi.org/10.1145/3431502

[11] X. Li, G. Feng, Y. Liu, S. Qin, and Z. Zhang, "Joint sensing, communication, and computation in mobile crowdsensing enabled edge networks," IEEE Transactions on Wireless Communications, vol. 22, no. 4, pp. 2818-2832, 2022. https://doi.org/10.1109/TWC.2022.3214535

[12] C. Xu, and W. Song, "An adaptive data uploading scheme for mobile crowdsensing via deep reinforcement learning with graph neural network," IEEE Internet of Things Journal, vol. 9, no. 18, pp. 18064-18078, 2022. https://doi.org/10.1109/JIOT.2022.3163456

[13] Y. Ren, H. Jiang, X. Feng, Y. Zhao, R. Liu, and H. Yu, "ACP-based modeling of the parallel vehicular crowd sensing system: Framework, components and an application example," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 2, pp. 1536-1548, 2022. https://doi.org/10.1109/TIV.2022.3221927

[14] K. Munjal, and R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," Complex & Intelligent Systems, vol. 9, no. 4, pp. 3759-3786, 2023. https://doi.org/10.1007/s40747-022-00756-z

[15] X. Yang, S. Zheng, T. Zhou, Y. Liu, and X. Che, "Optimized relinearization algorithm of the multikey homomorphic encryption scheme," Tsinghua Science and Technology, vol. 27, no. 3, pp. 642-652, 2021. https://doi.org/10.26599/TST.2021.9010047

[16] S. Li, S. Zhao, G. Min, L. Qi, and G. Liu, "Lightweight privacy-preserving scheme using homomorphic encryption in industrial internet of things," IEEE Internet of Things Journal, vol. 9, no. 16, pp. 14542-14550, 2021. https://doi.org/10.1109/JIOT.2021.3066427

[17] Z. Cheng, F. Ye, X. Cao, and M. Y. Chow, "A homomorphic encryption-based private collaborative distributed energy management system," IEEE Transactions on Smart Grid, vol. 12, no. 6, pp. 5233-5243, 2021. https://doi.org/10.1109/TSG.2021.3091624

[18] M. D. Boomija, and S. V. K. Raja, "Securing medical data by role-based user policy with partially homomorphic encryption in AWS cloud," Soft Computing, vol. 27, no. 1, pp. 559-568, 2023. https://doi.org/10.1007/s00500-022-06950-y

[19] J. Ma, S. A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," International Journal of Intelligent Systems, vol. 37, no. 9, pp. 5880-5901, 2022. https://doi.org/10.1002/int.22818

[20] A. Falcetta, and M. Roveri, "Privacy-preserving deep learning with homomorphic encryption: An introduction," IEEE Computational Intelligence Magazine, vol. 17, no. 3, pp. 14-25, 2022. https://doi.org/10.1109/MCI.2022.3180883