Secure and Efficient Federated Learning by Combining Homomorphic Encryption and Gradient Pruning in Speech Emotion Recognition

Samaneh Mohammadi[1,2], Sima Sinaei[1], Ali Balador[2], and Francesco Flammini[2]

[1] RISE Research Institutes of Sweden, Västerås, Sweden.
{samaneh.mohammadi, sima.sinaei}@ri.se
[2] Mälardalen University, Västerås, Sweden.
{ali.balador, francesco.flammini}@mdu.se

Abstract. Speech emotion recognition (SER) enables intelligent systems to detect emotions in spoken language, adapting to human needs. However, privacy concerns arise when analyzing speech data, as it reveals sensitive information like identity, emotions, age and gender. To address this, Federated Learning (FL) has been developed, allowing models to be trained locally and sharing model parameters with servers. However, FL introduces new privacy concerns when transmitting local model parameters between clients and servers, as third parties could exploit this information to reconstruct speech data or features, potentially disclosing sensitive information. In this paper, we introduce a novel approach called Secure and Efficient Federated Learning (SEFL) for SER applications. Our proposed method combines Paillier homomorphic encryption (PHE) with a novel gradient pruning technique. This approach enhances privacy and maintains confidentiality in FL setups for SER applications while minimizing communication and computation overhead and ensuring satisfactory model accuracy. As far as we know, this is the first paper that implements Paillier homomorphic encryption in FL setup for SER applications. During communication and aggregation, the confidentiality of local parameters is maintained. The server can aggregate ciphertexts without decrypting them, preventing access to confidential information. In our experiments, we utilized a publicly available SER dataset to assess the performance of the SEFL method. The results indicate significant efficiency gains when using a key size of 1024, including a reduction in computation time of up to 25% and a decrease in communication traffic of up to 70%. Notably, these improvements are achieved with minimal impact on accuracy, ensuring that the requirements of SER applications are met effectively.

Keywords: Federated Learning · Privacy-preservation · Homomorphic Encryption · Speech Emotion Recognition.

1 Introduction

Speech Emotion Recognition (SER) refers to the detection and classification of human emotions as they are expressed in spoken language [18]. This ability makes SER highly valuable in various fields, including mental health diagnosis and therapy, where it can aid mental health practitioners in their decision-making process and treatment monitoring by identifying a person's actual emotions [27]. Educational settings can also utilize SER to track students' emotional states and engagement levels in e-learning, enabling teachers to implement more effective teaching strategies [32]. Furthermore, SER has potential applications in the entertainment industry, such as the development of TV recommendation systems that accurately capture users' emotions and provide personalized experiences, resulting in higher levels of user satisfaction [20].

Analyzing human speech data can reveal sensitive information, including the speaker's biometric identity, personality traits, geographic origin, emotional state, age, gender, and overall health condition [21]. Ethical and privacy concerns arise when using speech data. Regulations such as the General Data Protection Regulation (GDPR) [36] and California Consumer Privacy Act (CCPA) [13] have been introduced to protect personal data. It is crucial to consider privacy concerns when developing and implementing SER application in various domains.

Federated Learning (FL) offers a promising solution to maintain data privacy while enabling machine learning (ML) models to be trained on decentralized devices [28]. FL trains ML models on local client devices without transferring raw data to a central server, which preserves data privacy and ensures compliance with regulations such as GDPR and CCPA. For SER applications, the initial processing of speech data and training perform on clients' device, and only local model parameters are sent to the central server for model aggregation [22]. This approach can reduce the risk of privacy breaches while still achieving accurate outcomes for SER applications.

However, FL faces new privacy concerns when it comes to transmitting local model parameters between clients and servers. This is a concern because the transmission data could potentially be exploited by third parties to perform attacks that reconstruct raw speech data or features and disclose sensitive information [15]. To address this issue, additional privacy mechanisms have been proposed together with FL to safeguard such applications.

One of the promising mechanisms commonly used in FL is Differential Privacy (DP) [19, 37, 40]. DP is employed to safeguard individual data points in specific scenarios. However, when applied to SER applications, DP does not offer acceptable accuracy due to the adverse effects of adding noise to voice data, which can distort the audio signal [30]. Furthermore, adding noise to SER model parameters can affect the model's utility by distorting or misaligning the parameters, leading to errors in the model's output [11]. This accuracy compromise can pose significant challenges for industrial SER applications that require precise results [3].

An alternative approach to ensuring privacy and preserving accuracy in SER applications is through the use of diverse homomorphic encryption methods, one

of which is Paillier homomorphic encryption (PHE) [10]. Incorporating these methods safeguards users' privacy and confidentiality in FL while maintaining the accuracy of SER models. PHE allows local model parameters to remain encrypted and secure during communication and computation with no adverse effect on the accuracy of the SER model. Although there are limited research publications on the potential benefits of PHE for SER applications in FL, a comprehensive analysis of implementing this approach can offer valuable insights for both academia and industry.

Additionally, using PHE as a privacy-preserving approach for SER in the context of FL presents further challenges. The use of PHE in FL may result in increased communication traffic and computation time [38], which can be especially problematic in settings with limited bandwidth or resource, such as edge devices. Therefore, careful consideration of these challenges is necessary when implementing PHE in FL systems to ensure the privacy and efficiency of SER applications.

In this paper, we propose a new method for SER called Secure and Efficient Federated Learning (SEFL), which combines Paillier homomorphic encryption with a novel gradient pruning technique. This approach enhances privacy in FL setups for SER applications while reducing communication traffic and computation time with almost maintaining acceptable model accuracy. The gradient pruning technique is applied to the gradient updates of each client in every training round. It is based on the magnitude of the gradients and aims to remove or prune gradients with low magnitudes. These low-magnitude gradients contribute less to weight updates and have limited impact on the overall performance.

The SEFL method effectively reduces the size of encrypted local model parameters transmitted between the client and server, leading to decreased communication traffic. Additionally, gradient pruning reduces the number of parameters and floating point operations (FLOPs), shortening the encryption and decryption time and thus addressing the computation time associated with encryption methods.

The novel contributions of this paper can be summarized as follows:

- Develop a novel SEFL algorithm for SER applications that ensures privacy and confidentiality while enhancing efficiency in terms of reduced communication traffic and computation time, as well as maintaining acceptable accuracy.
- Conduct a proof of concept implementation of Paillier homomorphic encryption in FL for SER applications to ensure the confidentiality of local model parameters.
- Evaluate SEFL on a public SER dataset to demonstrate its considerable gains in efficiency, such as a reduction of computation time by 10-25% and communication traffic by 50-70%, depending on pruning percentage, while having a very limited impact on accuracy, in order to meet the requirements of SER applications.

The remainder of this paper is structured as follows. In Section 2, we provide an overview of the background and related works on SER using FL, privacy-

preserving techniques in FL, and communication and computation-efficient FL. Section 3 presents the application of SEFL method for SER, including the non-functional requirements of SER, the threat model, and the proposed method SEFL with its algorithm. In Section 4, we present the experimental results obtained using SEFL in the SER reference application. Finally, Section 5 concludes the paper and provides insights for future developments.

2 Background and Related works

This section will cover related work in SER using FL, provide a brief overview of various homomorphic encryption techniques used as privacy-preserving mechanisms in FL, and review related works on communication and computation-efficient FL.

2.1 Speech Emotion Recognition using Federated Learning

Speech emotion recognition (SER) identifies and understands human emotions through speech. The SER application analyzes audio signals from human speech and applies ML algorithms to identify patterns and classify emotions conveyed by the speech. SER models require large amounts of data, including sensitive personal information like speech signals and emotions [2]. However, the centralized storage of this data entails privacy concerns. In order to mitigate these risks, FL offers a promising solution for collaborating on decentralized devices without transferring raw data [22].

In [34], an FL-based approach is presented for building a private decentralized SER model using data-efficient federated self-training with minimal on-device labelled samples. However, this method solely relies on the FL framework as a privacy-preserving technique and does not consider threat models from clients or servers in FL, nor does it consider other privacy-preserving techniques. Similarly, [6] proposes an FL-based federated adversarial learning framework to protect both data and deep neural networks in SER, using the FL framework for data privacy and adversarial training during the training stage for model robustness. However, like the previous method, this approach solely relies on the FL framework for privacy preservation and does not consider other privacy-preserving techniques in FL.

2.2 Privacy-Preserving Federated Learning

Homomorphic encryption (HE) is a technique used in FL to protect user privacy when intermediate parameters are exchanged between parties also allows for secure aggregation [41]. Using HE in FL, data can be encrypted before it is sent to the central server for model training. This means that the data remains private throughout the training process, as it can only be decrypted by the owner of the data. The central server can perform computations on the encrypted data using homomorphic operations, such as addition and multiplication, without

ever decrypting it. The encrypted results can then be sent back to the devices for decryption and aggregation, allowing the model to be trained without ever exposing sensitive data [25].

Specifically, an additively homomorphic scheme allows some operation to be performed directly on the ciphertexts $E(m_1)$ and $E(m_2)$, so that the result of the operation is a new ciphertext whose decryption yields the sum of plaintexts m_1 and m_2. Most prevalent among HE variants are Paillier [29], FV [9], and CKKS [7]. Paillier allows additions to encrypted data, whereas FV and CKKS allow additions and multiplications to encrypted data. It is possible to encrypt integers using the Paillier and FV schemes, but only approximate results can be obtained with the CKKS scheme. However, most HE variants add additional computational and communication overhead, making it more challenging to scale FL to large numbers of devices.

2.3 Communication and Computation-Efficient Federated Learning

During FL training, model parameters are iteratively transmitted between devices and a central server. However, this approach can lead to high communication overhead and slow down the learning process [33]. Additionally, Deep Neural Network (DNN) models often contain millions of parameters [31], and the training process is becoming more computationally and memory-intensive due to the increasing complexity of the networks and the training data [23]. One way to reduce overhead is to use compression techniques such as gradient pruning, which has been shown to be effective in reducing the size of models without sacrificing performance [16]. This technique involves setting a fixed threshold for the gradient values and removing parameters below this threshold.

A recent study in FL which aims to reduce communication and computation costs is the edge Stochastic Gradient Descent (eSGD) algorithm, which selects significant gradients for server updates. However, this can lead to accuracy loss and performance fluctuations [33]. An alternative solution proposed in [16] involves joint training and pruning of a DNN model in a federated manner, reducing its size and improving communication and computation time. Yet, the weight pruning method in [16] has been criticized for its inefficiency in computation and storage overhead [26]. Additionally, the proposed approach neglects communication overheads during the update process, which can be problematic with many clients or additional privacy-preserving mechanisms.

Implementing privacy-preserving mechanisms like Homomorphic Encryption (HE) on edge devices with limited computational capabilities and communication bandwidth in FL systems introduces significant overhead and impractically long training times [17]. Optimization strategies are necessary to address these challenges. Proposed solutions include batching multiple plaintexts into a single one to reduce computation overhead [38], but this approach still results in high communication overhead. Another approach is sparsification, where the client sends only a sparse subset of local states to the server, significantly reducing communication overhead [1, 14]. However, if gradient components are encrypted as a single ciphertext, the benefits of sparsification in HE are limited and may

harm model accuracy. Despite recent advancements, achieving a balance between communication, computation, and privacy in FL systems with HE remains a challenging task. The field is continuously evolving, and there is ample room for further improvements in efficiency.

3 Application of Secure and Efficient Federated Learning for Speech Emotion Recognition

In this section, we will provide an overview of the non-functional requirements for SER applications, describe the threat model for our system, and provide a detailed explanation of the proposed SEFL (Secure and Efficient Federated Learning) method for SER applications.

3.1 Non-functional Requirements of Speech Emotion Recognition Application

Non-functional requirements refer to the characteristics or qualities of a system that are related to its performance rather than its specific functionality. In the context of SER applications, important non-functional requirements include accuracy, privacy, efficiency in terms of communication traffic, and scalability. Satisfying these requirements is critical to ensure user needs and expectations while complying with legal requirements. A detailed explanation of the non-functional requirements is provided in this part, and the evaluation section illustrates how we meet these requirements.

1. *Privacy:*
   (a) Personal speech data must be kept on local devices only [36].
   (b) The central server must not be able to access local model parameters to infer sensitive information.
   (c) Communication between clients and servers should be protected from unauthorized access in order to keep SER parameters confidential.
2. *Efficiency:*
   (a) In order to reduce hardware costs and consider the typically resource-constrained edge devices, SER computation overhead must be minimized.
   (b) Communication overhead between SER clients and servers must be minimized in order to optimize network resource consumption when using limited bandwidth connections.
3. *Accuracy:*
   (a) The level of accuracy of SER applications must be kept high enough to reliably identify the correct emotions from speech samples. We can consider a baseline accuracy of a minimum 70% in detecting the four basic emotions - neutral, sad, happy, and angry [35].
4. *Scalability:*

(a) The scalability of SER must be such to allow the management of a large number of clients and related amounts of speech data with a linear rather than exponential increase in execution times.

It is important to highlight that those requirements can be highly interdependent. For example, privacy-preserving approaches can impact efficiency due e.g. the computation overhead associated with FL, encryption method, etc. In addition, it is necessary to consider the possibility of a 0-5% drop in accuracy when implementing an SER in the FL setup [34]. This paper defines communication traffic as the number of bits transferred between clients and servers. In SER centralized training, the amount of speech data each client sends to the central server depends on factors like the length of an audio clip, the sampling rate, and the pre-processing steps. For instance, the CREMA-D dataset [5] used 7,442 audio clips collected from 91 clients, each sending approximately 8-10 MB to the central server in each centralized training round.

3.2 Threat Model

In this paper, we assume that the server complies to the honest-but-curious (HBC) paradigm, which refers to a server that is not malicious and follows the FL protocol, but it is still curious about the data or models of the clients [24]. This raises the following potential threats.

- The HBC server has the potential to infer sensitive information, including the speaker's identity, through the reconstruction of speech data by model parameters. The server can potentially identify individuals by analyzing the distinctive characteristics of the speaker's voice, such as pitch, tone, and accent.
- HBC servers can analyze reconstructed speech data by model parameters to determine sensitive information about the speaker's emotional state. The speaker's emotional state or personality can be revealed through emotions such as anger, sadness, and anxiety.

3.3 Proposed Method: SEFL

To address the privacy threat posed by a server that is honest but curious, as well as to meet the non-functional requirements of the SER application, we propose a new approach called Secure and Efficient Federated Learning (SEFL). SEFL combines the use of Paillier homomorphic encryption with a novel gradient pruning method. The SEFL method ensures that the speech data remains on the end devices during training (Requirement 1.a). By deploying Paillier homomorphic encryption within this method, we guarantee that the HBC server only has access to ciphertext data and cannot infer sensitive information from the model parameters (Requirement 1.b). Additionally, the encrypted model parameters shared by clients ensure that unauthorized parties cannot access the SER model

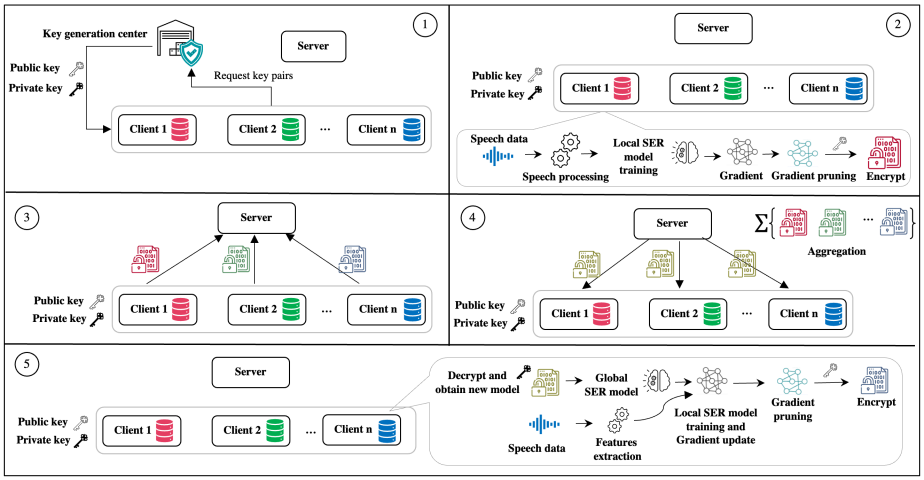

Fig. 1: An overall overview of secure and efficient federated learning for speech emotion recognition.

without compromising the cryptosystem (Requirement 1.c). Furthermore, SEFL incorporates gradient pruning based on magnitude on the client side aim to remove or prune gradients with low magnitudes, as they contribute less to weight updates and have limited effect on overall performance. This approach reduces encryption computation and communication overhead, resulting in improved efficiency and scalability (Requirements 2 and 4), while maintaining comparable accuracy to the initial model (Requirement 3).

Fig. 1 presents the overall overview of SEFL for SER application. In Step 1, we initialize the initial model, and clients who wish to participate in each training round send a request to the key generation center for a key pair. The key generation center collects these requests, generates public-private key pairs, and returns them to the clients. Moving to Step 2, clients perform speech processing to extract relevant features and train their SER models using a multilayer perceptron network locally on their own devices. Gradients are calculated using backpropagation, which propagates them from the loss function backwards to help adjust the network parameters based on the gradient, reducing the error between the output value and the desired one. Each client then applies gradient pruning techniques which is based on magnitude aim to remove or prune gradients with low magnitudes, as they contribute less to weight updates and have limited effect on overall performance. This pruning process helps reduce the overall computation and communication overhead during training without significantly impacting the model's accuracy. Additionally, in this step, the Paillier homomorphic encryption scheme is applied to encrypt the newly pruned gradients of each client.

Advancing to Step 3, each client transmits its encrypted gradient to the server. In Step 4, the server leverages homomorphic operations to aggregate

all encrypted client gradients and generate a new encrypted gradient, which it distributes to all clients. Step 5 involves clients decrypting the received new encrypted gradient from the server and updating their local SER model parameters accordingly. These steps continue to iterate until the desired model is achieved or the termination condition is met. For a more comprehensive understanding of the SEFL method, please refer to Algorithm 1, which provides an outline of the steps and rules involved. Additionally, Table 1 displays the parameters and descriptions used in the SEFL for SER algorithm.

Algorithm 1: SEFL for SER

Input: Number of iterations: T, Total Number of clients: N, Speech features of client: x, Number of selected clients: K, Local minibatch size: B, Initial global model: w_g^0, Pruning percentage: pp, Learning rate: η

Output: Secure and Efficient Federated Learning

Server broadcasts w_0^g
for $t \leq T$ **do**
  Key generation center:
  while *listening request from clients* **do**
  if *receive requests from clients* **then**
    Generate key pairs: Public key (pk), Private key (sk)
    Return key pairs {(pk),(sk)}
  Clients-side:
  Request key pairs from **key generation center**
  Initialize the model parameters w_i^t
  for $i \in$ *1,2, ..., K* **do**
    Forward propagation: $label_i = fp(x_i, w_i^t)$
    Compute loss: $c = \text{loss}(f^*(x_i), label_i)$
    if $c < e$ **then**
      Break
    else
      Back propagation: $grad_i = bp(x_i, w_i^t, c)$
      Gradient pruning: $\tilde{grad}_i = (grad_i, pp)$
      Encryption: $E_i = Enc(\tilde{grad}_i, pk)$
      Send E_i to the server
      Receive new aggregated encrypted model from server E_g^t
      Decryption: $grad_i^{t+1} = Dec(E_g^t, sk)$
      Update: $w_{i+1}^t = w_i^t - \eta \cdot grad_i^{t+1}$
  Server-side:
  Aggregation of encrypted local model
  $E_g^t = (E_i^t \bigoplus E_{i+1}^t \bigoplus ... \bigoplus E_K^t)$
  Broadcast updated model parameters E_g^t

Table 1: The parameters and descriptions in the SEFL for SER algorithm.

Parameter	**Meaning**
x	Extracted speech features of clients dataset
w	The parameters of the model
fp	Feed forward process
$Label$	The output label of SER in each itertaion
f^*	Activation function
$loss$	Loss function
c	Loss calculated by loss function
e	Minimum error
bp	Back propagation process
$grad$	Gradient calculated by bp process
η	learning rate
pp	Pruning percentage

Paillier Homomorphic Encryption Within SEFL, the Paillier homomorphic encryption scheme developed as a promising solution to ensure the confidentiality and privacy of participants' speech data in the context of FL, specifically for the SER application. The Paillier cryptosystem, being a partially homomorphic encryption scheme, allows the server to process and aggregate model parameters with the homomorphic property on the server without requiring decryption.

One key advantage of the Paillier homomorphic cryptosystem is its resistance against attacks from a honest-but-curious server. It has been designed to protect against possible breaches of confidentiality by ensuring that ciphertexts do not reveal any information about the plaintexts. This property is proven through its resilience against the chosen plaintext attack (CPA) based on the decisional composite residue problem. Consequently, Paillier emerges as the most efficient partially homomorphic encryption scheme available for FL settings [39].

Basically, Paillier encryption consists of three parts: key generation center, encryption, and decryption. We will discuss it in more detail in the following section.

Key generation center: After receiving clients' requests for a key pair, the Key Generation Center generates the corresponding key pairs and returns them to the clients. Here we explain how to generate keys in more detail by referring to lines [3-7] of Algorithm 1. Select two primes p and q that are sufficiently large and equal in length and satisfy $gcd(p \times q, (p-1) \times (q-1)) = 1$. Then, calculate n, λ and lcm represents the least common multiple as:

$$n = p \cdot q \tag{1}$$

$$\lambda = lcm(p-1, q-1) \tag{2}$$

An integer g is a generator and satisfies $g \in Z_{n^2}^*$ so that n can divide the order of g. Then, define $L(x)$ to calculate μ as:

$$L(x) = \frac{(x-1)}{n} \tag{3}$$

$$\mu = (L(g^{\lambda} \bmod n^2))^{-1}) \bmod n \tag{4}$$

Thus, the public and private key pair in this paper can be shown as $(pk, sk) = \{(n, g), (\lambda, \mu)\}$.

Encryption: The encryption process (line 19 of Algorithm 1) with the public key (pk) can be described as follows, assuming the plaintext is gradient of cleint in each iteration $grad$, the ciphertext is E, and for some random $r \in \{0, \ldots, n-1\}$:

$$E = g^{grad} \cdot r^n \bmod n^2 \tag{5}$$

Decryption: Thus, using a private key (sk), the ciphertext E and plaintext $grad$ can be decrypted as follows (line 22 of Algorithm 1):

$$grad = L(E^{\lambda} \bmod n^2) \cdot \mu \bmod n \tag{6}$$

Gradient Pruning To enhance the efficiency of SEFL methodology for SER deployment on edge devices with limited resources, we present a novel approach that combines Paillier homomorphic encryption with gradient pruning techniques. Gradient pruning techniques, based on magnitude, aim to remove or prune gradients with low magnitudes. The underlying principle behind this technique is that gradients with low magnitudes contribute less to weight updates. By selectively pruning these low-magnitude gradients, the computational and memory requirements associated with computing and storing gradients can be significantly reduced. This enables the SEFL method to reduce the size of encrypted local model parameters transmitted between the client and server, effectively minimizing communication traffic. Moreover, gradient pruning reduces the number of parameters and floating-point operations (FLOPs), leading to faster encryption and decryption times and mitigating computation time related to encryption methods.

The Algorithm 2 showcases gradient pruning techniques based on magnitude. This technique aims to remove or prune gradients with low magnitudes, which have minimal impact on the overall performance of the SER model. The algorithm incorporates a flexible pruning threshold for each layer of the neural network, allowing it to adapt to the specific requirements of each client during every training round. This adaptive approach enhances the effectiveness of the pruning process. By customizing the pruning threshold to match the unique characteristics of each layer, we ensure that only weights with minimal influence on the continuity of the loss function are pruned. This selective pruning strategy preserves the accuracy of the model while effectively reducing computational and memory overhead.

To determine the pruning threshold for each layer, we consider the number of parameters and the desired pruning percentage specific to that layer, as shown

Algorithm 2: Gradient pruning

Input: Client gradient: $grad_i$, Pruning threshold: p_i, Pruning percentage: pp
Output: $\tilde{grad}_i$

1 **for** $l \in g_i$ **do**
2 N_l = Number of parameters in each layer
3 Pruning index = $\mathrm{N}_l * pp/100$
4 p_i = Find pruning index-th value in l_{g_i}
5 **if** *Each amount in* $l <= p_i$ **then**
6 Remove gradients below threshold in this layer and update l_{g_i}
7 Update gradient based on pruning in each layer: $\tilde{grad}_i$
8 Return pruned gradient $\tilde{grad}_i$

in 2-4 lines of Alg. 2. By analyzing the gradients of each weight, we assess the rate of change in their magnitudes and make decisions regarding whether to prune or not based on this information. To seamlessly integrate this algorithm, we incorporate it into line 18 of our overall SEFL Alg. 1, ensuring that the pruning process is smoothly integrated into the larger training process. By effectively reducing the number of parameters in the network, our approach yields a more compact and efficient model that exhibits improved memory and computational efficiency. This optimization is particularly crucial for edge devices, where resource limitations necessitate highly efficient models.

4 Experimental Results

This section provides an overview of the industrial use case and simulation settings used to evaluate the SEFL method. It includes details about the public dataset used for evaluation, the speech processing and feature extraction, the SER model architecture, and the FL framework setting. We also comprehensively evaluate SEFL. The assessment analyzes SEFL's privacy implications, evaluates its effectiveness in reducing communication traffic and computation time, and determines its accuracy compared with the original model in terms of accuracy. Scalability is examined by analyzing how SEFL performs as client numbers increase and its impact on execution times.

4.1 Use case description and simulation setting

DAIS[3] (Distributed Artificial Intelligent Systems) is a pan-European project that aims to provide trustworthy connectivity and interoperability by combining the Internet of Things with artificial intelligence to create a distributed edge intelligence system to be used in several industrial applications. The project includes extensive industry-driven use cases in domains such as digital life, smart-manufacturing, and mobility. Speech emotion recognition in home entertainment

[3]DAIS Project Website: https://dais-project.eu/

recommendation systems is one of the most important use cases in DAIS, where digital content such as movies are recommended based on the users' emotions. This requires a distributed, efficient, and privacy-preserving SER system: that was one essential motivation for exploring SEFL in FL-SER.

As part of this study, we evaluated SEFL on one of the most widely used SER datasets, namely CREMA-D [5]. CREMA-D is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities. Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad). To train the SER model, we chose the four most commonly occurring emotion labels (neutral, sad, happy, and angry) based on the possible emotions expressed in the sentences.

For speech processing and feature extraction, we generate the Emo-Base feature set using the OpenSMILE toolkit [8]. The Emo-Base feature set is a widely used set of features for SER tasks. These features are extracted from the speech signal and capture various acoustic characteristics of the signal that are associated with different emotions. The features are designed to be highly discriminative for emotion recognition and have been shown to achieve state-of-the-art performance in various SER tasks. After extracting the features, we utilized a multilayer perceptron (MLP) architecture for the SER model and trained it using the FedSGD algorithms. The model consists of two dense layers with layer sizes of [256, 128] and ReLU activation function, along with a 0.2 dropout rate. We set a local training batch size of 20 and a learning rate of 0.1 to accelerate convergence in the FedSGD algorithm.

For the FL training on the CREMA-D dataset, each speaker serves as a unique client since there are 91 distinct speakers in the dataset. We employed 80% of the data for local training at each client and reserved the remaining 20% for validation. To ensure the robustness of our approach, we conducted five experiments with different test folds, and we report the average results of the five-fold experiments. The FL scenarios were conducted over 200 global training epochs. Our experiments were conducted on a Windows 10 Pro environment featuring an Intel(R) Core(TM) i7 CPU @1.80GHz 1.99 GHz processor and 16.0 GB of RAM.

4.2 Privacy considerations

The SEFL method effectively addresses the privacy requirements of the SER application while preventing information leakage by the HBC server. It guarantees that the speech data remains on the end devices throughout the training process, thus satisfying requirement 1.a. To enhance client confidentiality and protect against potential breaches, the method incorporates Paillier homomorphic encryption. By utilizing this encryption technique, the HBC server only has access to ciphertexts, ensuring that no information about the plaintexts is revealed [39]. Consequently, requirement 1.b is met, and the risk against the threat model is significantly reduced.

Furthermore, the method's design ensures that the private key remains accessible to participating clients, preventing unauthorized parties or eavesdroppers from accessing the SER model without compromising the entire cryptosystem. Furthermore, since we can alter the key pair during each iteration, even if the attacker manages to break a few rounds of training results, they would be unable to obtain the final result. This measure fulfils requirement 1.c. However, it is crucial to acknowledge that breaking a cryptosystem, although challenging, is not an impossible task. As suggested in [4], increasing the key size in this method enhances the level of privacy and security. By doing so, the difficulty for potential attackers or eavesdroppers to break the cryptosystem is heightened. However, it's important to note that increasing the key size typically results in longer execution times, as depicted in Fig. 2. Thus, finding the right balance between privacy requirements and execution time is crucial for optimizing the SEFL method.

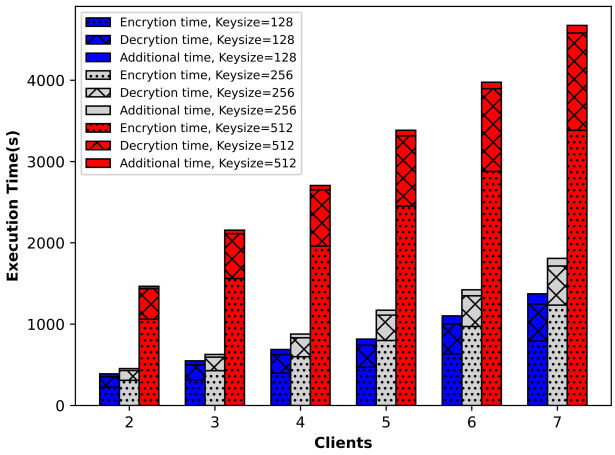

Fig. 2: Impact of key length and number of clients on total execution time.

4.3 Efficiency in terms of communication traffic

A typical scenario in centralized SER applications using the CREMA-D dataset involves a single client transmitting approximately 8-10 MB of speech data to the central server. While the SEFL method employs FL training for SER applications, where instead of sending raw speech data, only the local model update is transmitted, resulting in a substantial reduction of around 70% in data size, as evidenced in Table 2. Additionally, the SEFL method combines Paillier homomorphic encryption and gradient pruning on the client side. Different choices of key sizes and pruning percentages can impact the communication traffic between the client and the server.

Table 2: Communication traffic of SER in FL using PHE, and SEFL based on different key size

Method	**Type of Data**	**PP**	**Communication Traffic (MB)**			
			$KS = 128$	$KS = 256$	$KS = 512$	$KS = 1024$
FL for SER	Plaintext	-	2.18	2.18	2.18	2.18
PHE	Ciphertext	-	7.96	14.4	27.3	53.8
SEFL	Ciphertext	20%	**6.55**	**11.7**	**22.2**	**43.4**
SEFL	Ciphertext	40%	**5.05**	**8.99**	**17.0**	**32.7**
SEFL	Ciphertext	60%	**3.55**	**6.20**	**11.4**	**22.2**
SEFL	Ciphertext	80%	**2.20**	**3.37**	**6.01**	**11.3**

During our experiments, we conducted tests using different key sizes (KS) of 128, 256, 512, and 1024 bits, in combination with gradient pruning percentages (PP) of 20%, 40%, 60%, and 80%. The objective was to determine the optimal key size and pruning percentage. In Table 2, we present an overview of the communication traffic for FL of SER messages across three modes: 1) plaintext, 2) ciphertext for PHE, and 3) ciphertext for SEFL. Our findings indicate that setting the gradient pruning percentage to 80% allows for the use of a larger key size of 1024 bits, resulting in communication traffic of 11.3 MB, which is close to that of a centralized SER model. This configuration achieved an accuracy of 69.89% (as shown in Table 4), which is close to the acceptable levels observed in SER application baselines.

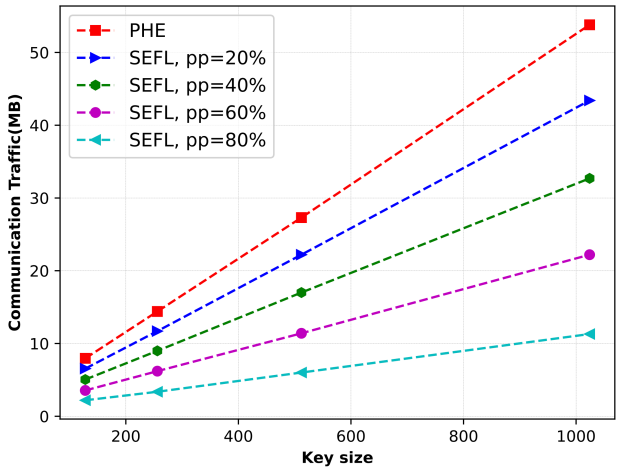

Fig. 3: Communication traffic for PHE and SEFL based on different key sizes.

Additionally, a key size of 512 bits, coupled with a lower pruning percentage of 60% , led to a communication traffic of 11.4 MB, comparable to the communication traffic of a centralized SER model. This configuration achieved an accuracy of 70.32%. Additionally, with a key size of 128 bits and a pruning percentage of 80%, the communication traffic is reduced to approximately 3 MB, similar to FL training use for SER applications. The corresponding accuracy achieved was 69.89%, which is close to the acceptable levels observed in SER application baselines. As illustrated in Fig. 3 and Table 2, doubling the key size in both PHE and SEFL leads to a linear increase in communication traffic. Notably, SEFL outperforms PHE by achieving up to an 80% reduction in clients' ciphertext message size when increasing the pruning percentage from 20% to 80% . This highlights the effectiveness of our proposed SEFL method in reducing communication traffic within FL systems.

4.4 Efficiency in terms of computation time

The SEFL method reduces the computation time required for encryption and decryption, aiming to enhance efficiency and fulfil the requirements. We measured the computation time for encryption and decryption in PHE to evaluate and compare the SEFL approaches for SER applications. By employing the experimental parameters specified in the previous subsections, we obtained the results presented in Table 3 and Fig. 4. Our findings confirm a substantial increase in encryption and decryption time as the key size exponentially grows.

Table 3: Encryption and decryption times of PHE and SEFL in FL-SER based on different key size

Method	**PP**	**Type of Computation**	**Computation Time (s)**			
			$KS = 128$	$KS = 256$	$KS = 512$	$KS = 1024$
PHE	-	Encryption	12.9087	15.4075	38.3163	187.3240
		Decryption	3.6393	4.1267	10.9752	55.1653
SEFL	20%	Encryption	**12.2097**	**13.2047**	**28.2548**	**170.359**
		Decryption	**3.5393**	**4.1433**	**10.1937**	**51.0521**
SEFL	40%	Encryption	**10.8051**	**13.9581**	**25.0631**	**162.3815**
		Decryption	**3.16783**	**4.0514**	**8.1276**	**48.44876**
SEFL	60%	Encryption	**9.3748**	**12.7632**	**26.7844**	**151.6185**
		Decryption	**2.6446**	**4.0597**	**7.70719**	**48.9399**
SEFL	80%	Encryption	**8.2718**	**11.4931**	**25.3584**	**140.4393**
		Decryption	**2.2468**	**3.9250**	**7.7071**	**47.2599**

Our findings, illustrated in Fig. 4 and Table 3, validate that increasing the pruning percentage from 20% to 80% results in a reduction of approximately 10% to 25% in both encryption and decryption times for SEFL. This reduction becomes particularly noticeable when utilizing a larger key size. Consequently,

SEFL effectively reduces computation times while safeguarding user privacy in SER within FL systems.

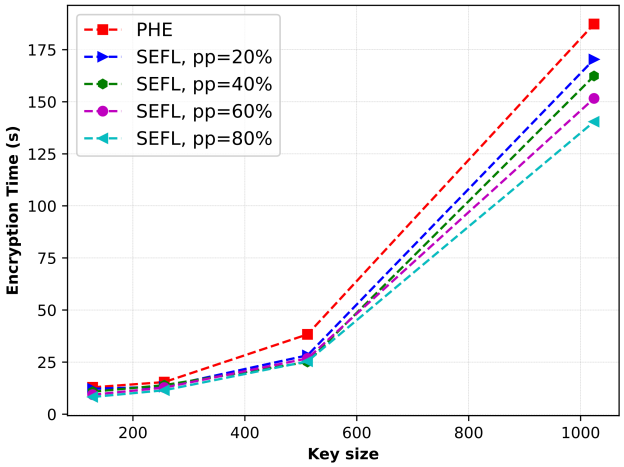

Fig. 4: Encryption times for PHE and SEFL based on different key size.

4.5 Accuracy and other performance related metrics: F1-score, precision, and loss

The requirements section mentioned that centralized SER systems typically achieve a minimum baseline accuracy of 70%. Additionally, it has been observed that there may be a possibility of a 0-5% drop in accuracy when implementing SER in the FL setup [34]. Our initial SER model in the FL setup achieved an accuracy of 72.90%, which meets the requirements. To evaluate the performance of SER in the FL setup using PHE and SEFL, we measured accuracy, F1-score, precision, and loss function. Our analysis indicates that using PHE maintains accuracy and other metrics at the same level as SER performance in FL. However, SEFL has a limited impact on accuracy and other metrics due to using gradient pruning techniques. Despite this limitation, even with the highest pruning percentage, the accuracy remains close to 70% , still satisfying the SER application's requirements as shown in Fig. 5 and Table 4.

We conducted experiments using a key size value of 128, with 20 clients per training round and 200 total epochs. We also applied gradient pruning at levels of 20%, 40%, 60%, and 80%. The results, as shown in Fig. 5 and Table 4, indicate that even with the highest level of gradient pruning applied (i.e., 80%), SEFL has only a minor impact on accuracy and other performance parameters. The accuracy achieved is still very close to the acceptable accuracy in the baseline for SER application.

Table 4: Performance comparison of FL in SER, PHE, and SEFL method, in terms of accuracy, F1-score, precision, and loss.

Method	PP	Accuracy	F1-score	Precision	Loss
FL in SER	-	**72.90%**	**64.84%**	**67.49%**	**0.675025**
PHE	-	**72.87%**	**64.81%**	**67.26%**	**0.678717**
SEFL	20%	**71.82%**	**63.52%**	**65.99%**	**0.686997**
SEFL	40%	**71.55%**	**62.54%**	**65.17%**	**0.689617**
SEFL	60%	**70.32%**	**61.46%**	**65.04%**	**0.706784**
SEFL	80%	**69.89%**	**58.39%**	**60.49%**	**0.74095**

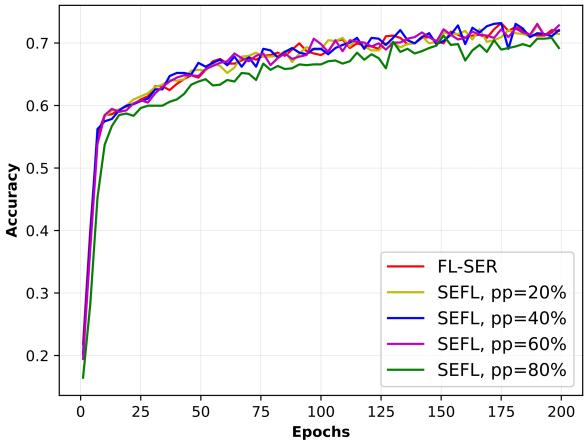

Fig. 5: Accuracy comparison of SER in FL and SEFL.

4.6 Scalability: the impact of different key size values and increasing client numbers on execution time

Scalability is another requirement for SER applications, as they must handle growing numbers of users and data while maintaining performance. The SEFL design itself exhibits scalability, as evidenced by the evaluation of total execution time, which increases linearly by approximately 1.25 times for each additional client across most key sizes. A simulation, illustrated in Figs. 6 and 2, demonstrates the increase in the number of clients from 2 to 7, with key sizes of 128, 256, and 512, while employing a fixed number of training epochs ($T = 20$).

In SEFL, the number of homomorphic operations necessary increases linearly with the number of clients, resulting in a corresponding rise in total execution time. Fig. 2 illustrates the correlation between execution time and the number of clients. It is worth noting that encryption time represents the most time-consuming aspect of the Paillier homomorphic encryption algorithm, taking approximately 2.5 times longer than decryption time, as depicted in Fig. 6.

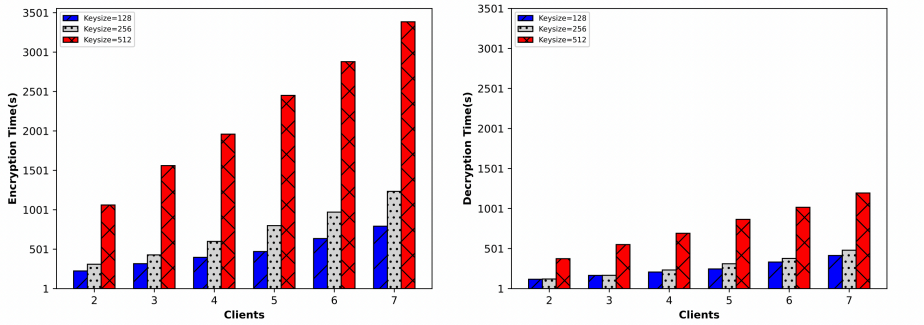

Fig. 6: Impact of key length and number of clients on encryption and decryption time

5 Conclusions and Future work

This paper introduces a novel approach, Secure and Efficient Federated Learning (SEFL), designed specifically for speech emotion recognition applications. SEFL utilizes Paillier homomorphic encryption and gradient pruning to ensure privacy and confidentiality when using FL. This approach significantly reduces computation time and communication traffic while maintaining acceptable model accuracy. Experimental evaluations have shown that SEFL, employing a key size of 1024, achieves a significant reduction of 25% in computation time and an impressive 70% reduction in communication traffic compared to PHE without gradient pruning. With these improvements, the proposed method manages to maintain a satisfactory model accuracy, reaching approximately 69.89%, which fulfils the requirements of SER applications. So, SEFL proves to be an effective solution for SER on resource-constrained edge devices, optimizing resource utilization by striking a balance between privacy and performance. With the increasing importance of trustworthy artificial intelligence in supporting higher levels of autonomy [12], we believe that the proposed method can be extended to other domains with similar requirements.

As part of our future research, we aim to explore the potential of a multi-key homomorphic encryption method in FL for SER. In this method, model updates are first encrypted using an aggregated public key before being shared with a server for aggregation. Decryption requires collaboration among all participating devices. This approach can prevent privacy leakage from publicly shared information in FL and is resistant to collusion between the participating devices and the server.

6 Acknowledgement and Disclaimer

This work was partially supported by the European project DAIS (Distributed Artificial Intelligent Systems) that has received funding from Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement No 101007273. The

KDT JU receives support from the European Union's Horizon 2020 research and innovation program and Sweden, Spain, Portugal, Belgium, Germany, Slovenia, Czech Republic, Netherlands, Denmark, Norway, Turkey. The work reflects only the authors' views; the European Commission is not responsible for any use that may be made of the information it contains.

References

1. Aji, A.F., Heafield, K.: Sparse communication for distributed gradient descent. arXiv preprint arXiv:1704.05021 (2017)
2. Akçay, M.B., Oğuz, K.: Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. Speech Communication **116**, 56–76 (2020)
3. Alnuaim, A.A., Zakariah, M., Alhadlaq, A., Shashidhar, C., Hatamleh, W.A., Tarazi, H., Shukla, P.K., Ratna, R.: Human-computer interaction with detection of speaker emotions using convolution neural networks. Computational Intelligence and Neuroscience **2022** (2022)
4. Barker, E.: Recommendation for Key Management: Part 1 – General. No. NIST Special Publication 800-57 Part 1 Revision 5, National Institute of Standards and Technology (2020)
5. Cao, H., Cooper, D.G., Keutmann, M.K., Gur, R.C., Nenkova, A., Verma, R.: Crema-d: Crowd-sourced emotional multimodal actors dataset. IEEE transactions on affective computing **5**(4), 377–390 (2014)
6. Chang, Y., Laridi, S., Ren, Z., Palmer, G., Schuller, B.W., Fisichella, M.: Robust federated learning against adversarial attacks for speech emotion recognition. arXiv preprint arXiv:2203.04696 (2022)
7. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: International conference on the theory and application of cryptology and information security. pp. 409–437. Springer (2017)
8. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the 18th ACM international conference on Multimedia. pp. 1459–1462 (2010)
9. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive (2012)
10. Fang, C., Guo, Y., Hu, Y., Ma, B., Feng, L., Yin, A.: Privacy-preserving and communication-efficient federated learning in internet of things. Computers & Security **103**, 102199 (2021)
11. Feng, T., Peri, R., Narayanan, S.: User-level differential privacy against attribute inference attack of speech emotion recognition in federated learning. arXiv preprint arXiv:2204.02500 (2022)
12. Flammini, F., Alcaraz, C., Bellini, E., Marrone, S., Lopez, J., Bondavalli, A.: Towards trustworthy autonomous systems: Taxonomies and future perspectives. IEEE Transactions on Emerging Topics in Computing pp. 1–13 (2022). https://doi.org/10.1109/TETC.2022.3227113
13. Goldman, E.: An introduction to the california consumer privacy act (ccpa). Santa Clara Univ. Legal Studies Research Paper (2020)
14. Han, P., Wang, S., Leung, K.K.: Adaptive gradient sparsification for efficient federated learning: An online learning approach. In: 2020 IEEE 40th international conference on distributed computing systems (ICDCS). pp. 300–310. IEEE (2020)

15. Jere, M.S., Farnan, T., Koushanfar, F.: A taxonomy of attacks on federated learning. IEEE Security & Privacy **19**(2), 20–28 (2020)
16. Jiang, Y., Wang, S., Valls, V., Ko, B.J., Lee, W.H., Leung, K.K., Tassiulas, L.: Model pruning enables efficient federated learning on edge devices. IEEE Transactions on Neural Networks and Learning Systems (2022)
17. Jiang, Z., Wang, W., Liu, Y.: Flashe: Additively symmetric homomorphic encryption for cross-silo federated learning. arXiv preprint arXiv:2109.00675 (2021)
18. Khalil, R.A., Jones, E., Babar, M.I., Jan, T., Zafar, M.H., Alhussain, T.: Speech emotion recognition using deep learning techniques: A review. IEEE Access **7**, 117327–117345 (2019)
19. Kim, M., Günlü, O., Schaefer, R.F.: Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2650–2654. IEEE (2021)
20. Kim, T.Y., Ko, H., Kim, S.H., Kim, H.D.: Modeling of recommendation system based on emotional information and collaborative filtering. Sensors **21**(6), 1997 (2021)
21. Kröger, J.L., Lutz, O.H.M., Raschke, P.: Privacy implications of voice and speech analysis–information disclosure by inference. Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers 14 pp. 242–258 (2020)
22. Latif, S., Khalifa, S., Rana, R., Jurdak, R.: Federated learning for speech emotion recognition applications. In: 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 341–342. IEEE (2020)
23. Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., Miao, C.: Federated learning in mobile edge networks: A comprehensive survey. IEEE Communications Surveys & Tutorials **22**(3), 2031–2063 (2020)
24. Liu, P., Xu, X., Wang, W.: Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity **5**(1), 1–19 (2022)
25. Liu, X., Li, H., Xu, G., Chen, Z., Huang, X., Lu, R.: Privacy-enhanced federated learning against poisoning adversaries. IEEE Transactions on Information Forensics and Security **16**, 4574–4588 (2021)
26. Ma, X., Lin, S., Ye, S., He, Z., Zhang, L., Yuan, G., Tan, S.H., Li, Z., Fan, D., Qian, X., et al.: Non-structured dnn weight pruning—is it beneficial in any platform? IEEE transactions on neural networks and learning systems **33**(9), 4930–4944 (2021)
27. Madanian, S., Parry, D., Adeleye, O., Poellabauer, C., Mirza, F., Mathew, S., Schneider, S.: Automatic speech emotion recognition using machine learning: Mental health use case. In: Pacific Asia Conference on Information Systems. p. 1 (2022)
28. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
29. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International conference on the theory and applications of cryptographic techniques. pp. 223–238. Springer (1999)
30. Pathak, M.A.: Privacy-preserving machine learning for speech processing. Springer Science & Business Media (2012)
31. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S.: Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE **105**(12), 2295–2329 (2017)

32. Tanko, D., Dogan, S., Demir, F.B., Baygin, M., Sahin, S.E., Tuncer, T.: Shoelace pattern-based speech emotion recognition of the lecturers in distance education: Shoepat23. Applied Acoustics **190**, 108637 (2022)
33. Tao, Z., Li, Q.: esgd: Commutation efficient distributed deep learning on the edge. HotEdge p. 6 (2018)
34. Tsouvalas, V., Ozcelebi, T., Meratnia, N.: Privacy-preserving speech emotion recognition through semi-supervised federated learning. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). pp. 359–364. IEEE (2022)
35. Tuncer, T., Dogan, S., Acharya, U.R.: Automated accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques. Knowledge-Based Systems **211**, 106547 (2021)
36. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
37. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q., Poor, H.V.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security **15**, 3454–3469 (2020)
38. Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., Liu, Y.: {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In: 2020 USENIX annual technical conference (USENIX ATC 20). pp. 493–506 (2020)
39. Zhang, J., Chen, B., Yu, S., Deng, H.: Pefl: A privacy-enhanced federated learning scheme for big data analytics. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6. IEEE (2019)
40. Zhao, Y., Zhao, J., Yang, M., Wang, T., Wang, N., Lyu, L., Niyato, D., Lam, K.Y.: Local differential privacy-based federated learning for internet of things. IEEE Internet of Things Journal **8**(11), 8836–8853 (2020)
41. Zhu, H., Wang, R., Jin, Y., Liang, K., Ning, J.: Distributed additive encryption and quantization for privacy preserving federated deep learning. Neurocomputing **463**, 309–327 (2021)