Tabriz Journal of Electrical Engineering (TJEE), vol. 53, no. 1, Spring 2023 Serial no. 103
DOI: 10.22034/tjee.2023.16064

Optimizing Data Transfer and Convergence Time for Federated Learning based on NSGA II

Mahdi Fallah[1], Pedram Salehpour[2*]

Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran.
[1] m.fallah1400@ms.tabrizu.ac.ir
[2] p.salehpoor@tabrizu.ac.ir

[*]Corresponding author

Received: 08/01/2023, Revised: 08/02/2023, Accepted: 25/03/2023.

Abstract
Face recognition from digital images is used for surveillance and authentication in cities, organizations, and personal devices. Internet of Things (IoT)-powered face recognition systems use multiple sensors and one or more servers to process data. All sensor data from initial methods was sent to the central server for processing, raising concerns about sensitive data disclosure. The main concern was that all data from all sectors that could contain confidential information was placed in a central server. Federated learning can solve this problem by using several local model training servers for each region and a central aggregation server to form a global model in IoT networks. This article presents a novel approach to optimize data transfer and convergence time in federated learning for a face recognition task using Non-dominated Sorting Genetic Algorithm II (NSGA II). The aim of the study is to balance the trade-off between training time and model accuracy in a federated learning environment. The results demonstrate the effectiveness of the proposed approach in reducing data transfer and convergence time, leading to improved performance in face recognition accuracy. This research provides insights for researchers and practitioners to enhance the efficiency of federated learning in real-world applications.

Keywords
Federated Learning, NSGA II, few-shot learning, face recognition.

1. Introduction

The face is one of modern society's most illuminating forms of communication. Contrary to face recognition by people to understand their peers, which is a natural phenomenon, facial geometry recognition by machines is still a difficult challenge. Face recognition is identifying a person using a digital photograph of their face [1]. People have recently been authenticated using biometric identity techniques, including face recognition, fingerprint recognition, iris recognition, etc. An individual is recognized using a digitized facial picture in face recognition, which has been the subject of study in a very active research community for more than a decade. A person is recognized based on some distinctive facial characteristics [2]. Face recognition authentication technology has various applications, such as city and organizational monitoring systems [3]. However, recognizing people's faces from diverse image data with different characteristics is a time-consuming task that must be done with high accuracy [4].
In image-based learning and recognition, due to the large volume of data, processing and learning the model for recognition requires high processing power, including a powerful processor, and is very time-consuming. On the other hand, if there is a lack of training data in the model training phase, the system's accuracy will be very low in classification and recognition task [27]. Unlike conventional deep learning methods that require a large number of data samples to train their model, humans can learn by seeing very few samples of a phenomenon [5]. In recent years, researchers in image processing have developed a new concept called few-shot learning in machine vision to create the ability to learn with limited labeled samples. In this type of learning, in addition to learning based on the training labeled data, the system uses the previous information in the trained categories related to the new category to strengthen the model for the new category [6].
In theory, the existence of several objectives results in a set of optimum solutions, typically termed Pareto-optimal solutions rather than a single ideal answer. One of these Pareto-optimal solutions cannot be deemed superior to the other without sufficient information [7]. In order to solve this, a user must identify as many Pareto-optimal solutions as possible [8]. By focusing on a single Pareto-optimal solution at a time, traditional optimization approaches (including the multi-criteria decision-making methods) advise reducing the multiobjective issue to a single-objective optimization problem [28]. Since using such a strategy to identify many answers, it must be applied numerous times to discover a new solution with each simulation run [9]. To find the optimal answer for multi-objective problems, researchers have introduced a method called NSGA II. This method, which is used to optimize the answer, is based on the conventional genetic algorithm, but unlike

it, it is multi-purpose [10]. In the field of face recognition, there is a trade-off between training time and recognition accuracy, and as much as we try to increase accuracy, time is lost. A balance between these goals (training time and recognition accuracy) must be created to find the optimal answer [11].

The high volume of information in image processing operations such as face recognition and the high cost of transferring this information from edge servers to central servers make using centralized methods, such as centralized deep learning (CDL), non-optimal [12, 13]. On the other hand, local model training methods such as localized deep learning (LDL) cannot be performed in few-shot learning operations because local servers only have local models trained with limited local data [14, 15]. In addition, security concerns about information leakage from central servers make it impossible to use centralized methods in sensitive situations because releasing this information means releasing the identity information of different people [16, 17]. Since Google initially launched it, federated learning (FL) has proved essential in enhancing the performance of a variety of applications. FL is an environment that makes cooperative machine learning possible [18, 19].

The main contributions of this work are as follows:

- We present a novel architecture for few-shot learning in distributed Internet of Things (IoT) networks that is based on federated learning.
- By utilizing NSGA II optimizer algorithm, we defined two fitness functions for training time and accuracy to make a balance between these two goals.
- The federated learning architecture, along with the combination of locally-generated models from local data, was utilized to optimally distribute traffic and processing load across local servers while preserving personal data privacy. This resulted in the generation of well-trained global models.

The remainder of this article is organized as follows. In part 2, we reviewed the previous works in the field of face recognition and the methods used to increase accuracy and reduce training time. Section 3 presents a new face recognition method using the federated learning algorithm and NSGA II optimizing algorithm with the two goals of increasing recognition accuracy and reducing training time. Section 4 presents the results obtained from the simulation of the proposed method and its superiority over previous works. Finally, section 5 summarizes the proposed method and its results.

2. Related Works

In recent years, much research has been done in the field of few-shot learning, which aims to improve learning and recognition with a small number of labeled training data. Liang et al. in [20] introduced an OICS-VFSL model for microservice-oriented intrusion detection to apply to IoT devices with limited resources. They presented a unified framework for few-shot learning that used variational feature representation. It primarily consisted of two fundamental operations: 1) intra-class distance optimization based on variational feature representation and 2) inter-class distance optimization based on feature concatenation. The evaluation results showed the appropriate performance of this method in detecting zero-day attacks with imbalanced data in the Internet of Things (IoT) networks. Despite the effectiveness of this method in detecting zero-day attacks, the combination of incremental clustering, feature selection, and labeling can lead to a higher computational complexity compared to simpler clustering methods. Moreover, there is a risk of overfitting the data if the model is too complex or if there is not enough training data. Thirdly, the quality of the cluster labels generated by the OICS-VFSL model may depend on the quality of the feature selection and labeling methods used.

To address the low accuracy of existing algorithms in light of inadequate traffic, Zhao et al. in [21] presented Festic, a few-shot learning-based solution for IoT traffic classification. Festic outperformed other state-of-the-art algorithms like BSNN and IoT Sentinel. Festic enhances recall by 10.19%, precision by 9.13%, and F1-measure by 9.80% compared to BSNN. Festic raises recall by 8.81%, accuracy by 8.10%, and F1-measure by 8.87% compared to IoT Sentinel. Although the method resulted in an increase in accuracy, it was observed to have a prolonged training time for the model.

Yang et al. in [22] put forth a confidence-based sample selection approach that simultaneously integrates the classification and detection tasks. Analyses showed that the highest quantity of information they chose outperforms randomly selected samples, demonstrating the efficacy of their strategy for selecting redundant IoT data sets. The model's accuracy is 2.7% more than random selection when the classification network is employed to test the samples. Despite this, confidence-based sample selection can introduce bias into the training process if the samples selected for training are not representative of the underlying distribution of the data. In addition, confidence-based sample selection adds an additional layer of complexity to the training process, making it more difficult to debug or understand why the model makes certain predictions.

Jia et al. in [23] provide a unique few-shot learning-based technique for classifying IoT traffic that can identify IoT traffic effectively with just a few labeled samples and suggest a more reliable traffic characteristic. To classify the different forms of IoT traffic, FITIC uses the raw IoT flow as its input, creating features from the two parts of flow statistics and payload bytes before introducing a classification model based on feature similarity. Compared to BSNN and FS-Net, FITIC's average F1-measure was 14.13% and 8.11% higher, respectively. Additionally, they discover that FITIC outperforms the other two approaches in its capability to label novel classes, with an average recall of 96.5% for novel classes, 9.6% greater than BSNN, and 9.58% higher than FS-Net. However, FITIC may not be well-suited for large-scale IoT traffic datasets if the computational cost of the method becomes too high.

When there needed to be more samples, Yang et al. in [24] studied a technique for fusing IoT devices with deep learning models. For limited sample sets, they provided a text sentiment analysis model known as FSLM. Using the self-attention mechanism, they trained a basic model

Tabriz Journal of Electrical Engineering (TJEE), vol. 53, no. 1, Spring 2023 Serial no. 103
DOI: 10.22034/tjee.2023.16064

to extract sentiment features from the input text. They then constructed a Siamese network using two self-attention models with identical parameters. Finally, used Mahalanobis distance to assess how similar the feature vectors of various categories are to one another. According to the test results, their suggested FSLM model offers superior classification performance over other models. The FSLM approach has some benchmark importance for situations when text sentiment analysis lacks sample numbers. On the other side, fusing IoT devices through FSLM can present several drawbacks, including complexity, scalability issues, integration difficulties, latency in the fusion process, security concerns for sensitive data, high costs for implementation and maintenance, and the need for specialized hardware, software, and personnel. As the number of IoT devices increases, the scalability of the FSLM system may become an issue, while integrating different devices with varying protocols and hardware configurations can also be challenging. FSLM can also introduce latency, affecting real-time performance, and pose security risks if sensitive data is transmitted between devices. The cost of implementation and maintenance can also be high, making it a costly solution for organizations.

3. Methodology

The federated learning structure is the central component of the method that we have proposed. The first layer of this architecture, as shown in Fig. 1, is comprised of surveillance cameras deployed in several different surveillance zones capable of capturing images of high quality and degree of precision. Network infrastructure equipment such as switches and routers are utilized in each surveillance area in order to connect the cameras to the local server. The local servers in each region are connected to the central server set, which may be hosted internally by the organization or externally by a third party.

Collecting and processing multimedia data is delegated to the local servers in each area. The memory space required for high-quality multimedia files is typically ample, and it is not easy to transfer such files across networks. In order to improve quality of service (QoS) metrics like response time, one of the primary functions of local servers is to perform load balancing in order to find the best path from data collecting equipment and sensors like cameras to the local server. Managing the local network infrastructure to enhance QoS metrics is one of the most critical tasks of local servers.

In the default mode of the federated learning algorithm, the data collected from the cameras of each area are processed by the local training server and using deep neural network learning algorithms to build the machine learning model. The main drawback of this model is that it was created only with local data collected from that area. When faced with new samples unfamiliar with that area, it cannot perform proper face recognition. On the other hand, few-shot learning also needs models from other regions to strengthen the learning model to create a more robust model for face recognition. For this purpose, there is a central server to collect all these local models from different areas and create a global model by aggregating all local models in several communication rounds with local servers. By sending this global model to all local servers, these servers can perform few-shot learning because now they have aggregated models of other areas. However, nevertheless, no area has access to the raw data of other areas, and privacy is maintained for each area's data. In addition, because only local and global models are transmitted between the central server and local servers instead of raw data transfer, there is no concern about private information leakage.

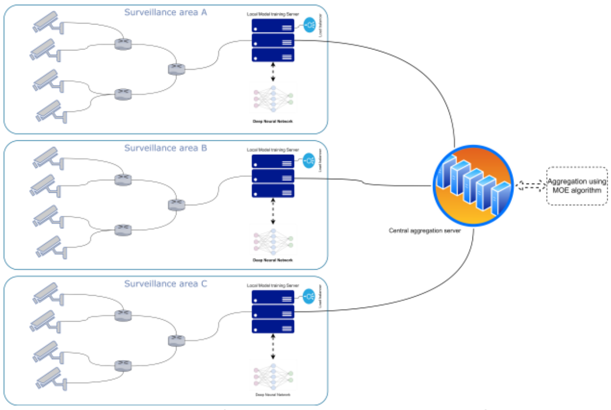

Fig. 1. Arcitecture of proposed method based on federated learning with NSGA II optimizer.

Our proposed method has two main goals: improving the training time and increasing the accuracy of face recognition. The local training server uses an NSGA II module that periodically evaluates the generated models to find the best solution to satisfy these two goals. NSGA II is an evolution-based algorithm that uses principles from genetic algorithms and non-dominated sorting to find a set of non-dominated solutions, known as the Pareto front, representing a trade-off between the objectives. NSGA II works by first generating an initial population of candidate solutions. The solutions are then evaluated based on the multiple objectives, and their non-domination levels are determined. The solutions are sorted into different levels of non-domination, with solutions in the first level being the most non-dominated. The solutions are then subjected to genetic operations such as crossover and mutation to generate a new population. The process is repeated until a satisfactory set of solutions is obtained.

In the context of federated learning, NSGA II can optimize data transfer and convergence time by finding a trade-off between the two objectives. NSGA II can determine the optimal amount of data to be transferred between the client devices and the server and the optimal frequency and duration of communication to ensure efficient data transfer and quick convergence.NSGA II can also handle the dynamic nature of federated learning, where the network and data distribution can change over time. This makes NSGA II well-suited for real-world federated learning scenarios, where the optimization of data transfer and convergence time is critical for the success of the learning process.

Unlike the genetic algorithm, two fitness functions can be used in this algorithm to evaluate the generated generation and select the best generation that creates a balance between the two fitness functions. Each of the local models has different parameters, and each has a

DOI: 10.22034/tjee.2023.16064

different effect on the final efficiency and accuracy of that model. First of all, we define the fitness function for the accuracy of the model according to equation 1 [25]:

$$V(j)=\left|\sum_{j=1}^{53} M_j^{i,k}-\sum_{j=1}^{53} M_j^{i,(k-1)}\right| \tag{1}$$

In this equation we take $M^{i,k}$ as model parameter matrix from the *i*-th model after *k*-th federated learning global model aggregation rounds. Accordingly, $M_j^{i,k}$ is the *j*-th column of matrix $M^{i,k}$. Sum of all absolute values of all parameters in the *j*-th layer is $\left|\sum M_j^{i,k}\right|$. The engagement of the *j*-th column on total model accuracy can be calculated as *v*(*j*). When the value of *v*(*j*) increases, the significance of data from column *j* increases in the model. For federated model aggregation, each surveillance area's local model training server sorts the *v*(*j*) values of the model's parameter matrix by decreasing order, then sends only the parameters of the first *n* columns to the federated learning centralized aggregation server. The particular reason for the employment of this fitness function is that the best models and parameters that have the most significant effect on face recognition will always be sent to the central server for aggregation, which will eventually improve the accuracy of the global model. Additionally, due to a decrease in the amount of data transmitted between the local servers and the central server, there is less traffic load on the network, and the quality of service (QoS) is improved.

Reducing the amount of time needed for training is the second objective for which a fitness function should be defined. The fact that time directly correlates with the amount of energy used by electronic equipment is one of the essential aspects to consider in this scenario. The longer the processing takes, the more energy will be used. The second fitness function for the NSGA II module should be defined to optimize the training time. According to the equation presented by Dev et al. in [26], the reduction of model training time depends on factors such as the impact of each data in global model aggregation, the distance between central aggregation server and local model training server, and the distance between each sensor and local model training server. Moreover, it depends on the network's delay in data transmission, the amount of data produced by each sensor, the amount of local model produced by each local model training server in each area, and the total volume of data and model. We define the second fitness function as equation 2:

$$\begin{aligned} OF_i &= \nu * OF_{local}+\phi * OF_{load} \\ &+\chi OF_{data}+\varphi(1-OF_{dis}) \\ &+\omega(1-OF_{delay}) \end{aligned} \tag{2}$$

where the total of the weighted parameters (ν,ϕ,χ,φ,ω) should equal zero.

After transmitting and receiving every packet, a node expends a set amount of energy. As a result, this leftover energy significantly contributes to increased network lifespan and performance. In addition, in the discussion of few-shot learning, the fewer data samples the local model is formed by the local server, the more optimal the final time will be. Based on this, the energy of the sensors after sending images to the local server will be according to equation 3:

$$X_{p+1}(C_N^p)=X_p(C_N^p)-X(C_N^p) \tag{3}$$

where $X_{P+1}\left(C_N^p\right)$ determines the energy left after the data packets are sent to the local model training server. $X\left(C_N^p\right)$ is the p-th sensor's energy loss.

Equation 4 is also used to determine the local model training server's potential energy:

$$X_{p+1}(C_{local}^i)=X_p(C_{local}^i)-X(C_{local}^i) \tag{4}$$

consequently, upon the receipt of data packets from sensors, $X_{P+1}\left(C_{local}^i\right)$ determines the energy kept in local model training server. $X\left(C_{local}^i\right)$ is the energy lost by the i-th local model training server. Furthermore, finally, we take the fitness function for data generation as equation 5:

$$\begin{aligned} OF_{data} &= \frac{1}{k}\left\{\sum_{p=1}^{k} X(C_N^p)\right\} \\ &+\frac{1}{T_{local}}\left\{\sum_{i=1}^{local} X(X_{local}^i)\right\} \end{aligned} \tag{5}$$

To calculate the fitness function related to the local model OF_{local} and the fitness function for traffic load OF_{load}, we used Xively. This Google IoT platform provides Cloud-based APIs to connect and develop IoT applications and takes advantage of the MQTT protocol to transfer data. In this situation, we want a model that is suitable for few-shot learning and does not have a heavy load on the network. For performance analysis, the data pertaining to load and the local model are input into Xively. Finally, the output of Xively is used to calculate equation 2.

Calculating the distance between each sensor and the local server, as well as the distance between the local server and the central aggregation server, is the next step in the process of determining the fitness function. This distance is determined using equation 6, which can be found below:

$$OF_{dis}=\sum_{p=1}^{P}\sum_{i=1}^{T_{local}}\frac{\left\|D_N^p-D_{local}^i\right\|+\left\|D_{local}^i-D_{Central}\right\|}{Q*Q} \tag{6}$$

where $\left\|D_N^p-D_{local}^i\right\|$ represents the distance between the p-th node and the local server, $\left\|D_{local}^i-D_{central}\right\|$ represents the distance between the local server and the central aggregation server. *Q* also denotes the furthest distance over which each camera is capable of capturing photos.

The last component in calculating the fitness function is related to the network delay. When calculating the delay associated with passing data, both the transmission delay (T_t) and the propagation delay (T_p) are taken into consideration as factors. The effectiveness of the network may be improved by reducing the amount of time that passes between the transmission of data packets from their origin to their destination. Equation 7 is a representation of the fitness function that is used to evaluate the latency:

$$OF_{delay}=\frac{Max\sum_{i=1}^{T_{local}} local_i}{K} \tag{7}$$

In equation 7, K is total number of sensors in each area and $Max\sum_{i=1}^{T_{local}} Local_i$ represents the delay.

Tabriz Journal of Electrical Engineering (TJEE), vol. 53, no. 1, Spring 2023 Serial no. 103
DOI: 10.22034/tjee.2023.16064

4. Evaluation and Results

In recent studies such as [29], the authors have proposed utilizing a Genetic Algorithm (GA) based optimization technique within the SMEC environment, referred to as GAME. This approach aims to determine the optimal solution by employing the GA optimization technique in the SMEC environment. To evaluate the effectiveness of the proposed method, a comprehensive analysis was conducted and compared with existing offloading policies using established benchmark datasets. The results of this analysis confirmed the superiority of the GA-based optimization technique within the SMEC environment. The critical drawbacks of using Genetic Algorithms (GA) for optimizing SMEC systems are slow convergence speed, parameter sensitivity, and a tendency to get stuck in local optima. GA can take a long time to find the optimal solution for complex SMEC systems, making it challenging to find the best solution in a timely manner. Additionally, GA is sensitive to the choice of parameters, such as mutation rate and crossover rate, which can significantly impact the optimization results and make the optimization process unpredictable. Finally, GA may get trapped in local optima, leading to sub-optimal results, which can reduce the effectiveness of the optimization process for SMEC systems.

In another study [30], researchers presented a framework for load balancing in fog computing environments called OLBA (Online Load Balancing Algorithm). This framework aims to enhance critical quality of service (QoS) metrics, including turnaround time, resource utilization, response time, and delay, by balancing the workload among fog devices. The framework utilizes a Particle Swarm Optimization (PSO) method to identify the optimal solution by first locating the local best and then comparing all the local bests to find the global best. This approach can be slow to converge to the optimal solution, resulting in more extended load balancing times and decreased system efficiency. It also requires careful tuning of its parameters and can be prone to get stuck in local minima, leading to suboptimal solutions. Additionally, PSO may not scale well to larger systems with high dynamic loads.

However, because the environment is changed in this paper, it is not fair to campare previous results of PSO and GA in our environment. The proposed method is compared with PSO and GA algorithms in a new IoT environment based on federated learning architecture responsible for the face recognition task.

To evaluate the proposed method, we simulated a network with 50 sensors in a system with an 8th generation Intel Core i7 processor and 16GB of RAM by MATLAB programming language. In the first test, according to Fig. 2, each sensor produced 50 images. In the first phase, sensors injected 500 images to the network. By comparing the optimization done by the NSGA II algorithm, our proposed method reached convergence with 423 images. At the same time, PSO and GA converged with 462 and 470 samples, respectively. In 1000 samples, our proposed method converged with 798 images, and PSO and GA algorithms converged with 857 and 895 images, respectively. By increasing the number of samples to 1500, NSGA II, PSO, and GA algorithms converged with 1104, 1215, and 1377 images, respectively. Finally, with the number of 2000 samples, the NSGA II algorithm could converge the final model with 330 and 453 fewer samples than the PSO and GA algorithms. Generally, our proposed algorithm converges to an optimal solution with a smaller number of data samples in each phase of the experiment. This is a crucial factor that influences the overall training time of the model. Upon comparison with traditional optimization algorithms such as GA and PSO, it can be inferred that the training time of the model is significantly reduced when using the proposed algorithm. This test shows that the proposed optimization algorithm can converge with a smaller number of samples than the state-of-the-art algorithms and execute few-shot learning optimally.

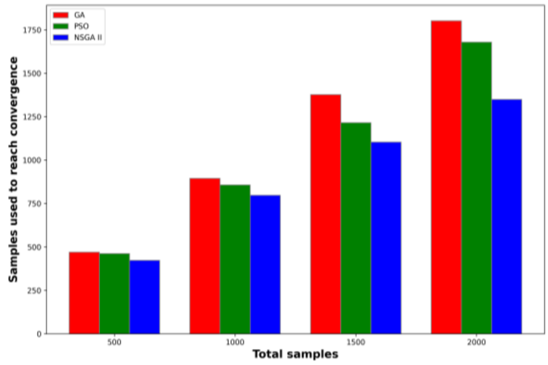

Fig. 2. Samples needed to converge the model by the proposed algorithm compared to PSO and GA.

In the second experiment, we assessed the recognition accuracy of our proposed algorithm, as depicted in Fig. 3, compared to the PSO and GA algorithms over a set period. During the first hour, our proposed method achieved an accuracy of 18%, while PSO and GA reached 20% and 21%, respectively. In the second hour, there was a noticeable improvement in the accuracy of our proposed method, with an increase of 34%, compared to a 25% and 18% improvement in PSO and GA, respectively. By the third hour, the recognition accuracy of our proposed method reached 82%, whereas PSO and GA had 74% and 78% accuracy, respectively. Finally, in the fourth hour, the accuracy of our proposed method and PSO remained unchanged, but GA saw a 1% improvement from the previous hour. These results demonstrate that the recognition accuracy of our proposed optimization method surpasses that of PSO and GA algorithms over the same period.

The comprehensive outcomes of the evaluations demonstrate that our proposed method outperforms state-of-the-art techniques in multiple ways. Firstly, by training the model with a limited number of samples, it reduces the computational burden and enables efficient few-shot learning. Secondly, within the same time frame, our method demonstrates an average recognition accuracy of 5.25% higher than PSO and 4.25% higher than GA for face recognition tasks.

In the following evaluation, we studied the impact of individual parameters of the fitness function described in equation 2 on the convergence of the final model and compared the results with the aggregate state described in equation 2. As shown in Fig. 4, when using the

OF_{data} as fitness function, the final model converged using 1870 data samples. Using the OF_{dis}, OF_{delay}, OF_{local}, and OF_{load} parameters as single parameter fitness functions, the final model converged using 1645, 1901, 1673, and 1477 data samples, respectively, out of the 2000 data samples entered into the network. However, when using the aggregate state of all the parameters described in equation 2 (OF_i), the final model converged using only 1341 data samples. This evaluation suggests that the cumulative use of these parameters significantly improves the convergence of the final model with fewer data samples. The results of this evaluation highlight the importance of the parameters of the fitness function in determining the convergence of the final model and demonstrate the efficiency of the cumulative use of these parameters in reducing the number of required data samples.

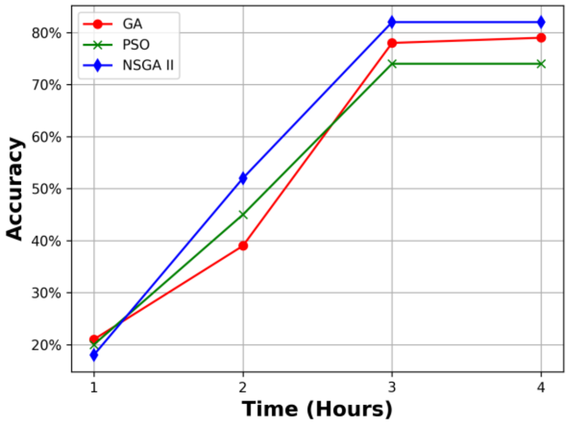

Fig. 3. Accuracy of the proposed method compared to PSO and GA over time.

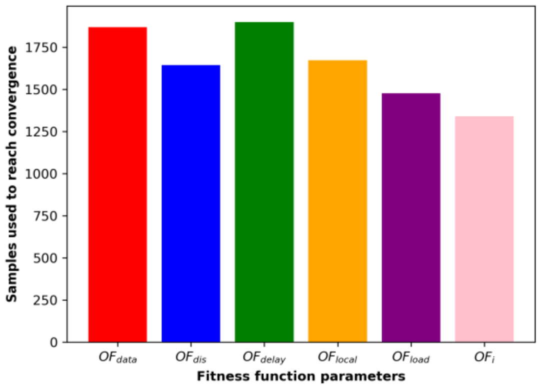

Fig. 4. Impact of each parameters of equation 2 in model convergence.

5. Conclusion

In a nutshell, this paper aims to describe a novel approach to face recognition in IoT networks. In the suggested approach, reducing the time spent on training while simultaneously improving recognition accuracy was two of the goals. We utilized the NSGA II optimization algorithm in order to accomplish these objectives. By defining the first fitness function, we endeavored to select the most appropriate local parameter to aggregate the global model. For the second objective, the second fitness function was defined by considering the distance between the local server and central server and cameras, the amount of energy used, the amount of traffic load, and how practical the local models are. This was done while keeping in mind the principle that the more time spent training the model, the more energy consumed. Our system is built mainly on federated learning, which creates a local model by considering data security and using a deep learning algorithm. This is the primary foundation on which our system is based. A global model is created by performing a combination of local models obtained from various local servers. This model is then sent to all of the local servers.
It is possible to implement few-shot learning in the most effective manner using the federated learning method and aggregating models from all locations.
In addition to the evaluation of the proposed approach in multiservice networks, future work could also focus on exploring the scalability of the system and its performance under different network conditions and configurations. Additionally, integrating additional data sources and considering alternative optimization algorithms could further improve the recognition accuracy and energy efficiency of the system. Furthermore, incorporating user privacy concerns and testing the robustness of the system to attacks and data tampering can also be a part of the future work. Overall, the proposed approach in this paper opens up avenues for further research and improvement in the field of face recognition in IoT networks.

6. References

[1] A. K. Agrawal and Y. N. Singh, "Evaluation of face recognition methods in unconstrained environments," Procedia Computer Science, vol. 48, pp. 644-651, 2015.
[2] P. Kaur, K. Krishan, S. K. Sharma, and T. Kanchan, "Facial-recognition algorithms: A literature review," Medicine, Science and the Law, vol. 60, no. 2, pp. 131-139, 2020.
[3] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones," in Proceedings of the 12th annual international conference on Mobile systems, applications, and services, 2014, pp. 109-122.
[4] M. Dabbah, W. Woo, and S. Dlay, "Secure authentication for face recognition," in 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing, 2007: IEEE, pp. 121-126.
[5] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7229-7238.
[6] R.-Q. Wang, X.-Y. Zhang, and C.-L. Liu, "Meta-prototypical learning for domain-agnostic few-shot recognition," IEEE Transactions on Neural Networks and Learning Systems, 2021.
[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in International conference on parallel problem solving from nature, 2000: Springer, pp. 849-858.
[8] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and

NSGA-II," IEEE transactions on evolutionary computation, vol. 13, no. 2, pp. 284-302, 2008.
[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE transactions on evolutionary computation, vol. 6, no. 2, pp. 182-197, 2002.
[10] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of NSGA-II for optimizing machining process parameters," Procedia Engineering, vol. 15, pp. 3978-3983, 2011.
[11] S. Verma, M. Pant, and V. Snasel, "A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems," Ieee Access, vol. 9, pp. 57757-57791, 2021.
[12] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semi-Supervised Federated Learning Based Intrusion Detection Method for Internet of Things," IEEE Internet of Things Journal, 2022.
[13] R. Zhao, L. Yang, Y. Wang, Z. Xue, G. Gui, and T. Ohtsuki, "A Semi-Supervised Federated Learning Scheme via Knowledge Distillation for Intrusion Detection," in ICC 2022-IEEE International Conference on Communications, 2022: IEEE, pp. 2688-2693.
[14] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," Knowledge-Based Systems, vol. 216, p. 106775, 2021.
[15] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," Computers & Industrial Engineering, vol. 149, p. 106854, 2020.
[16] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3930-3944, 2021.
[17] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," Future Generation Computer Systems, vol. 115, pp. 619-640, 2021.
[18] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," Information Processing & Management, vol. 59, no. 6, p. 103061, 2022.
[19] D. C. Nguyen et al., "Federated learning meets blockchain in edge computing: Opportunities and challenges," IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12806-12825, 2021.
[20] W. Liang, Y. Hu, X. Zhou, Y. Pan, I. Kevin, and K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," IEEE Transactions on Industrial Informatics, vol. 18, no. 8, pp. 5087-5095, 2021.
[21] Z. Zhao, Y. Lai, Y. Wang, W. Jia, and H. He, "A Few-Shot Learning Based Approach to IoT Traffic Classification," IEEE Communications Letters, vol. 26, no. 3, pp. 537-541, 2021.
[22] J. Yang, Z. Zhang, and Y. Li, "Agricultural Few-Shot Selection by Model Confidences for Multimedia Internet of Things Acquisition Dataset," in 2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), 2022: IEEE, pp. 488-494.
[23] W. Jia, Y. Wang, Y. Lai, H. He, and R. Yin, "FITIC: A Few-shot Learning Based IoT Traffic Classification Method," in 2022 International Conference on Computer Communications and Networks (ICCCN), 2022: IEEE, pp. 1-10.
[24] L. Yang, Y. Li, J. Wang, and N. N. Xiong, "FSLM: An intelligent few-shot learning model based on Siamese networks for IoT technology," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9717-9729, 2020.
[25] Y. Liu et al., "Fedvision: An online visual object detection platform powered by federated learning," in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, vol. 34, no. 08, pp. 13172-13179.
[26] K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, S. Bhattacharya, P. Hegde, and S. Singh, "Energy optimization for green communication in IoT using harris hawks optimization," IEEE Transactions on Green Communications and Networking, vol. 6, no. 2, pp. 685-694, 2022.
[27] R. Khodabakhsh, M. R. Haghifam, and M. K. Sheikh El Eslami, "Transmission system congestion management with demand-side flexibility resources considering distribution system performance improvement," Tabriz Journal of Electrical Engineering, vol. 52, no. 2, pp. 103-114, 2022, doi: 10.22034/tjee.2022.15432.
[28] Y. Darmani and M. Sangelaji, "QDFSN: QoS-enabled Dynamic and Programmable Framework for SDN," Tabriz Journal of Electrical Engineering, vol. 51, no. 1, pp. 1-10, 2021. [Online]. Available: https://tjee.tabrizu.ac.ir/article_13279_ed1fc1e794b4dc90ea1723a22dbc17b0.pdf.
[29] S. Chakraborty and K. Mazumdar, "Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing," Journal of King Saud University-Computer and Information Sciences, 2022.
[30] S. Harnal, G. Sharma, N. Seth, and R. D. Mishra, "Load balancing in fog computing using qos," Energy Conservation Solutions for Fog-Edge Computing Paradigms, pp. 147-172, 2022.