



Performance Improvement of Fuzzy C-Means Clustering Algorithm by Optimized Early Stopping for Inhomogeneous Datasets

Chae-Rim Han^{1*}, Sun-Jin Lee², and Il-Gu Lee^{1,2*}

¹Department of Convergence Security Engineering, Sungshin Women's University, 02844, Korea

²Department of Future Convergence Technology Engineering Sungshin Women's University, 02844, Korea

Abstract

Responding to changes in artificial intelligence models and the data environment is crucial for increasing data-learning accuracy and inference stability of industrial applications. A learning model that is overfitted to specific training data leads to poor learning performance and a deterioration in flexibility. Therefore, an early stopping technique is used to stop learning at an appropriate time. However, this technique does not consider the homogeneity and independence of the data collected by heterogeneous nodes in a differential network environment, thus resulting in low learning accuracy and degradation of system performance. In this study, the generalization performance of neural networks is maximized, whereas the effect of the homogeneity of datasets is minimized by achieving an accuracy of 99.7%. This corresponds to a decrease in delay time by a factor of 2.33 and improvement in performance by a factor of 2.5 compared with the conventional method.

Index Terms: Deep reinforcement learning, Early stopping, Neural network, Overfitting

I. INTRODUCTION

Neural networks can restore several damaged neurons or distorted data owing to their fault tolerance and parallelism model relationships between complex data, thus allowing the prompt learning of nonlinear relationships between large-scale data. Owing to these advantages, neural networks have been used in various fields, such as text, voice, and image recognition, in addition to natural language processing.

During the training process of an artificial intelligence (AI) model, overfitting occurs when the computational volume increases or the dataset is excessively optimized [1]. When overfitting occurs, the model achieves high accuracy on the learning data; however, the accuracy on the new data is lower owing to the low amount of learning, which signifi-

cantly influences the neural network and network performance. As a solution to this problem, early stopping is a method of storing models in optimal epochs by terminating learning when the validation loss does not decrease further after a particular epoch [2]. The data should be independent and homogeneous for the valid implementation of early stopping. However, in a heterogeneous distributed network environment where learning data are scarcer than the data required for local processing, the data collected from a node are not independently and identically distributed (IID); therefore, the accuracy is low [3].

Numerous studies have been conducted to optimize the generalization capability of neural networks. The index data division (IDD) algorithm [4,5], a representative method, is a method of division according to the index distance rule,

Received 5 January 2023, Revised 9 May 2023, Accepted 10 August 2023

*Corresponding Author Il-Gu Lee (E-mail: iglee@sungshin.ac.kr)

Department of Convergence Security Engineering, Sungshin Women's University, 02844, Korea

Open Access <https://doi.org/10.56977/jicce.2023.21.3.198>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

wherein data are distributed. This method demonstrates high performance when the rules by which the data are split are known; however, in an environment where the quality of the learning data is non-homogeneous, the learning accuracy significantly degrades. The random data division (RDD) algorithm [6,7] is a prompt method for randomly segmenting data; however, it can degrade network performance. The block data division (BDD) algorithm [8,9] demonstrates superior performance to alternative methods because it randomly and evenly arranges data. However, it presents a significant difference in the learning precision depending on the data arrangement rules.

Recent studies have been conducted to optimize the generalized capabilities of neural networks using clustering techniques, including fuzzy c-means clustering (FMC) [11,12], center-based clustering [14], density-based clustering [15], and hierarchical clustering [16,17]. These methods improve the clustering precision of neural networks through silhouette analysis. However, these studies did not address the class imbalance problem because the accuracy varies depending on the homogeneity of the dataset. In addition, the original dataset can be freely modified, given that the data are balanced using an oversampling scheme.

The early stopping hyperparameters should be optimized by multiplying the weights with respect to the data-learning ratio to improve the accuracy of conventional FMC techniques. Therefore, this study proposes an early stopping algorithm for the application of an optimal patience hyperparameter. The main contributions of this study are as follows.

- The data homogeneity and learning accuracy were improved by applying an early stopping method based on an optimal patience hyperparameter.
- By applying FMC algorithms to classify the data, we mitigated the problems of network degradation and precision damage associated with the unrefined datasets of conventional methods.
- The proposed method achieved an accuracy of 99.7%, corresponding to a decrease in delay time by a factor of 2.33 and improvement in performance by a factor of 2.5 compared with the conventional method.

The remainder of this paper is organized as follows. Section II presents a comparison and analysis of related studies. Section III proposes a novel data measurement method that improves the conventional early stopping method. Section IV presents an evaluation of the latency, loss, and accuracy of the proposed method in comparison with the FMC and existing early stopping methods (IDD and RDD). Finally, Section V concludes the study.

II. RELATED WORK

This section presents an analysis of previous research and

the limitations of the data segmentation and clustering methods used for neural network learning. The latency and accuracy of each algorithm were measured in the same environment at 3,000 iterations and 30 epochs, respectively.

A. Data Segmentation

During the early stopping algorithm training process, the data segmentation scheme determines the precision of the neural network. In this case, the data division refers to the operation of dividing data into training, validation, and test sets. Table 1 presents an analysis of the conventional early stopping algorithm data division method.

Table 1. Previous research on the data segmentation method of the early stopping algorithm

Previous research	Feature	Limitation	Latency	Accuracy
IDD [4,5]	Splits data based on manually specified indices	<ul style="list-style-type: none"> • 262.74 s (low) of processing time • Difficult to quantify objective performance 	1.7 s	94.6%
RDD [6,7]	<ul style="list-style-type: none"> • Splits data based on automatically generated percentages • High performance for 10,000 datasets large datasets 	Overfitting occurs with less than	2.3 s	87.08%
BDD [8,9]	<ul style="list-style-type: none"> • Random data division algorithm and memory allocation in blocks • Reduced memory usage 	83.41% (low) accuracy for data arranged by specific rules	2.64 s	Depends on the status of the data array

The IDD algorithm improves segmentation performance by segmenting the data according to a manually specified index. Moreover, IDD demonstrates high performance when the data are segmented according to a particular segmentation rule; however, in most cases, quantifying the objective performance is challenging because the rules by which the datasets are sorted are not known. In addition, the speed is low because the user must manually specify an index.

The RDD algorithm divides data according to an automatically generated percentage. Although it is faster than IDD and BDD, its performance is high when the dataset is large because overfitting is highly likely to occur with less than 10,000 datasets.

The BDD algorithm divides data by allocating memory in blocks to an RDD algorithm, which can reduce memory usage compared with other algorithms. However, because only randomly specified training datasets are used for training, data arranged according to specific rules have a low accuracy of 83.41% [8,9].

As a result of experiments, the latency of IDD, RDD, and BDD were derived as 1.7, 2.3, and 2.64 s, respectively. The accuracies of IDD and RDD were 94.6 and 87.08%, respectively; however, for BDD, different values were derived depending on the state of arrangement of the data. Therefore, IDD and RDD were selected as control groups.

B. Clustering

Clustering is a type of unsupervised learning method based on the grouping of data with similar characteristics and is classified as soft clustering [10], hard clustering [13], and hierarchical clustering [15,16] (i.e., the clustering of data based on the probability of belonging to a cluster, depending on the distance between clusters). Table 2 presents an analysis of the related clustering studies.

The FMC proposed by J.C. Dunn is a soft clustering method, which is the most high-level clustering method; however, its precision varies with the homogeneity of the dataset because it does not solve the class imbalance problem arising during training. Furthermore, variations may occur in the original dataset during the training process because the data are balanced using an oversampling method.

Hard clustering is classified as center-based clustering [14], which allocates data on a central basis, and density-based clustering [15], which allocates data on a distance basis. Center-based clustering clusters the points closest to any cluster center. As the number of features increases, the precision of the method decreases. Furthermore, given that center-based clustering uses an iterative algorithm, with an increase in the number of iterations, the latency increases, in addition to the sensitivity of the outliers. Density-based clustering forms clusters by moving a cluster center to a dense data location. Herein, clustering can be achieved without setting the number of clusters; however, the performance is sig-

nificantly dependent on the cluster radius variable, and the latency increases.

Hierarchical clustering assumes each data point to be a cluster, calculates the distance between clusters, and generates and clusters similar data pairs with short distances into one cluster. Hierarchical clustering is limited because the delay time increases with the number of features.

To overcome the limitations of the algorithms presented above, this study proposes a novel early stopping algorithm based on the FMC algorithm with optimal clustering performance.

III. SYSTEM MODEL AND METHODS

This section describes the framework of the enhanced FMC (EFMC), which is an early stopping algorithm based on the FMC algorithm. Fig. 1 shows a structural diagram of the proposed algorithm. As shown in Fig. 1, the EFMC conducts training using the FMC for algorithms that perform data division using the optimal patience hyperparameter values.

A. Patience Hyperparameter

The patience hyperparameter stops training when the extent of the increase in performance does not match the increase in the number of epochs in the early stopping mechanism and exhibits a significant influence on neural network training [18].

Average precision (AP) is expressed in (1) [19], where r denotes *recall*, i denotes the number of points to be measured using 11-point interpolation, and $\max p(r)$ denotes the maximum precision measured at the recall value.

Table 2. Related work on clustering

Previous research	Feature	Limitation	Latency	Accuracy
Soft clustering (FMC) [10-12]	<ul style="list-style-type: none"> Represents the degree of membership as a probability Calculates weights by maximizing similarities within clusters and minimizing similarities between clusters 	<ul style="list-style-type: none"> Accuracy varies with respect to the homogeneity of the dataset Variations in the original dataset 	2.04 s	Depends on data homogeneity
Hard clustering [13]	Center-based clustering [14]	Clusters points closest to the cluster center	1.71 s	96.76%
	Density-based clustering [15]	Clusters by moving the center of a cluster to a dense data location	1.56 s	90.53%
Hierarchical clustering [16,17]	Clusters by calculating the distance between clusters and creating similar data pairs with short distances into one cluster	Increase in latency when the number of features increases	1.65 s	97.2%

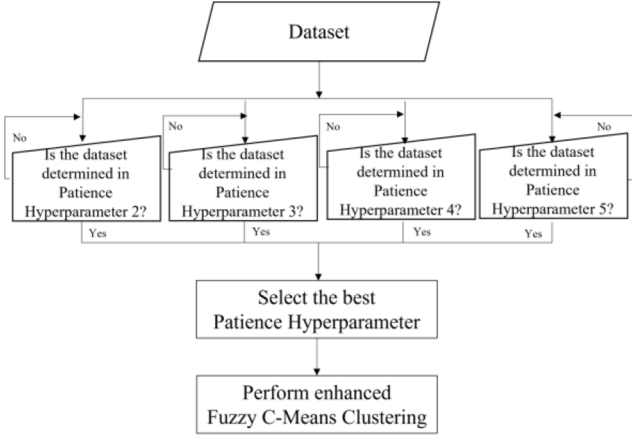


Fig. 1. Structure of EFMC.

$$AP = \sum_i (\eta_{i+1} + \eta_i) \max p(r) \quad (1)$$

The mean average decision (mAP) was derived by calculating the area below the AP graph in (1) and the average value for each class. As the initialization was performed randomly during training, each experiment was repeated five times to ensure accuracy. The optimal parameters were determined by comparing the mAP values when the patience hyperparameters were 2, 3, 4, and 5. Fig. 2 presents the test results of the average mAP values for the validation and test sets with respect to the patience hyperparameter values. The mAPs for the validation set with respect to patience hyperparameters 2-5 were 86.74, 89.07, 88.45, and 0, whereas those for the training set were 86.29, 88.32, 87.26, and 0, respectively. In an environment where the optimal hyperparameter varies depending on the dataset, the EMFC framework sets patience hyperparameter 3, which exhibits the optimal mAP value on the dataset used in the experiment, as the optimal value through the above process.

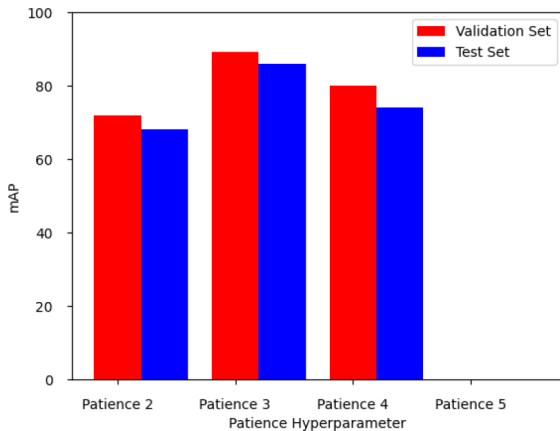


Fig. 2. mAP on the validation and test sets with respect to the patience hyperparameter.

B. Fuzzy C-Means Clustering

Conventional techniques are limited in that the learning performance is degraded when a non-homogeneous dataset is input because early stopping algorithms are not refined. To overcome this limitation, the data were sorted using FMC [20,21]. 7,860-character data and 6,000 32×32 image data were collected from the Modified National Institute of Standards and Technology (MNIST) and Canadian Institute for Advanced Research (CIFAR-10) datasets, respectively. Both datasets were divided into training, validation, and test sets in a ratio of 60:20:20 [22]. Table 3 lists the number of data samples used in the experiments.

Table 3. Experimental dataset

Dataset category	MNIST dataset	CIFAR-10 dataset
Training set	4,720	3,600
Validation set	1,570	1,200
Test set	1,570	1,200
Total	7,860	6,000

Data were sorted with respect to the distance from the cluster center, and the data were distributed with respect to the dataset division ratio. The proposed algorithm can improve precision because it uses an optimal algorithm to divide the data from the optimal patience hyperparameter value, and the homogeneity of the dataset has a slight influence because it is independent of the data distribution rule.

The homogeneity of the dataset was calculated as the average distance between the pointer and the center of the cluster used by the FMC. With an increase in the average distance, the homogeneity increases, and when N is set as the total number of pointers and D as the distance function between clusters, homogeneity H is calculated using (2) [23], where p_i denotes the i^{th} cluster pointer, and $C(p_i)$ denotes the center of p_i .

$$H = \frac{1}{N} \sum_i D(p_i, C(p_i)) \quad (2)$$

The operation process of the FMC-based early stopping algorithm is shown in Fig. 3 [24,25], where j is a dataset classified as $C(p)$, k is a dataset classified as cluster K , m is the average distance between p_i and p_{i+1} , w is a randomly assigned weight, d is the Euclidean distance from p to C , and C' is the center of cluster K .

The calculation formula of m_i for classifying p allocated by $C(p)$ is given in (3) [26], where $d(p_i)$ denotes the distance between the i^{th} pointers and $d(p_{i+1})$ denotes the distance between the $(i+1)^{\text{th}}$ pointers. Among p classified as $C(p)$, p satisfying $d(p_{ij}) > 2d$ is reclassified as cluster K , and the minimum value of $C'(p_{ik})$ in the process is determined using (4); here, $d(p_{ik})$ and $d(p_{ij})$ denote the Euclidean distances from p to cluster K and from p to $C(p)$, respectively. The

Lagrange multiplier method is used to derive the minimum value of $C'(p_{ik})$ [27].

$$m_i = \frac{1}{N} \sum_{i=1}^n \sqrt{|d(p_{i+1}) - d(p_i)|^2} \quad (3)$$

$$C'(p_{ik}) = \frac{1}{\sum_{i=1}^n \frac{d(p_{ik})^{w-1}}{d(p_{ij})}} \quad (4)$$

As shown in Fig. 3, EFMC operates in four stages. When the dataset is input, first, cluster initialization is performed, and p is assigned as $C(p)$ when $d(p_{ij}) > m(p_{ij})$. Subsequently, p allocated by $C(p)$ is grouped into cluster K when $d(p_{ij}) > 2d$. Based on p and C , the values of D_{ik} and D_{ij} are obtained, and w is randomly assigned to calculate $C'(p_{ik})$. In this case, if $C(p) > C'(p)$, the value changes to $C'(p)$, and the process is repeated until the $C'(p)$ value is minimized.

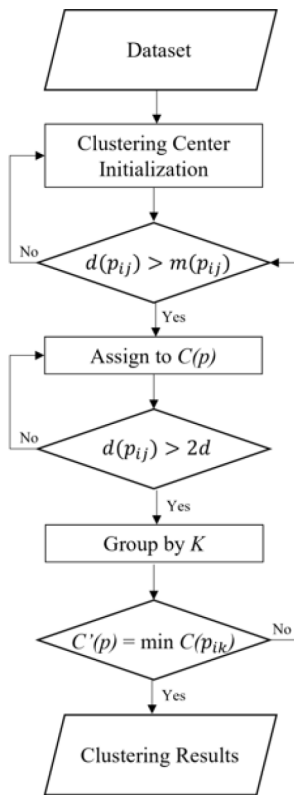


Fig. 3. Flowchart of EFMC.

IV. PERFORMANCE EVALUATION AND ANALYSIS

This section describes the evaluation environment for the proposed early stopping method and presents a comparison and analysis method for the speed and accuracy of the training and validation sets with respect to the IDD, RDD, and FMC.

A. Experimental Setup

The collected image data were preprocessed through one-hot encoding and then augmented using the ImageDataGenerator class in the TensorFlow Keras library. Augmentation means increasing the amount of data by adding noise while maintaining the nature of the original data.

The collected data were extracted and combined with pixel x of the same size from a randomly selected $C(p_i)$. Next, non-homogeneous data with a p -value of less than 0.05 were extracted using the Kolmogorov-Smirnov (KS) test. P -value is an indicator of the probability that the statistic obtained when assuming the null hypothesis is true [28], and the KS test evaluates the null hypothesis that the cumulative probability distribution (CDF) of the data matches the CDF of vector x . The KS test does not require a separate assumption of the shape or number of samples (nonparametric static test) and is suitable for the comparison of large samples. The experiment was conducted using the kstest function in MATLAB.

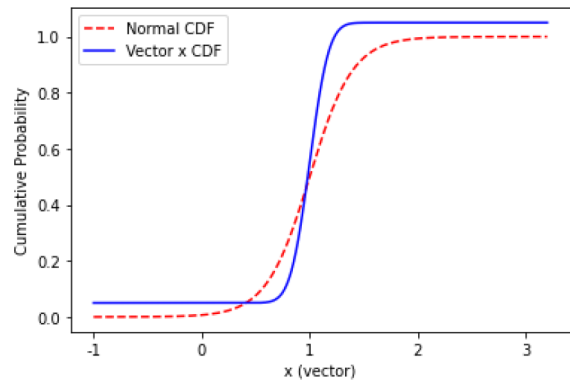


Fig. 4. KS test cumulative probability distribution plot for inhomogeneous datasets.

Fig. 4 shows a KS test cumulative probability distribution diagram for 7,860 nonuniform data samples extracted from 64,580 randomly selected MNIST data samples. In the graph, the x-axis represents the vector x , whereas the y-axis represents the cumulative probability. Evidently, the hypothesis (h) and test statistic (k) were 1 and 0.2346, respectively. The KS test algorithm compared the test statistic value k with the p -value to determine the validity of the null hypothesis; if k was smaller than the p -value, the null hypothesis was rejected. Herein, the KS test rejected the null hypothesis at the 5% significance level and adopted the alternative hypothesis; thus, the extracted 7,860 data samples were non-homogeneous.

TensorFlow Keras library was used to create convolutional neural networks (CNNs) and the patience hyperparameter, which was updated whenever learning was conducted using

the perception class. The model was optimized using cross-entropy and the Adam optimizer and was set to early stopping if the model remained unchanged more than three times.

Data clusters utilized the scikit-learn fuzzy-c-means package, and the accuracy was evaluated with respect to a logistic-regression class. Finally, it was visualized as a pyplot using the Matplotlib library from the Pandas package.

The hardware specifications for the experiment are presented in Table 4.

Table 4. Hardware specifications

Hardware	Specification
CPU	Intel(R) Core™ i7-1065G7 CPU @ 1.30 GHz
GPU	Intel(R) Iris(R) Plus Graphics
GPU Memory	7.9 GB
RAM	16 GB
SSD	477 GB

B. Evaluation Analysis

1) Latency

Fig. 5 presents a comparison graph of the delay times of the EFMC, IDD, RDD, and FMC with respect to the number of iterations and homogeneity of the MNIST character dataset. In the graph, the x-axis represents the homogeneity of the dataset, whereas the y-axis represents the iteration. Homogeneity is an index measured by the ratio of nonhomogeneous noise distributed in a dataset. The noise was measured by multiplying the maximum point of the loss value of the loss function by the epsilon parameter. The epsilon parameter is an index that denotes the degree of damage to the image data, and in the experiment, it was set in the range of 0-0.5. The number of epochs was set to 30, and the iterations and epochs were set as factors with the same meaning. The time required for one epoch was measured as 6 s.

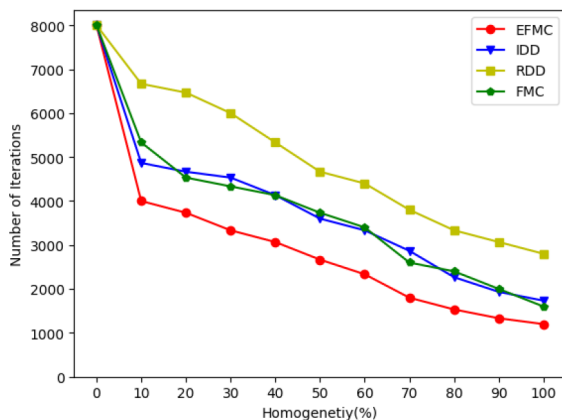


Fig. 5. Latency of EFMC, IDD, RDD, and FMC with respect to the number of iterations and homogeneity.

Latency and iterations were inversely proportional, and the latencies of the four models were the same when the homogeneity was 0%. However, when the homogeneity was 10% or higher, the latency of the EFMC was optimal. The latency of EFMC in the homogeneity range of 0-100% and iteration range of 0-8,000 improved by an average factor of 2.05 with a homogeneity of 50% and a factor of up to 2.33 over 3,000 iterations.

Table 5 compares the latencies of the EFMC, IDD, RDD, and FMC with respect to the homogeneity of CIFAR-10. The number of iterations was set to 4,000, and the other conditions were the same as those shown in Fig. 4.

As with the results on the MNIST character dataset, the homogeneity and latency were inversely proportional, and the latencies of the four models were the same when the homogeneity was 0%; however, EFMC had the best latency when the homogeneity was more than 10%. In the homogeneity range of 0-100%, the latency of EFMC improved by an average factor of 1.87 and factor of up to 2.12 with a homogeneity of 60%.

Table 5. Latency of EFMC, IDD, RDD, and FMC with respect to homogeneity

Homogeneity (%)	Latency (s)			
	EFMC	IDD	RDD	FMC
0	8	8	8	8
10	4.7	6.13	6.88	6.35
20	4.27	5.97	6.3	6.21
30	3.92	5.4	6.02	5.98
40	2.43	3.06	3.47	3.19
50	1.71	2.7	2.98	2.84
60	1.29	2.13	2.74	2.3
70	1.32	2.17	2.8	2.32
80	1.3	2.21	2.83	2.3
90	1.33	2.22	2.88	2.35
100	1.33	2.22	2.87	2.34

2) Accuracy

Fig. 6 presents a comparison graph of the loss values and accuracies of EFMC, IDD, RDD, and FMC on the MNIST character dataset. Fig. 6(a) presents the measurement results for the loss value with respect to the number of epochs, and Fig. 6(b) presents the evaluation results in terms of the accuracy with respect to the number of epochs. In this case, a complementary relationship between the loss value and the accuracy was established. As the number of epochs increased, the loss value decreased and accuracy improved.

With reference to the experimental results, the proposed EFMC loss value was lower than the others, and it was improved by an average factor of 2.2 compared with the other three models. At epoch 30, the loss improved by a factor of up to 2.71, essentially presenting the smallest devia-

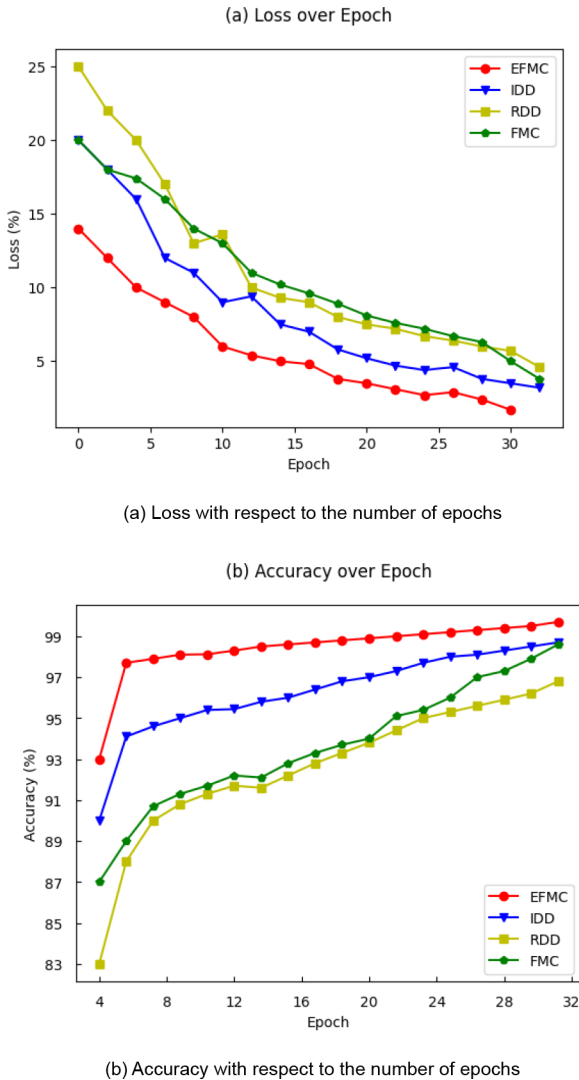


Fig. 6. Performance of EFMC, IDD, RDD, and FMC with respect to the epoch: (a) Loss and (b) Accuracy.

tion among the four models. In addition, EFMC demonstrated the highest accuracy and was improved by an average factor of 1.16. The accuracy improved by a factor of up to 1.24 at epoch 0, essentially presenting the smallest deviation among the four models.

Table 6 presents a comparison of the epoch accuracies of the CIFAR-10 image dataset. Similar to the results for the MNIST character dataset, the epoch and accuracy exhibited a proportional relationship, and EFMC achieved the best accuracy. At 0-30 epochs, EFMC improved by an average factor of 1.1 and achieved the highest accuracy of up to 98.3% at epoch 30.

Fig. 7 presents a distribution diagram comparing the accuracies of EFMC, IDD, RDD, and FMC with respect to the homogeneity of the dataset. Fig. 7(a) presents the result for

Table 6. Accuracy of EFMC, IDD, RDD, and FMC in terms of the number of epochs

Epoch	Accuracy (%) EFMC	Accuracy (%) IDD	Accuracy (%) RDD	Accuracy (%) FMC
0	91.72	87	78.86	84.98
5	96.48	89.09	80.77	86.6
10	97.86	90.7	81.4	87.55
15	97.9	92.28	82	88.71
20	98.04	92.66	85.95	91.24
25	98.21	93.67	86.08	93.3
30	98.3	93.8	86.54	93.63

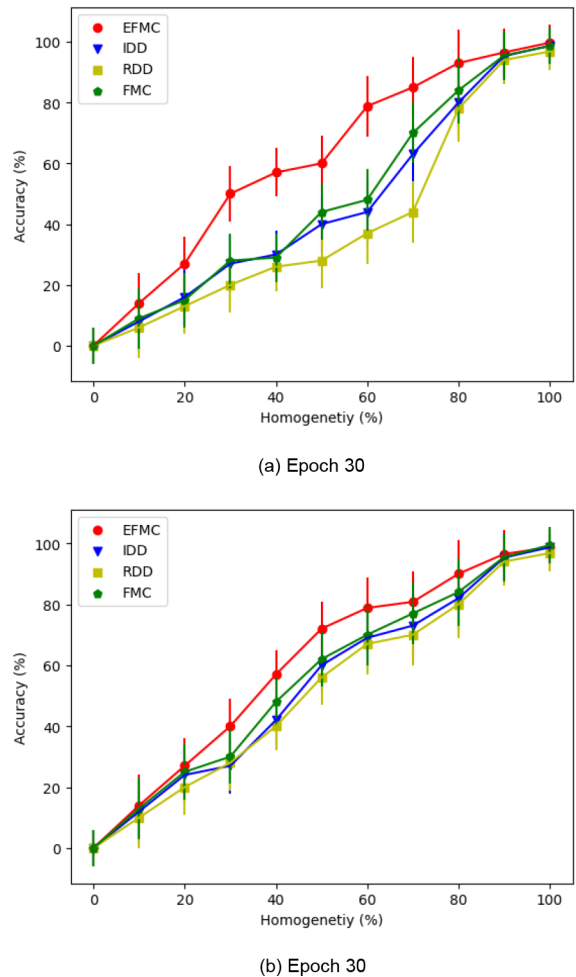


Fig. 7. Accuracy of EFMC, IDD, RDD, and FMC with respect to homogeneity: (a) Epoch 30 and (b) Epoch 150.

epoch 30, whereas Fig. 7(b) presents the result for epoch 150. In this experiment, the accuracy was measured using the coefficient of determination (R^2) [29], which indicates the accuracy of an independent variable representing the suitability of a dependent variable in a regression model. According to (5), R^2 is calculated by dividing the sum of the

squared deviations of each data point by the sum of the squared deviations. As the value converges to 1, the error decreases, and the accuracy increases. In (5), p_i denotes the predicted value, a_i denotes the actual value, and \bar{a}_i denotes the average value.

$$R^2 = \frac{\sum (p_i - \bar{a}_i)^2}{\sum (a_i - \bar{a}_i)^2} \quad (5)$$

The R^2 values of the four models were similar when the homogeneity was 0%, as can be observed in both graphs; however, the R^2 value of EFMC was optimal when the homogeneity was 10% or greater. In the homogeneity range of 0-100% at epoch 30, the accuracy of EFMC improved by an average factor of 1.52 and by a maximum factor of 2.5 when the homogeneity was 40%. The accuracy of EFMC at epoch 150 improved by an average factor of 1.14 compared with the other three models, and by a maximum factor of 1.47 when the homogeneity was 50%. The performance of the proposed EFMC exhibited an increase in the average accuracy by factors of 1.33 and 1.7, even with a smaller number of epochs, and the increase in accuracy was larger than that of other comparative models.

The EFMC algorithm applied using the proposed early stopping method exhibited optimal average clustering performance because it improved the delay time, loss value, and accuracy by factors of 2.33, 2.71, and 2.5, respectively, compared with conventional models.

V. CONCLUSIONS

This study proposed an early stopping algorithm based on the FMC algorithm. Herein, the performance of the proposed algorithm was analyzed. Conventional techniques such as IDD, BDD, and RDD exhibit low accuracy because they are trained with rough datasets, and the network performance is degraded. FMC, center-based clustering, density-based clustering, and hierarchical clustering vary in precision depending on the homogeneity of the dataset and are limited with respect to the transformation of the original dataset during training. Herein, the accuracy was improved by dividing the data by the optimal patience hyperparameter value derived from the neural networks. Based on this, the early stopping algorithm was improved such that the homogeneity of the dataset was unaffected by the FMC algorithm. The experimental results revealed that the proposed technique demonstrated improved clustering performance with respect to latency, loss value, and accuracy by factors of 2.33, 2.71, and 2.5, respectively, compared with the conventional method with fewer epochs.

The proposed EFMC heuristically sets the patience hyperparameter in advance during data preprocessing according to the dataset. Although accuracy and homogeneity are guaran-

teed, it has a structural limitation that slows down as the types of datasets become more diverse. Future research will focus on increasing efficiency by establishing an experimental environment considering various heterogeneous nodes and by automatically setting an optimal patience hyperparameter.

ACKNOWLEDGEMENTS

This work was partially supported by the Korean Government (MOTIE) (P0008703, the Competency Development Program for Industry Specialist), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

REFERENCES

- [1] C. Corneanu, M. Madai, S. Escalera, and A. Martinez, "Explainable early stopping for action unit recognition," in *IEEE 15th International Conference on Automatic Face and Gesture Recognition (FG 2020)*, Buenos Aires, Argentina, pp. 693-699, 2020. DOI: 10.1109/FG47880.2020.00080.
- [2] F. Lauer and G. Bloch, "Ho-Kashyap classifier with early stopping for regularization," *Pattern Recognition Letters*, vol. 27, no. 9, pp. 1037-1044, Jul. 2006. DOI: 10.1016/j.patrec.2005.12.009.
- [3] S.-E. Jeon, S.-J. Lee, and I.-G. Lee, "Hybrid in-network computing and large-scale data processing," *Elsevier Computer Networks*, vol. 226, p. 109686, May 2023. DOI: 10.1016/j.comnet.2023.109686.
- [4] H. Demuth, M. Beale, and M. Hagan, "Neural network toolbox for use with MATLAB user's guide," in *The MathWorks Inc*, 6th ed. Natick, MA: 2008.
- [5] G. Manogaran, P. M. Shakeel, S. Baskar, C. -H. Hsu, S. N. Kadry, R. Sundarasekar, P. M. Kumar, and B. A. Muthu, "FDM: Fuzzy-optimized data management technique for improving big data analytics," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, pp. 177-185, Jan. 2021. DOI: 10.1109/TFUZZ.2020.3016346.
- [6] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural networks*, vol. 1, no. 4, pp. 761-767, Jun. 1998. DOI: 10.1016/S0893-6080(98)00010-0.
- [7] M. Elkano, J. A. Sanz, E. Barrenechea, H. Bustince, and M. Galar, "CFM-BD: A distributed rule induction algorithm for building compact fuzzy models in big data classification problems," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 1, pp. 163-177, Jan. 2020. DOI: 10.1109/TFUZZ.2019.2900856.
- [8] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," in *Proceedings of the 1997 International Joint Conference on Neural Networks*, Houston, USA, pp. 1930-1935, 1997. DOI: 10.1109/ICNN.1997.614194.
- [9] J. Gao, C. Huo, Y. Zhen, and G. Zhang, "Study on block and hierarchical division control of power internet of things," in *IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, pp. 18-21, 2020. DOI: 10.1109/ICEIEC49280.2020.9152320.
- [10] H. Shi, J. Yan, M. Ding, T. Gao, S. Li, Z. Zhang, and Z. Li, "An

- improved fuzzy c-means soft clustering based on density peak for wind power forecasting data processing,” in *Asia Energy and Electrical Engineering Symposium*, Chengdu, China, pp. 801-804, 2020. DOI: 10.1109/AEES48850.2020.9121374.
- [11] J. C. Bezdek and J. C. Dunn, “Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions,” *IEEE Transactions on Computers*, vol. C-24, no. 8, pp. 835-838, Aug. 1975. DOI: 10.1109/T-C.1975.224317.
- [12] C. R. Mardiantien, I. Atastina, and I. Asror, “Product segmentation based on sales transaction data using agglomerative hierarchical clustering and FMC model,” in *IEEE 3rd International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, pp. 280-285, 2020. DOI: 10.1109/ICOIACT50329.2020.9332023.
- [13] M. Ahmed and A. Barkat, “Performance analysis of hard clustering techniques for big iot data analytics,” in *Cybersecurity and Cyberforensics Conference (CCC)*, Melbourne, Australia, pp. 62-66, 2019. DOI: 10.1109/CCC.2019.000-8.
- [14] Y. Rong and Y. Liu, “Staged text clustering algorithm based on K-means and hierarchical agglomeration clustering,” in *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, pp. 124-127, 2020. DOI: 10.1109/ICAICA50127.2020.9182394.
- [15] A. Bechini, F. Marcelloni, and A. Renda, “TSF-DBSCAN: A novel fuzzy density-based approach for clustering unbounded data streams,” *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 3, pp. 623-637, Mar. 2022. DOI: 10.1109/TFUZZ.2020.3042645.
- [16] J. Leskovec, A. Rajaraman, and J. Ullman. “Mining of massive datasets,” in *Cambridge University Press*, ch. 7, 2020.
- [17] J. Gao, C. Huo, Y. Zhen, and G. Zhang, “Study on block and hierarchical division control of power internet of things,” in *IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, pp. 18-21, 2020. DOI: 10.1109/ICEIEC49280.2020.9152320.
- [18] X. Ying, “An overview of overfitting and its solutions,” *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022022, Feb. 2019. DOI: 10.1088/1742-6596/1168/2/022022.
- [19] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. B. da Silva, “A comparative analysis of object detection metrics with a companion open-source toolkit,” *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021. DOI: 10.3390/electronics10030279.
- [20] W. Lixin, T. Xuejing, W. Hongrui, and S. Yang, “Identification method of fuzzy inference system based on improved fuzzy clustering arithmetic,” *Control and Decision*, vol. 22, pp. 77-79, 2008.
- [21] J. C. Bezdek, “Pattern recognition with fuzzy objective function algorithms,” *Advanced Applications in Pattern Recognition (AAPR)*, 1981.
- [22] M. S. Salekin, A. B. Jelodar, and R. Kushol, “Cooking state recognition from images using inception architecture,” in *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh, pp. 163-168, 2019. DOI: 10.1109/ICREST.2019.8644262.
- [23] N. Kang, J. Kang, and H. -S. Yong, “Performance comparison of clustering techniques for spatio-temporal data,” *Journal of the Korea Intelligent Information Systems*, vol. 10, no. 2, pp. 15-37, 2004.
- [24] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203, 1984. DOI: 10.1016/0098-3004(84)90020-7.
- [25] S. Huang, H. Dang, R. Jiang, Y. Hao, C. Xue, and W. Gu, “Multi-layer hybrid fuzzy classification based on SVM and improved PSO for speech emotion recognition,” *Electronics*, vol. 10, no. 23, p. 2891, Nov. 2021. DOI: 10.3390/electronics10232891.
- [26] H. Fang, J. G. Huang, and F. H. Chu. “Grey relation evaluation model of weapon system based on rough set,” *Acta Armamentarii*, vol. 29, no. 2, pp. 253-256, 2008.
- [27] A. L. De and C. A. Guo, “An image segmentation method based on the fusion of vector quantization and edge detection with applications to medical image processing,” *International Journal of Machine Learning and Cybernetics*, vol. 5, pp. 543-551, Aug. 2014. DOI: 10.1007/s13042-013-0205-1.
- [28] D. S. Dimitrova, V. K. Kaishev, and S. Tan, “Computing the kolmogorov-smirnov distribution when the underlying cdf is purely discrete, mixed or continuous,” *Journal of Statistical Software*, vol. 95, no. 10, pp. 1-42, Oct. 2020. DOI: 10.18637/jss.v095.i10.
- [29] E. W. Weisstein. Correlation Coefficient. MathWorld--A Wolfram Web Resource. [Online] Available: <https://mathworld.wolfram.com/CorrelationCoefficient.html>.



Chae-Rim Han

is a researcher at the CSE Lab of Convergence Security Engineering, Sungshin Women's University, Seoul, Korea. Her research interests include convergence security and deep reinforcement learning.



Sun-Jin Lee

received her B.S. degree in convergence security engineering at Sungshin Women's University in 2021, and her M.S. degree in the department for future convergence technology engineering at Sungshin Women's University in 2023. She is currently pursuing a Ph.D. degree in future convergence technology engineering at Sungshin Women's University, Seoul, Korea. Her research interests include wireless networks, endpoint security, and convergence security using deep learning.



Il-Gu Lee

received his B.S. Degree in electrical engineering at Sogang University, Seoul, Korea, in 2003, and M.S. degree from the department of information and communications engineering at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2005. He received his Ph.D. degree from the Graduate School of Information Security in computer science & engineering, KAIST, in 2016. He is currently a Professor at the Department of Convergence Security Engineering, Sungshin Women's University (SWU), Seoul, Korea. Before joining SWU in March 2017, he was a Senior Researcher at the Electronics and Telecommunications Research Institute (ETRI) during 2005-2017 and served as a Principal Architect and Project Leader for Newratek (KR) and Newracom (US) during 2014-2017. His research interests are wireless/mobile networks with an emphasis on information security, networks, wireless circuits, and systems. He has authored/coauthored more than 160 technical papers in the areas of information security, wireless networks, and communications, and holds approximately 160 patents. In addition, he is an active participant in and contributor to the IEEE 802.11 WLAN standardization committee.