The object of the study is the classification process of imbalanced multi-class data in air quality analysis based on the air pollution standard index (ISPU), which involves numerical environmental features and categorical output classes.

This study addresses the problem of imbalanced multi-class classification in air quality data based on the air pollution standard index (ISPU), where conventional classification techniques tend to be biased toward majority classes and fail to accurately identify minority classes. To overcome this limitation, a cost-sensitive learning strategy combined with adaptive error weighting, grid search with k-fold cross-validation, and principal component analysis (PCA) is applied. The dataset consists of 1,147 samples with an imbalanced distribution across three classes. The results demonstrate that the proposed method achieves an accuracy and F1-score of 98.55% and an area under the ROC curve (AUC) of 0.999, while significantly improving minority class sensitivity. This performance is explained by the cost-sensitive method that increases the penalty for minority class errors and by PCA, which enhances feature representation and learning stability. Compared to existing methods, the proposed method provides a more balanced and robust classification performance without modifying the original data distribution. This method can be effectively applied to ISPU-based air quality classification results and other imbalanced multi-class classification problems, although it requires careful parameter optimization for different data characteristics

Keywords: cost-sensitive, artificial neural network, imbalanced classification, multi-class classification, PCA, machine learning

UDC 004.032.26:004.8
DOI: 10.15587/1729-4061.2026.356119

DEVELOPMENT OF COST-SENSITIVE ARTIFICIAL NEURAL NETWORK OPTIMIZATION IN SOLVING IMBALANCED MULTI-CLASS CLASSIFICATION PROBLEMS

Bosker Sinaga
Corresponding author
Bachelor of Computer Science, Master of Computer Science, PhD Student*
E-mail: boskersinaga@gmail.com
ORCID: https://orcid.org/0000-0001-9664-3307
Yuhandri
Bachelor of Computer Science, Master of Computer Science, Doctor, Professor*
ORCID: https://orcid.org/0000-0002-8576-5488
Gunadi Widi Nurcahyo
Bachelor of Computer Science, Master of Computer Science, Doctor, Associate Professor*
ORCID: https://orcid.org/0000-0003-0714-0244
*Department of Information Technology
Universitas Putra Indonesia "YPTK"
Raya Lubuk Begalung str., Lubuk Begalung Nan XX, Kec. Lubuk Begalung, Kota Padang, Sumatera Barat, Indonesia, 25145

Received 28.01.2026
Received in revised form 04.04.2026
Accepted 16.04.2026
Published 30.04.2026

How to Cite: *Sinaga, B., Yuhandri, Nurcahyo, G. W. (2026). Development of cost-sensitive artificial neural network optimization in solving imbalanced multi-class classification problems. Eastern-European Journal of Enterprise Technologies, 2 (9 (140)), 40–63. https://doi.org/10.15587/1729-4061.2026.356119*

1. Introduction

Advances in digital technology and environmental monitoring systems have generated large amounts of air quality data with increasingly complex structures [1]. Data sourced from the environmental, health, and urban sectors is not only growing, but is also multidimensional and influenced by various interacting variables [1]. In the context of air pollution, air pollution index (API) data reflects the dynamics of interactions between meteorological factors, human activities, and environmental conditions that change spatially and temporally. This complexity requires an adaptive and accurate analytical method to produce a reliable classification system and support data-driven decision-making in air quality management [2].

Classification is one of the approaches commonly used to analyze air quality data and categorize pollution levels based on specific standards. [3]. This method works by mapping data into several classes that represent air quality conditions, and has been widely applied in various environmental and public health monitoring systems [2]. In practice, air pollution standard index (ISPU) classification plays an important role in providing early warnings, evaluating health risks, and formulating environmental policies [4]. However, conventional classification methods often assume a balanced data distribution, making them less capable of handling class imbalance issues that commonly occur in ISPU data, especially when classes with extreme pollution levels have far fewer samples [4].

The main challenge in modern air quality data classification is how to improve the sensitivity of the method to minority classes without sacrificing the performance of majority classes [5, *6]*. This imbalance in data distribution can cause method to be biased and produce predictions that are

unfair and inaccurate [7, 8]. To overcome this problem, a new method is needed that not only focuses on global accuracy, but also considers the cost of classification errors between classes [9, 10]. A cost-sensitive method with the integration of adaptive error weighting and dimension reduction methods is expected to transform the learning process into a more stable, representative, and fair one. Thus, the classification results can better reflect the actual air pollution conditions and contribute significantly to improving the air quality prediction system and more effective environmental decision-making [11].

Therefore, study focused on developing a cost-sensitive artificial neural network method integrated with adaptive error weighting and principal component analysis (PCA) is highly relevant, as it can provide a robust solution for handling imbalanced multi-class classification by enhancing the sensitivity of the minority class, reducing the bias of the majority class, and improving the method's accuracy and generalization ability.

2. Literature review and problem statement

The paper [12] proposes constrained optimization on deep neural networks (DNNs) to improve minority class accuracy without compromising majority class performance. This method effectively reduces bias in imbalanced data, but is still limited to binary classification and has not been tested on multi-class problems with complex levels of imbalance. The paper [13] examines unbalanced multi-class classification on a big data scale through distributed resampling using Apache Spark. Although capable of improving performance on large datasets, this method is highly dependent on data manipulation and has the potential to cause overfitting and high computational costs.

The paper [14] applies cost-sensitive learning to convolutional neural networks (CNNs) for anomaly detection in highly imbalanced ECG data and showing a significant improvement in accuracy. However, this method is specifically designed for medical time-series data, so its generalization to multidimensional tabular data is still limited. The paper [15] combines SMOTE and CNN to address class imbalance in various medical datasets. Although it improves the distribution of minority classes, this method still risks reducing generalization ability due to its dependence on synthetic oversampling.

The paper [16] develops a cost-sensitive neural network-based ensemble method combined with metaheuristic optimization successfully improved the g-mean and F1-score metrics. However, the high complexity of the method and optimization process limits its application to large datasets and real-time systems. The paper [17] emphasizes the role of feature selection through a combination of information gain, FFT, and SMOTE to improve random forest performance. Although effective, this method still relies on resampling techniques and does not directly integrate cost-sensitive learning into the method's loss function.

The paper [18] applies cost-sensitive neural networks for gastrointestinal disease classification and demonstrates improved method accuracy and interpretability. However, this study is limited to a single domain and a relatively small number of classes. The paper [19] develops a cost-sensitive CNN for cervical cancer classification with the support of explainable AI. Although it increases confidence in prediction results, this method has not been equipped with an adaptive cost parameter optimization method. The paper [20] proposes adaptive cost-sensitive learning (AdaCSL) to minimize the distribution difference between training data and test data and demonstrates better performance than static approaches. However, this method requires complex distribution estimation and increases computational costs, making its application to large-scale multi-class classification still challenging. [1] shows that the use of balanced class weights can improve the performance of ANNs in imbalanced multi-class classification. This study confirms the effectiveness of cost-sensitive loss in ANNs. However, the class weights used are still static.

Based on the overall review, it can be concluded that previous studies still face limitations in the form of a dominant focus on binary classification, dependence on resampling techniques that risk overfitting, and the use of static and less adaptive cost parameters. These limitations prevent the method from optimally handling imbalanced multi-class classification problems.

The main problem lies in the limitations of conventional classification methods, particularly artificial neural networks (ANN), in handling imbalanced multi-class data, which leads to bias toward the majority class and reduced performance for minority classes. Additionally, high-dimensional data, such as ISPU, often introduces feature redundancy that can negatively affect learning stability and efficiency. Based on these limitations, there remains a need for a method that is capable of simultaneously addressing class imbalance, improving minority class sensitivity, and enhancing feature representation while maintaining stable and efficient learning.

3. The aim and objectives of the study

The aim of this study is to improve classification performance in imbalanced multi-class problems by enhancing minority class sensitivity, reducing bias toward majority classes, and achieving balanced and reliable classification outcomes. This will make it possible to obtain a more balanced and accurate classification that better reflects real data conditions, particularly in air quality analysis based on ISPU.

To achieve this aim, the following objectives were accomplished

– to develop a cost-sensitive learning method that adjusts classification error weights according to class distribution;

– to implementation of the proposed cost-sensitive artificial neural network (COST-ANN) method;

– to integrate principal component analysis (PCA) to enhance feature representation and improve learning stability in high dimensional data;

– to evaluate classification performance using metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).

4. Material and methods

4. 1. The object and hypothesis of the study

The object of the study is the classification process of imbalanced multi-class data in air quality analysis based

on the air pollution standard index (ISPU), which involves numerical environmental features and categorical output classes.

The main hypothesis of the study is that the application of a cost-sensitive learning method combined with adaptive error weighting and dimensionality reduction can improve classification performance by increasing minority class sensitivity while maintaining overall accuracy and stability.

The assumptions made in the study are as follows:

– the dataset is representative of real air quality conditions and reflects the actual class imbalance distribution;

– the selected features are relevant and sufficient to represent the classification problem;

– the training and testing data are drawn from the same distribution;

– the evaluation metrics used are adequate to assess classification performance.

The simplifications adopted in the study are as follows:

– the classification problem is limited to tabular numerical data without considering temporal or spatial dependencies;

– the method architecture is fixed during the training process;

– parameter optimization is performed using Grid Search within a predefined range;

– external factors affecting air quality are not explicitly modeled beyond the available dataset features.

4. 2. Dataset

The study dataset consists of air pollution standard index (ISPU) data used as the basis for method development and evaluation, with the data sample presented in Table 1.

The ISPU dataset consists of 1,147 data points with 8 numerical attributes and 3 output classes (good, moderate, unhealthy) in the environmental domain. The class distribution is unbalanced, with the moderate class as the majority (937 data), followed by unhealthy (133) and good (77), as reflected in the IR value of 12.17 and ID value of 8.22. This condition indicates a high level of imbalance and has the potential to cause bias in conventional classification methods.

Table 1

Sample dataset

No.	A1	A2	A3	A4	A5	A6	A7	A8	Class
1	46	65	46	8	9	38	65	2	Moderate
2	37	55	47	7	11	28	55	2	Moderate
3	43	62	50	7	15	14	62	2	Moderate
4	40	52	48	8	12	10	52	2	Moderate
5	40	62	48	12	12	18	62	2	Moderate
6	33	44	47	14	12	24	47	3	Good
7	58	46	48	22	12	33	58	1	Moderate
8	37	47	7	26	17	17	47	2	Good
9	63	96	11	26	27	52	96	2	Moderate
10	67	97	14	28	28	62	97	2	Moderate
...	...	...	...	...	...	...	...	...	...
1143	65	116	36	22	58	18	116	2	Unhealthy
1144	50	87	36	20	46	17	87	2	Moderate
1145	59	107	37	20	62	16	107	2	Unhealthy
1146	65	127	36	23	47	21	127	2	Unhealthy
1147	42	80	35	18	41	13	80	2	Moderate

4. 3. Steps of the study

This study proposes a cost-sensitive artificial neural network (COST-ANN) method to address class imbalance issues in air pollutant index classification, with the study flow shown in Fig. 1.

Fig. 1 demonstrates the study methodology flow for developing and evaluating the COST-ANN method for ISPU classification, which includes data preprocessing (Z-Score normalization, PCA, and data partitioning), cost-sensitive method formulation, training and testing, and performance evaluation using classification metrics and visualization of results comparison.

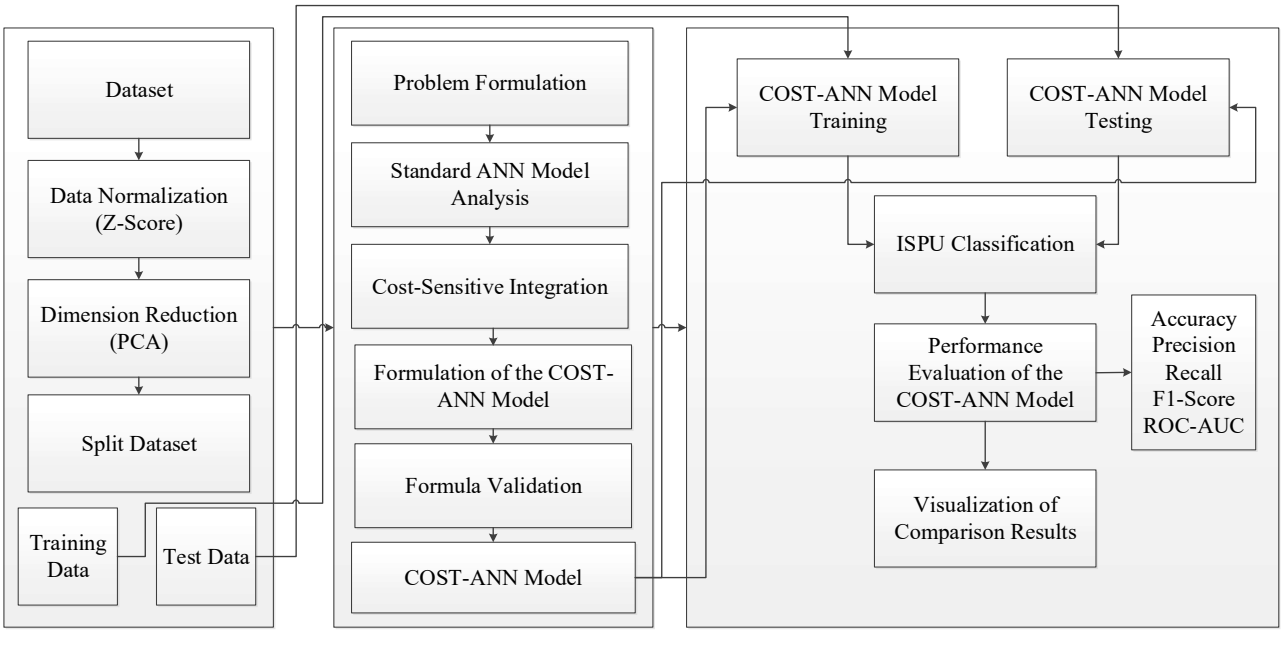

Fig. 1. Study flow chart

4. 4. Data transformation

The data transformation stage was carried out using Z-Score to standardize all numerical variables in the ISPU dataset, so that differences in scale between features could be eliminated and the neural network learning process became more stable and balanced [21]

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad (1)$$

x_{ij} expresses the value of feature j in observation i, μ_j – the mean value of feature j, and σ_j – the standard deviation of feature j. Through this transformation, each numerical variable is transformed so that it has a mean of zero and a standard deviation of one [22].

4. 5. Dimension reduction

Dimension reduction in this study was performed using principal component analysis (PCA) to transform high-dimensional data into a lower-dimensional space while retaining key information. Let's suppose that the Z-Score transformation data is expressed as a matrix $X \in R^{N\times m}$, where N denotes the number of observations and $m$number of input variables, and the first step in PCA is to centralize the data [23]

$$X_c = X - \mu, \quad (2)$$

where μ – the mean vector of each variable. Next, the covariance matrix is formed

$$C = \frac{1}{N-1} X_c^{\top} X_c. \quad (3)$$

The covariance matrix is then decomposed to obtain pairs of eigenvalues and eigenvectors

$$Cv_k = \lambda_k v_k, \quad (4)$$

where λ_k declare the eigen value to-k and v_k declare the corresponding eigenvectors. Eigenvectors are arranged based on the largest eigenvalues and selected as many as r main components, thereby forming a transformation matrix

$$W = \left[v_1, v_2, \ldots, v_r \right]. \quad (5)$$

Data projection into a lower-dimensional space is performed by

$$Y = X_c W, \quad (6)$$

where $Y \in R^{N\times r}$ – a representation of reduced dimension data. The application of PCA on the ISPU dataset aims to reduce data complexity, eliminate correlations between variables, and improve the efficiency and stability of the learning process in the subsequent multi-class classification methods stage.

4. 6. Artificial neural network (ANN)

Mathematically, ANN represents a mapping function from a multidimensional feature space n to the output space through a nonlinear function composition, expressed as

$$\hat{y}_i = f\left(x_i; \theta\right), \quad (7)$$

where $x_i \in R^n$ – the input feature vector, y_i – the classification result of the method, and θ represents all neural network parameters consisting of weights and biases in each layer. In multi-class classification problems, ANN generates class probabilities as output through the softmax activation function. The classification probability for class k is formulated as

$$\hat{y}_{i,k} = \frac{\exp\left(z_{i,k}\right)}{\sum_{j=1}^{K} \exp\left(z_{i,j}\right)}, \quad (8)$$

with $z_{i,k}$ expresses the linear activation of the output neuron for class k, where K – the number of classes. This formulation ensures that the ANN output value is within the range [0, 1] and the sum of all probabilities equals one. To measure the error between the actual label and the predicted result, standard ANNs use the categorical cross-entropy loss function, defined as

$$L_{ANN} = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_{i,k} \log\left(\hat{y}_{i,k}\right), \quad (9)$$

where $y_{i,k}$ – a one-hot representation of the actual class label. This loss function calculates the average prediction error across all data and all classes involved. The loss function formulation assumes that each classification error has the same weight, without considering the distribution of data amounts in each class. Implicitly, the error contribution of each data-class pair is considered identical in the network parameter optimization process θ [24].

4. 7. Multi-class classification problems

Multi-class classification is a supervised learning problem that aims to map each data object into one of several predetermined classes. Given a training dataset

$$D = \left\{ \left(x_i, y_i \right) \right\}_{i=1}^{m}, \quad (10)$$

with $x_i \in R^n$ as a dimensional feature vector n and $y_i \in \{1, 2, \ldots, K\}$ as a class label, where K states the number of classes involved in the classification process. Each pair (x_i, y_i) represents an observation used to study the relationship between features and classes. The main objective of multi-class classification is to determine a mapping function

$$f : \mathbb{R}^n \to \left\{ 1, 2, \ldots, K \right\}, \quad (11)$$

which generates class predictions $\hat{y}_i = f(x_i)$ so that prediction errors across all data can be minimized. This process is generally formulated as an optimization problem for a loss function L, that is

$$\min_f \frac{1}{m} \sum_{i=1}^{m} L\left(y_i, \hat{y}_i \right), \quad (12)$$

which measures the degree of mismatch between the actual label and the predicted result. In the context of real data, the distribution of data amounts in each class is often uneven. Mathematically, class imbalance can be expressed as

$$D_k \left| D_k \right| \neq \left| D_j \right|, \text{for } k \neq j, \quad (13)$$

with $|D_k|$ denotes the number of samples in class-k. This imbalance causes the contribution of errors in classes with

small amounts of data to be relatively lower to the overall loss function value. As a result, the mapping function $f(.)$obtained through the optimization process tends to better represent the majority class, while the method's ability to recognize minority classes becomes limited [25].

4. 8. Hardware and software environment

The experiments in this study were conducted using a computer with an Intel Core i7 processor, 16 GB RAM, and Windows 11 operating system. The method was implemented using Python programming language with libraries including NumPy, Pandas, Scikit-learn, and TensorFlow/Keras. The experiments were executed in the Google Colab environment to support efficient computation and method training.

5. Development of cost-sensitive artificial neural network (COST-ANN)

5. 1. Cost-sensitive artificial neural network method optimization

The proposed cost-sensitive artificial neural network method is formulated by extending the standard artificial neural network objective function for classification tasks, as described in [4], through the incorporation of class-dependent penalty weights to handle imbalanced data. This formulation enables the method to assign higher penalties to misclassification of minority classes, thereby improving classification balance [26]. Using the sigmoid activation function, the output probability is defined as

$$\hat{y}_i = P\left(y_i = 1 \mid x_i\right), \tag{14}$$

where x_i declare the feature vector on sample i, y_i – the actual label, and $\hat{y}_i$ – the predicted probability that the sample i belongs to the positive class. Standard binary cross-entropy loss function [27] formulated as

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log\left(\hat{y}_i\right) + \left(1-y_i\right)\log\left(1-\hat{y}_i\right)\right], \tag{15}$$

with m state the total number of samples, $y_i \in \{0,1\}$ as an actual label, and $\hat{y}_i$ as the prediction probability. This formulation is derived from the negative log-likelihood of the Bernoulli distribution and assumes that errors in both classes have equal contributions [28]. To overcome these limitations in unbalanced data, penalty weights are introduced to each class so that the loss function evolves into cost-sensitive binary cross-entropy

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left[C_1\, y_i \log\left(\hat{y}_i\right) + C_0\left(1-y_i\right)\log\left(1-\hat{y}_i\right)\right], \tag{16}$$

where C_1 – the penalty weight for the positive class (which is generally the minority class) and C_0 – the penalty weight for the negative class (majority class). When the minority class is underrepresented, the value C_1 set higher to increase the penalty for errors in that class. In order for this formulation to be extended to multi-class classification, binary labels are represented using one-hot encoding [29]

$$y_i = \left[y_{i,1}, y_{i,2}\right],\ y_{i,k} \in \{0,1\},\ \sum_{k=1}^{2} y_{i,k} = 1. \tag{17}$$

The prediction probability is expressed as

$$\hat{y}_i = \left[\hat{y}_{i,1}, \hat{y}_{i,2}\right],$$

with $\hat{y}_{i,1} = \hat{y}_i$ and $\hat{y}_{i,2} = 1 - \hat{y}_i$ [30]. With this substitution, the cost-sensitive binary loss function can be rewritten as [31]

$$L = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{2} c_k\, y_{i,k} \log\left(\hat{y}_{i,k}\right), \tag{18}$$

where c_k declares the penalty weight for class-k. This formulation shows that loss cost-sensitive binary is a special form of weighted cross-entropy based on vector classes. The formulation is then generalized to multi-class classification with K class. Under these conditions, the ANN produces activation scores $z_{i,k}$ for each class-k, which is then normalized using the softmax function [32]

$$\hat{y}_{i,k} = \frac{e^{z_{i,k}}}{\sum_{j=1}^{K} e^{z_{i,j}}}. \tag{19}$$

Actual labels are represented using one-hot encoding [33]

$$y_i = \left[y_{i,1}, y_{i,2}, \ldots, y_{i,K}\right],\ \sum_{k=1}^{K} y_{i,k} = 1. \tag{20}$$

The standard categorical cross-entropy loss function is formulated as [34]

$$L = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_{i,k} \log\left(\hat{y}_{i,k}\right), \tag{21}$$

which was then modified into the COST-ANN loss function by incorporating class penalty weights

$$L = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} c_k\, y_{i,k} \log\left(\hat{y}_{i,k}\right), \tag{22}$$

where c_k declares the penalty weight for class-k. Value c_k determined adaptively and proportionally based on data distribution, taking into account the proportion of samples in each class and the level of global imbalance.

Several methods have been proposed to determine the class weight c_k in handling class imbalance. The inverse class frequency method calculates class weights inversely proportional to the number of samples, namely $c_k = 1 / count(y = k)$, so that minority classes gain greater weight [35, 36]. This method is often applied to weighted loss functions, but it is static because it depends only on the distribution of the sample size [37, 38].

Another widely used method is balanced class weight, which determines the weight based on the total number of samples and the number of classes using the formula $c_k = n / (K \cdot count(y = k))$. This scheme gives more proportional weight to minority classes and is widely adopted in cost-sensitive learning, but it remains static because it does not yet consider adaptive methods [35, 37].

This study proposes a new class weighting method formulated as follows:

$$Costs = \left[\lambda x \cdot \left(\frac{x}{\max\left(x\right)}\right)^{(1)}, \left(\frac{x}{\max\left(x\right)}\right)^{(2)}\right], \tag{23}$$

$$x^{(1)} = \left[1 - \left(\lambda x \cdot \frac{Count(y=j)}{m} \right) \right]_{j \in \{1,\ldots,K\}}, \quad (24)$$

$$x^{(2)} = \left[\frac{1}{\frac{Count(y=j)}{\max(Count(y))}} \right]_{j \in \{1,\ldots,K\}}. \quad (25)$$

This formula combines two weighting methods based on class proportion relative to total data, normalization relative to the majority class, and adaptive parameter λ to control sensitivity. This hybrid method has not been explicitly found in previous studies and is a major contribution to this study.

5. 2. Implementation of the proposed cost-sensitive artificial neural network (COST-ANN) method

5. 2. 1. Grid search with cross-validation

The optimal cost vector is determined using grid search with cross-validation to evaluate various candidates based on method performance and generalization ability. The search results are displayed in the form of graphs and performance metrics, with Iteration 1 presented in Fig. 2.

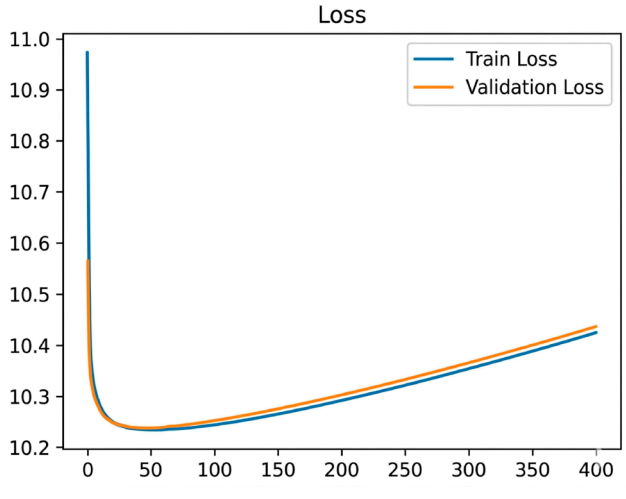

Fig. 2. Training and validation loss curves for fold 1 in iteration 1 using 5-fold cross-validation

The loss graph demonstrate that the Train and validation loss values drop sharply at the beginning of the epoch, indicating that the learning process is proceeding effectively in the early stages. After reaching a minimum, both curves gradually rise slightly but remain stable and close to one another. This pattern indicates that the method has reached a state of convergence and is not overfitting. The accuracy results are shown in Fig. 3.

The accuracy plot demonstrate that the training and validation accuracies increase rapidly at the beginning of the epoch, reaching high values in the range of 97–98%. After that, both curves tend to stabilize with only a very small difference. This indicates that the method has good generalization ability and is capable of maintaining consistent performance between the training and validation data.

After the process for fold 1, testing is then performed on fold 2 in iteration 1, as shown in Fig. 4.

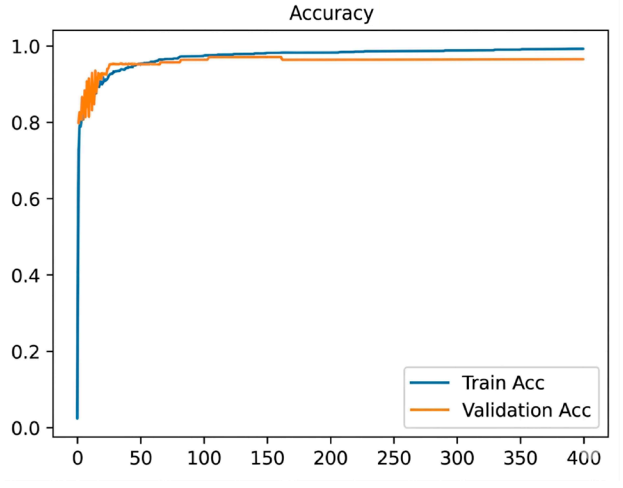

Fig. 3. Training and validation accuracy curves for fold 1 in iteration 1 using 5-fold cross-validation

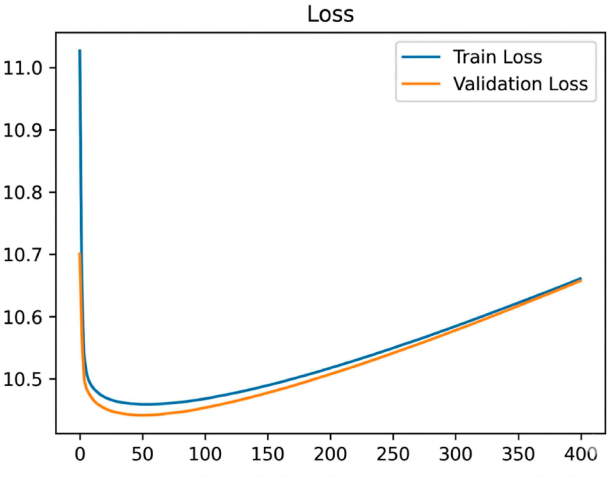

Fig. 4. Training and validation loss curves for fold 2 in iteration 1 using 5-fold cross-validation

In the early stages, the training loss remains high, then decreases steadily as the number of epochs increases. The validation loss follows a similar trend to the training loss, indicating a stable and unbiased learning process. The accuracy results are shown in Fig. 5.

The training accuracy was initially low, while the validation accuracy was already quite high. After a few epochs, both the training and validation accuracies increased steadily to around 98–99%, indicating rapid convergence and good method performance. Next, testing was performed on fold 3 in iteration 1, as shown in Fig. 6.

Fold 3 demonstrates a rapid decrease in training loss from the start, followed by a corresponding decrease in validation loss, with a very small difference, indicating stable learning. The accuracy results are shown in Fig. 7.

Training accuracy increased gradually until it reached an optimal value, while validation accuracy remained high and consistent, indicating good generalization ability. Next, testing was conducted on fold 4 in iteration 1, as shown in Fig. 8.

On fold 4, the method demonstrates excellent convergence, with a rapid increase in accuracy from the early epochs. The training loss and validation loss values decrease steadily and remain nearly parallel throughout the training process. The accuracy results are shown in Fig. 9.

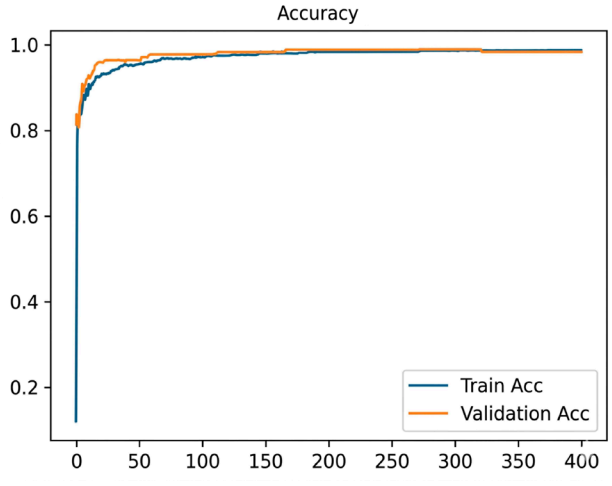

Fig. 5. Training and validation accuracy curves for fold 2 in iteration 1 using 5-fold cross-validation

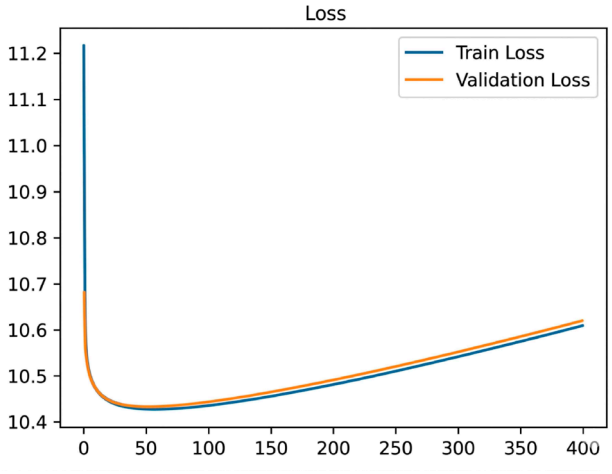

Fig. 6. Training and validation loss curves for fold 3 in iteration 1 using 5-fold cross-validation

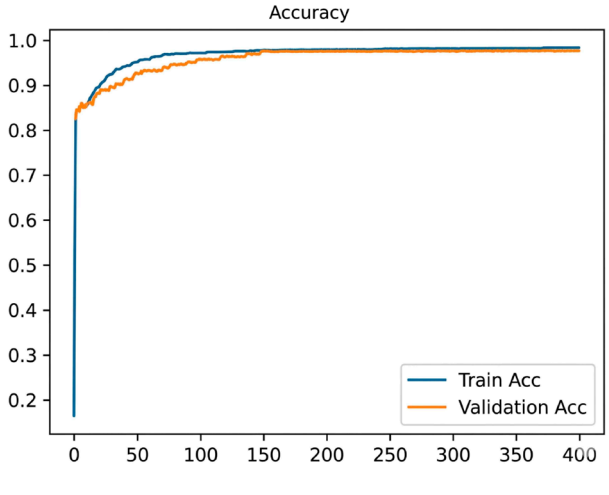

Fig. 7. Training and validation accuracy curves for fold 3 in iteration 1 using 5-fold cross-validation

The training and validation accuracies reach high values with a very small difference, indicating that the method is not overfitting and is able to capture the data patterns effectively on this fold. Next, testing was conducted on fold 5 in iteration 1, as shown in Fig. 10.

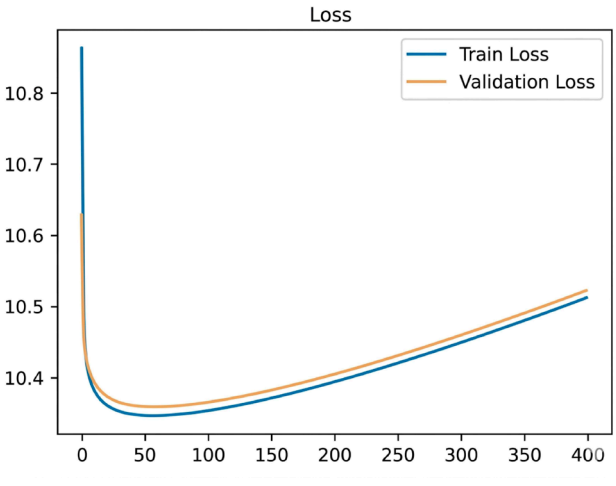

Fig. 8. Training and validation loss curves for fold 4 in iteration 1 using 5-fold cross-validation

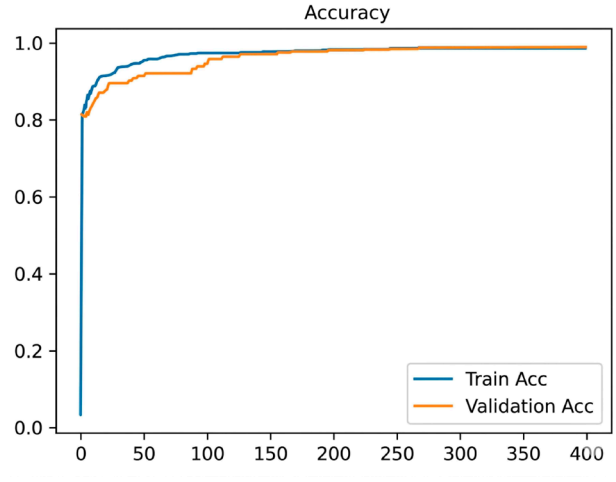

Fig. 9. Training and validation accuracy curves for fold 4 in iteration 1 using 5-fold cross-validation

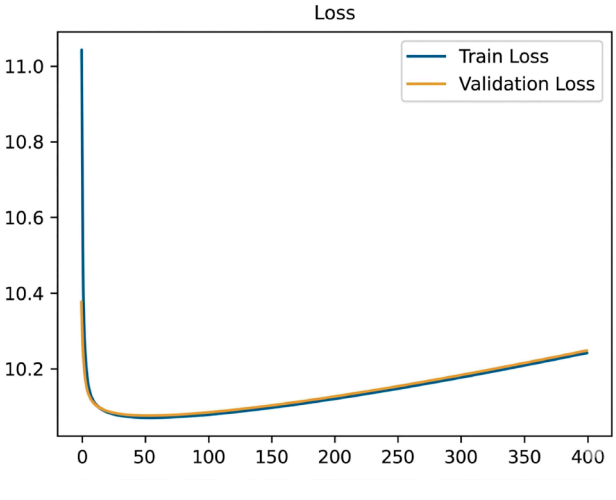

Fig. 10. Training and validation loss curves for fold 5 in iteration 1 using 5-fold cross-validation

Fold 5 also exhibits a learning pattern consistent with the previous folds. At the start of training, the loss value is relatively high and the training accuracy is low, but both improve significantly as the number of epochs increases. The accuracy results are shown in Fig. 11.

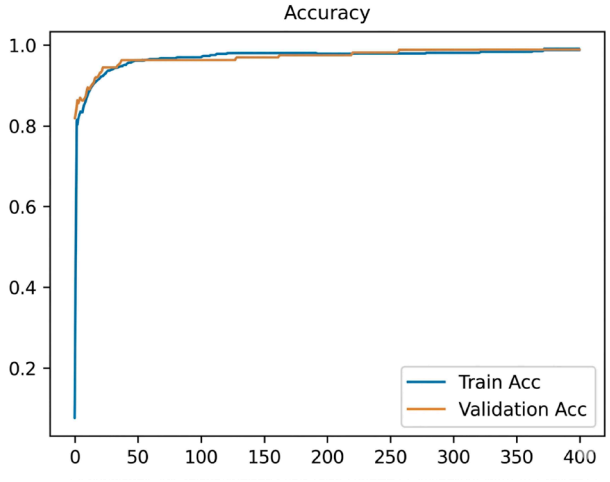

Fig. 11. Training and validation accuracy curves for fold 5 in iteration 1 using 5-fold cross-validation

The training and validation accuracies eventually stabilize at high levels with only a small difference between them. This indicates that the method performs stably and is robust to data variations in the final fold.

Overall, the five folds demonstrate a consistent learning pattern characterized by stable convergence, significant improvements in accuracy, and minimal differences between the training and validation sets. This suggests that the COST-ANN method with the cost vector used possesses good generalization ability and does not suffer from overfitting; consequently, the results of the first iteration can serve as a baseline for comparison in subsequent iterations, as shown in Fig. 12.

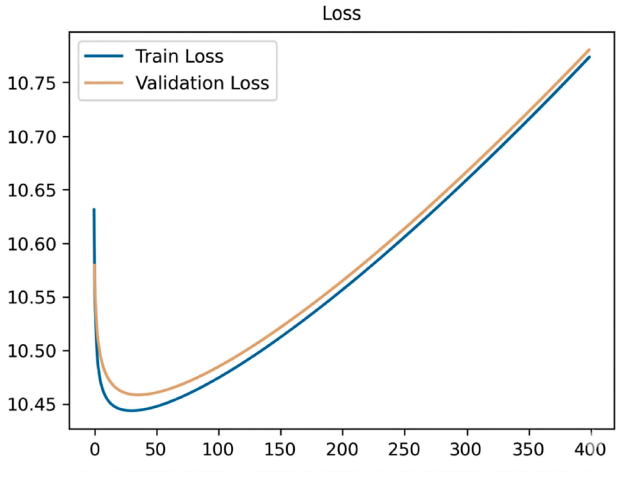

Fig. 12. Training and validation loss curves for fold 1 in iteration 2 using 5-fold cross-validation

At the beginning of training, the training loss was relatively high, then decreased steadily as the number of epochs increased. The validation loss followed a pattern parallel to the training loss, with a very small difference until the end of training, indicating a stable learning process. The accuracy results are shown in Fig. 13.

Training accuracy was initially low, while validation accuracy was already at a fairly good level. As the number of epochs increased, training accuracy improved significantly to reach the 96–97% range, followed by validation accuracy approaching 97–98%, with a very small difference, indicating the method's good generalization ability.

Following the evaluation process for fold 1, the evaluation for fold 2 was then conducted, as shown in Fig. 14.

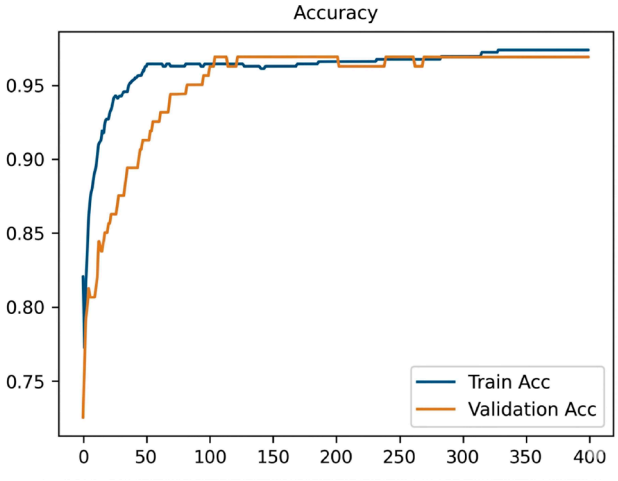

Fig. 13. Training and validation accuracy curves for fold 1 in iteration 2 using 5-fold cross-validation

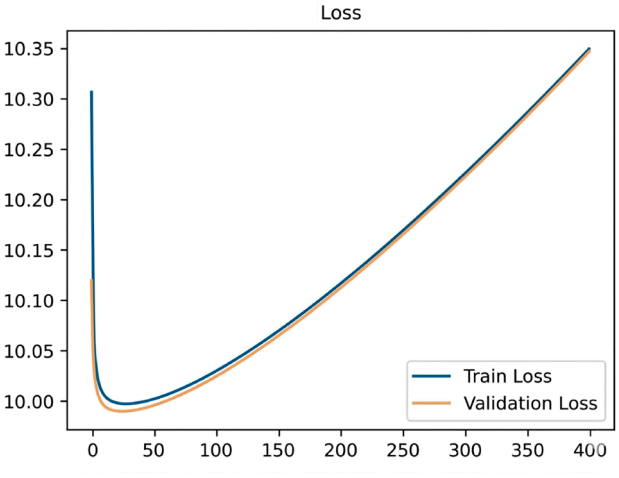

Fig. 14. Training and validation loss curves for fold 2 in iteration 2 using 5-fold cross-validation

The training loss and validation loss decreased consistently and in tandem throughout the epochs, indicating that the optimization process was effective with no signs of overfitting. The accuracy results are shown in Fig. 15.

At the beginning of training, there was a discrepancy between the training and validation accuracies, but the method quickly reached a state of convergence. Both training and validation accuracies increased to a high range (around 96–98%) with a stable curve pattern.

Following the evaluation process for fold 2, the evaluation for fold 3 was then conducted, as shown in Fig. 16.

At fold 3, the training loss decreases rapidly at the beginning of training, followed by the validation loss, which follows a similar pattern. The two curves tend to stay close throughout the training process, indicating stable learning. The accuracy results are shown in Fig. 17.

The training accuracy increases gradually, while the validation accuracy remains high and consistent. The closeness of the two curves indicates that the method is able to adapt well to the data distribution and maintain balanced performance.

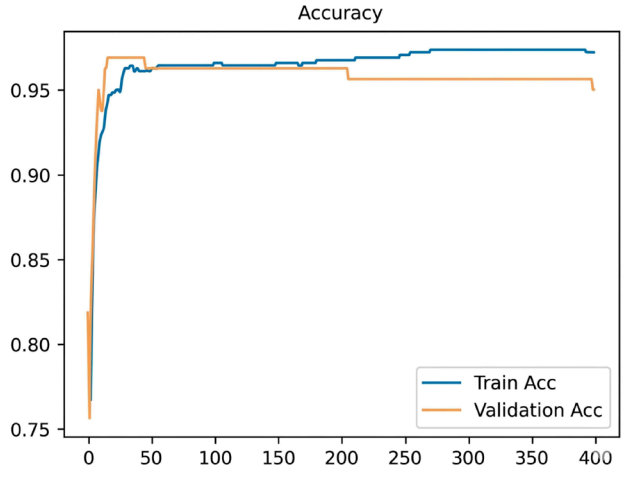

Fig. 15. Training and validation accuracy curves for fold 2 in iteration 2 using 5-fold cross-validation

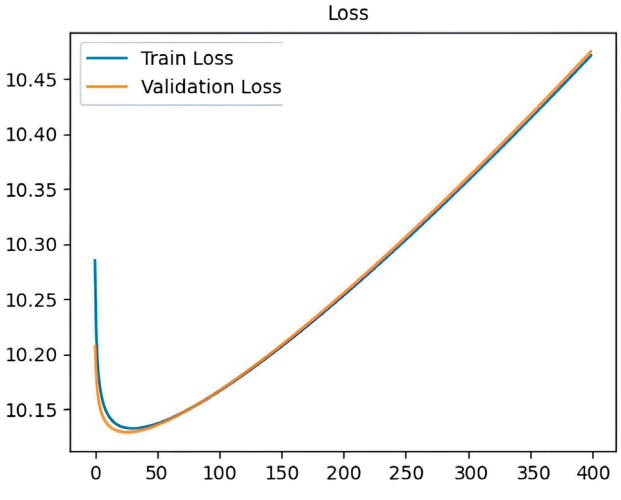

Fig. 16. Training and validation loss curves for fold 3 in iteration 2 using 5-fold cross-validation

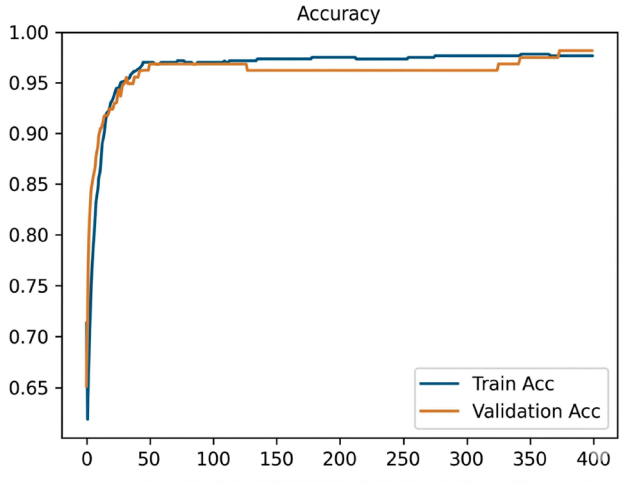

Fig. 17. Training and validation accuracy curves for fold 3 in iteration 2 using 5-fold cross-validation

Following the evaluation process for fold 3, the evaluation for fold 4 was then conducted, as shown in Fig. 18.

On fold 4, the method demonstrates excellent convergence, with a rapid and stable increase in accuracy. The training loss and validation loss values decrease in parallel. The accuracy results are shown in Fig. 19.

The training and validation accuracies reach high values with a very small difference. This indicates that the method is not overfitting and has strong generalization capabilities for the data in this fold.

Following the evaluation process for fold 4, the evaluation for fold 5 was then conducted, as shown in Fig. 20.

Fold 5 exhibits a learning pattern consistent with the other folds. In the early stages, the loss value remains relatively high and training accuracy is not yet optimal, but both improve significantly as the number of epochs increases. The accuracy results are shown in Fig. 21.

Training and validation accuracy eventually stabilize at a high level (around 96–98%) with minimal variation. This indicates that the method performs robustly against data variations in the final fold.

Overall, all folds in the second grid search demonstrate a stable and consistent learning pattern, with a significant improvement in accuracy and a small discrepancy between the training and validation sets. This suggests that the second cost vector is capable of producing a more effective and stable learning process compared to the previous candidate.

The results of the third iteration of the grid search using 5-fold cross-validation are presented in Fig. 22.

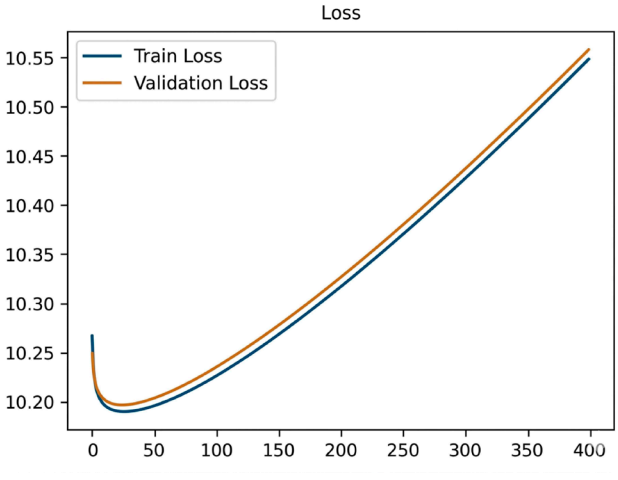

Fig. 18. Training and validation loss curves for fold 4 in iteration 2 using 5-fold cross-validation

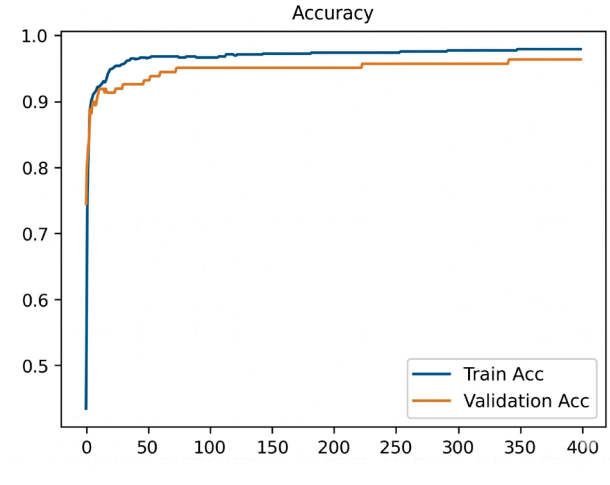

Fig. 19. Training and validation accuracy curves for fold 4 in iteration 2 using 5-fold cross-validation

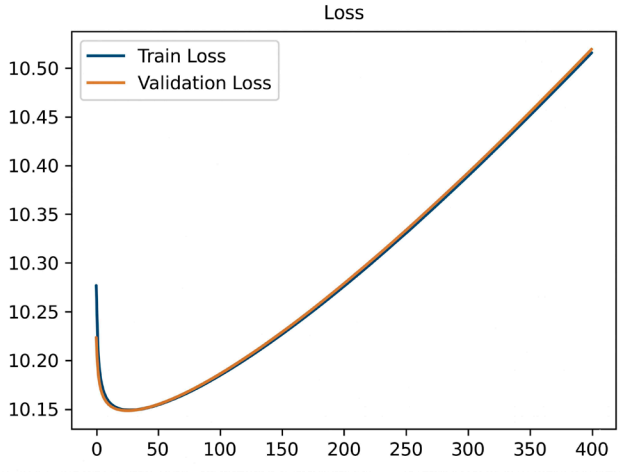

Fig. 20. Training and validation loss curves for fold 5 in iteration 2 using 5-fold cross-validation

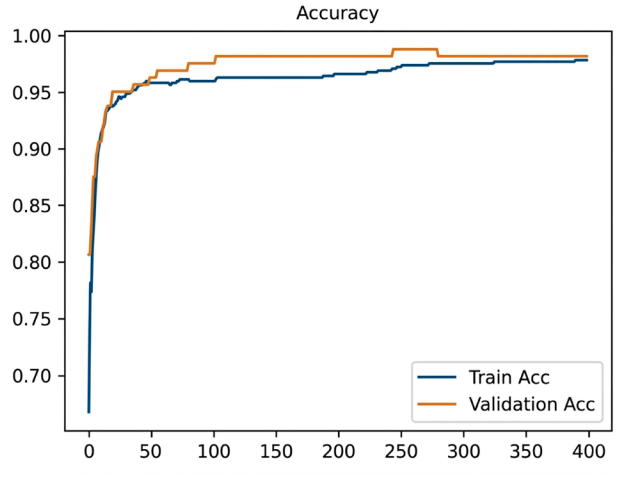

Fig. 21. Training and validation accuracy curves for fold 5 in iteration 2 using 5-fold cross-validation

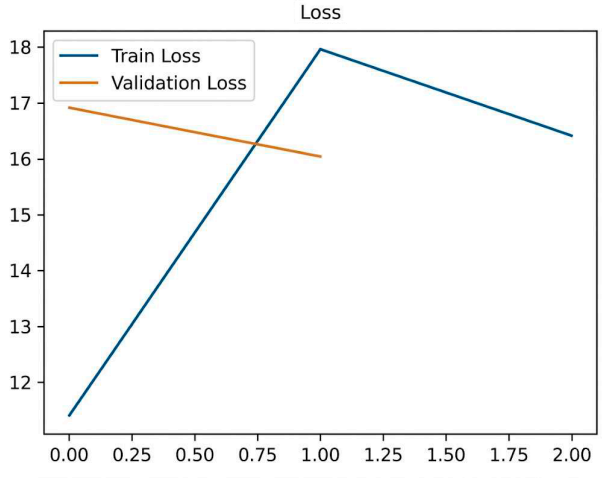

Fig. 22. Training and validation loss curves for fold 1 in iteration 3 using 5-fold cross-validation

In fold 1, the method exhibited unstable learning behavior from the start of training. The train loss and validation loss values did not follow a normal downward trend but instead fluctuated and tended to increase. The accuracy results are shown in Fig. 23.

The training and validation accuracies did not demonstrate significant improvement and remained at very low levels. This implies that the method failed to learn due to numerical instability caused by the cost vector used.

Following the evaluation process for fold 1, the evaluation for fold 2 was then conducted, as shown in Fig. 24.

In fold 2, the method's instability is more evident, with sharp fluctuations in the train loss and validation loss values. The loss curves do not demonstrate convergence but rather uncontrolled fluctuations between epochs. The accuracy results are shown in Fig. 25.

Training and validation accuracy remain stagnant at low levels with no improvement. This reinforces the indication that the optimization process is not proceeding properly due to numerical issues with the loss function.

Following the evaluation process for fold 2, the evaluation for fold 3 was then conducted, as shown in Fig. 26.

Fold 3 exhibits a non-convergent learning pattern, in which the loss value does not decrease gradually but fluctuates inconsistently. The accuracy results are shown in Fig. 27.

The training and validation accuracies do not demonstrate any significant improvement throughout the training process. This confirms that the method is unable to adjust its parameters effectively, resulting in its failure to recognize the data patterns in this fold.

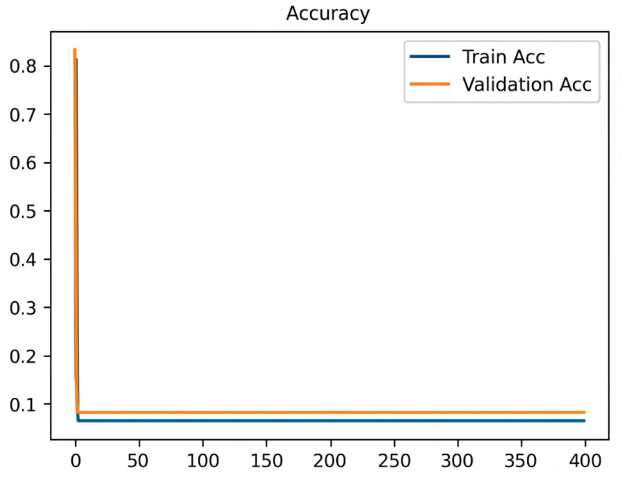

Fig. 23. Training and validation accuracy curves for fold 1 in iteration 3 using 5-fold cross-validation

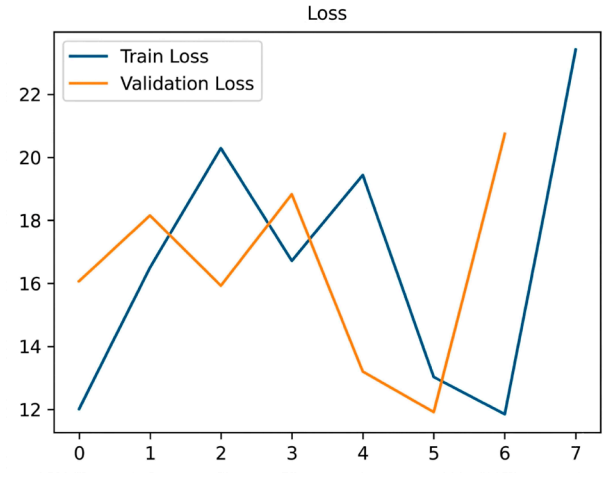

Fig. 24. Training and validation loss curves for fold 2 in iteration 3 using 5-fold cross-validation

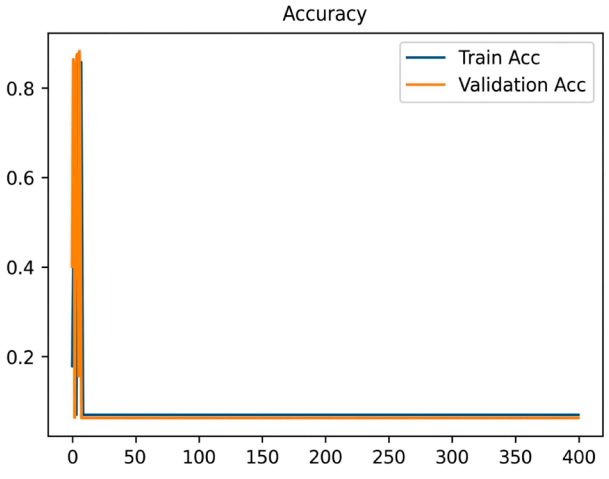

Fig. 25. Training and validation accuracy curves for fold 2 in iteration 3 using 5-fold cross-validation

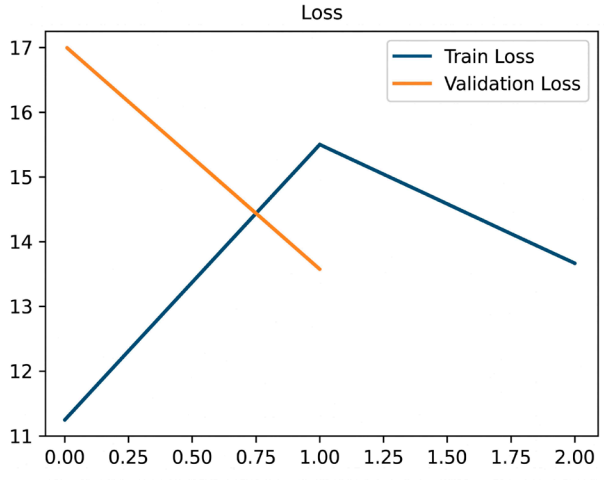

Fig. 26. Training and validation loss curves for fold 3 in iteration 3 using 5-fold cross-validation

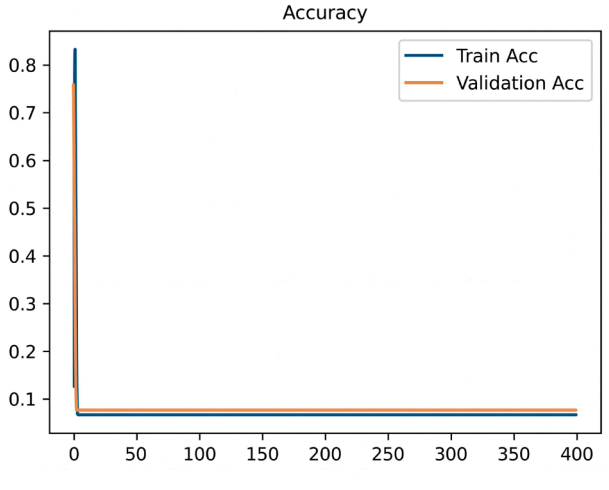

Fig. 27. Training and validation accuracy curves for fold 3 in iteration 3 using 5-fold cross-validation

Following the evaluation process for fold 3, the evaluation for fold 4 was then conducted, as shown in Fig. 28.

In fold 4, the method experienced a similar learning failure, characterized by high and unstable train loss values and a validation loss that did not demonstrating a downward trend. The accuracy results are shown in Fig. 29.

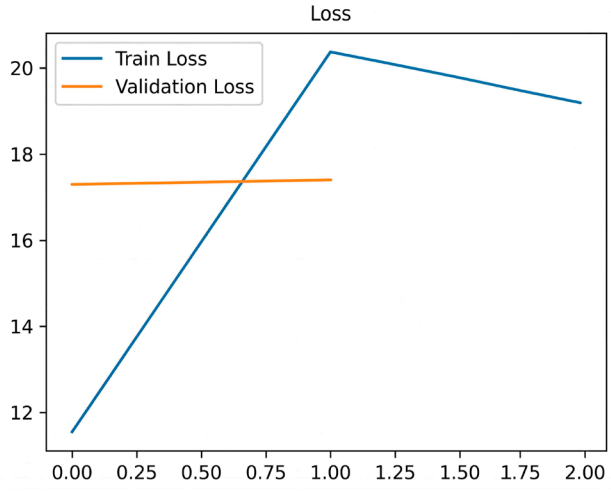

Fig. 28. Training and validation loss curves for fold 4 in iteration 3 using 5-fold cross-validation

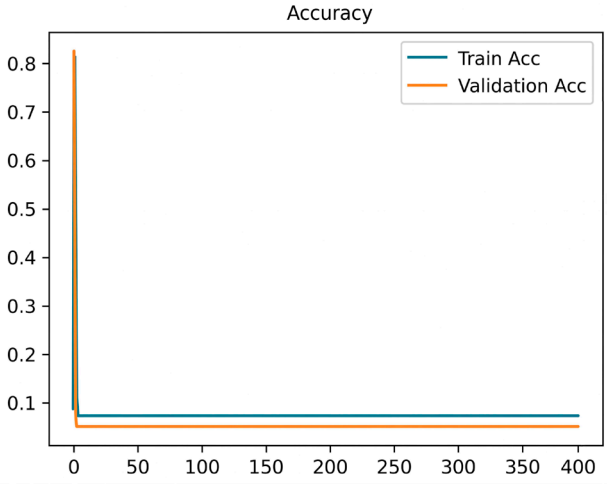

Fig. 29. Training and validation accuracy curves for fold 4 in iteration 3 using 5-fold cross-validation

Accuracy remained low and did not improve significantly. This confirms that the backpropagation process was not running optimally due to instability in the loss function calculations.

Following the evaluation process for fold 4, the evaluation for fold 5 was then conducted, as shown in Fig. 30.

Fold 5 also exhibits a pattern of learning failure consistent with the other folds. The loss value fluctuates significantly and does not converge. The accuracy results are shown in Fig. 31.

Training and validation accuracy remain very low throughout the epoch. This confirms that the method is unable to learn from the data due to issues with the cost vector used.

The selection of the optimal cost vector was performed via grid search with k-fold cross-validation across three candidates. The results demonstrate that Iteration 1 produced stable learning but did not yet address class imbalance. Iteration 2 delivers the best performance with lower validation loss and more stable accuracy, making it effective in handling imbalance. Conversely, iteration 3 fails because it produces a NaN loss and low

accuracy, so it is eliminated. Based on the comparison, the best cost vector is obtained in iteration 2 because it provides the most stable and optimal performance. The results of the grid search with cross-validation evaluation are shown in Fig. 32.

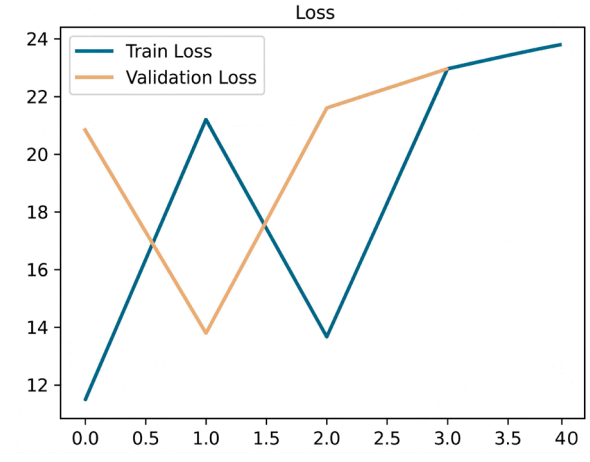

Fig. 30. Training and validation loss curves for fold 5 in iteration 3 using 5-fold cross-validation

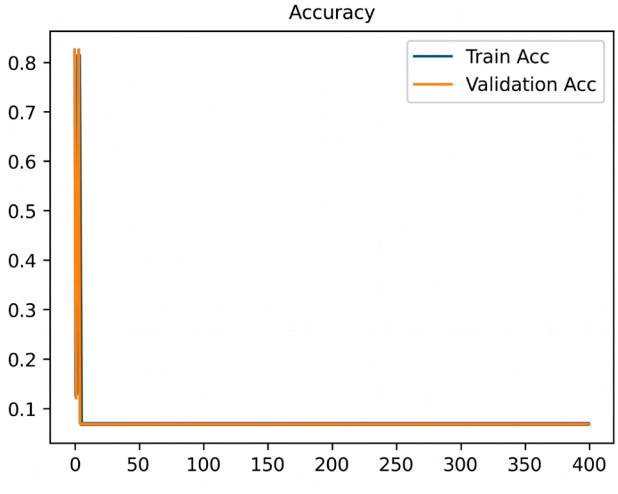

Fig. 31. Training and validation accuracy curves for fold 5 in iteration 3 using 5-fold cross-validation

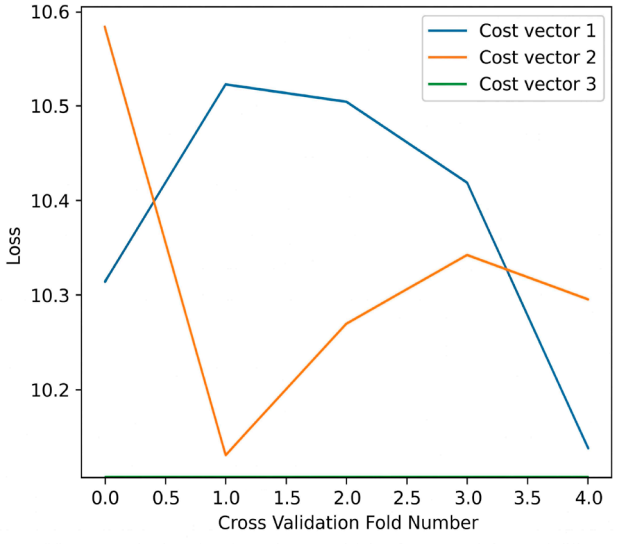

Fig. 32. Grid Search with Cross-Validation Evaluation

Fig. 32 the results shows a comparison graph comparing the cross-validation loss values, which reinforces the findings in the grid search 1–3 average loss table. The graph demonstrates that cost vector 2 consistently produces the lowest and most stable loss values compared to cost vector 1 and cost vector 3 in almost all folds. Cost vector 1 still shows greater loss fluctuations, while cost vector 3 fails to produce a valid training process. The alignment between the numerical results in the table and the visual patterns in this graph confirms that cost vector 2 is the optimal and most effective parameter to use in training the final COST-ANN method.

5. 2. 2. COST-ANN optimization and performance results

The optimization process aims to minimize the loss function so that the method can produce more accurate classification. The method training process is performed iteratively using a backpropagation-based gradient descent algorithm. Each output row demonstrating the progression of the train loss, test loss, train accuracy, and test accuracy values at each epoch as a concrete representation of the evaluation and parameter update steps [24], illustrated by the graph showing the shift in the training results of the COST-ANN method in Fig. 33.

The training graph demonstrate that the COST-ANN method with cost vector number 2 converged stably over 400 epochs. Training and testing accuracy increased rapidly in the early phase and then stabilized in the range of 0.98 and 0.97 with a small difference, indicating the method's good generalization ability. The training and testing loss values remain relatively higher due to the cost-sensitive weighting method, but demonstrating stable convergence patterns. The alignment of the loss pattern and accuracy stability indicate that the optimization process is consistent without overfitting, enabling the method to effectively handle imbalanced multi-class classification.

A classification performance evaluation was conducted to assess the effectiveness of the cost-sensitive artificial neural network (COST-ANN) method in handling an imbalanced ISPU dataset using a confusion matrix, which provides information on the distribution of correct and incorrect classifications across each class, particularly in the context of imbalanced data [9], as described in Fig. 34.

The confusion matrix results demonstrating strong alignment with class-based evaluation metrics in the COST-ANN method. In the GOOD class, the classification accuracy rate reached 91% with a class accuracy value of 0.988, recall of 0.913, and precision of 0.913, indicating that most of the data in this class can be recognized and predicted correctly. The MODERATE class shows significant performance with a confusion matrix diagonal value of 0.97, supported by recall 0.972, precision 0.993, and F1-score 0.982, which indicates an excellent balance of classification. For the UNHEALTHY class, all actual data was successfully recognized with a recall of 1.000, although a precision of 0.870 indicates that there are still a few incorrect classification from other classes, resulting in an F1-score of 0.930. Overall, the consistency between the confusion matrix and the evaluation metrics per class confirms that the COST-ANN method is capable of providing accurate, balanced classification performance and has good generalization capabilities across all three classes.

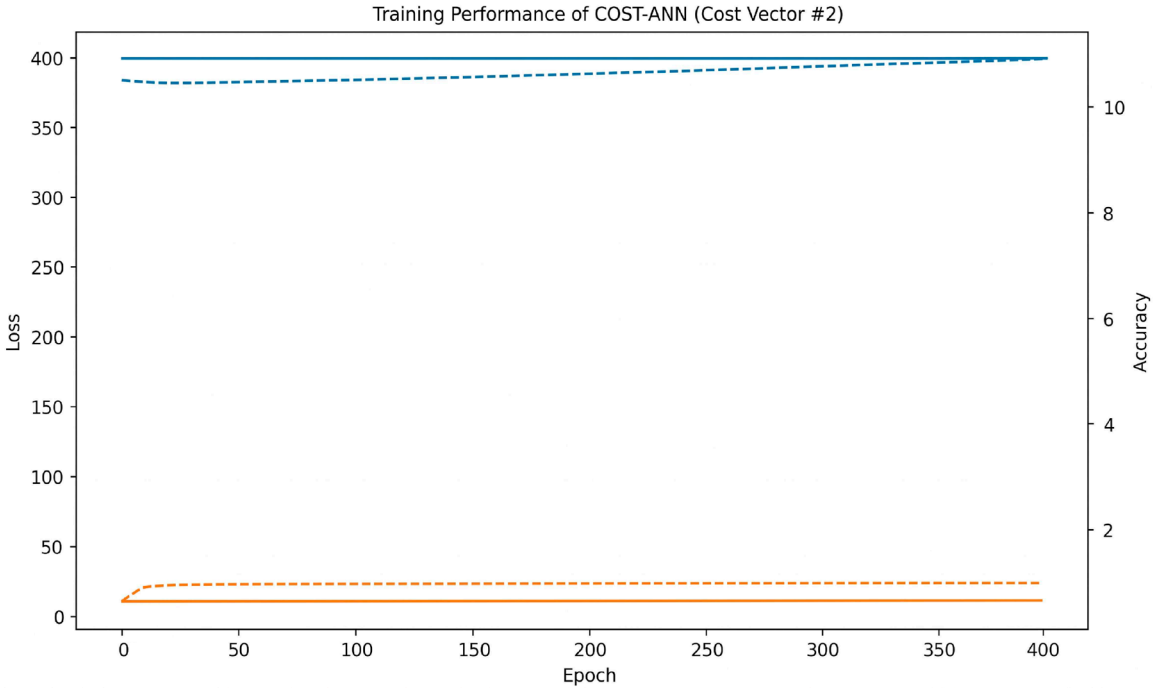

Fig. 33. Shift graph of COST-ANN method training results

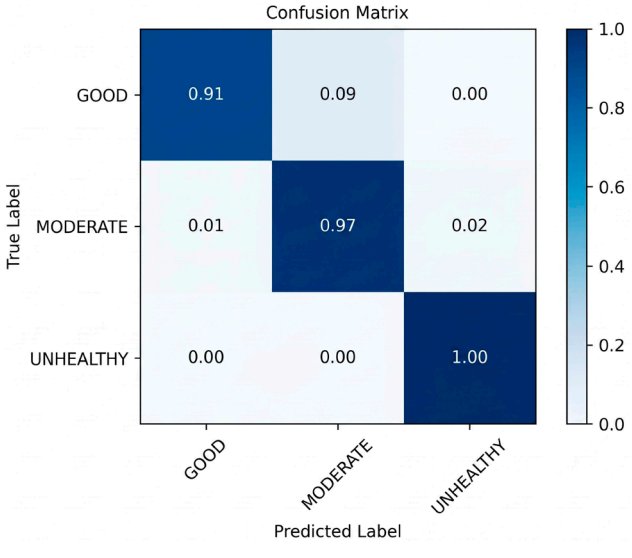

Fig. 34. Confusion matrix COST-ANN Optimal Cost Dataset Air Pollution Standard Index (ISPU)

5. 3. Implementation of cost-ANN with integration in principal Component Analysis (PCA)

5. 3. 1. Grid search with cross-validation

The initial stage of implementing COST-ANN optimization with PCA was carried out using the grid search with cross-validation method to systematically determine the best parameters. This process evaluated various candidate cost vectors against data that had been transformed using PCA. The evaluation results were used to select the parameters that provided the lowest loss value and the most stable accuracy. Thus, this stage became an important basis for forming an optimal PCA-based COST-ANN method.

The results of the grid search with cross-validation are shown in Fig. 35.

At the beginning of training, the training loss was relatively high, then decreased in parallel with the validation loss as the number of epochs increased. The closely aligned curves indicate a stable learning process. The training and validation accuracy results are shown in Fig. 36.

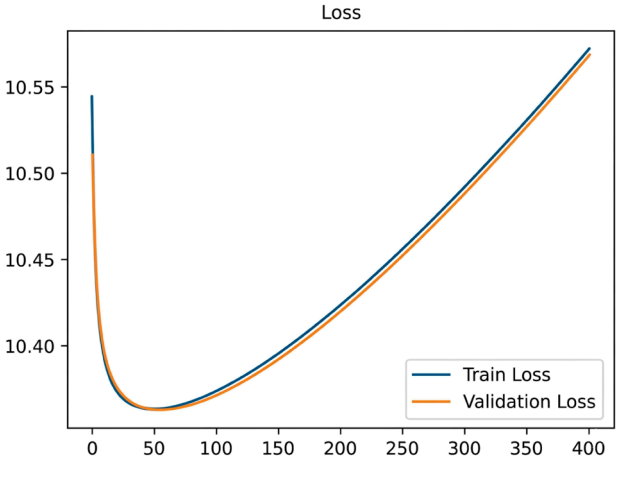

Fig. 35. Training and validation loss curves for fold 1 in iteration 1 using 5-fold cross-validation

Training accuracy was initially low, while validation accuracy was already quite good from the start. As the number of epochs increased, training accuracy improved significantly to the range of 96–98%, followed by validation accuracy that remained stable above 95%, indicating good generalization ability.

Following the evaluation process for fold 1, the evaluation for fold 2 was then conducted, as shown in Fig. 37.

In fold 2, the loss curve demonstrates a steady decline. This indicates that the method is able to adapt well to the data distribution in this fold. The training and validation accuracy results are shown in Fig. 38.

The method exhibits a slightly slower learning pattern in the early stages compared to other folds, with lower training accuracy. However, performance improves con-

sistently as the number of epochs increases. Training and validation accuracy eventually reach high values with a small difference.

Following the evaluation process for fold 2, the evaluation for fold 3 was then conducted, as shown in Fig. 39.

Fold 3 exhibits a stable learning pattern with a rapid decrease in training loss early in the training process. The training and validation accuracy results are shown in Fig. 40.

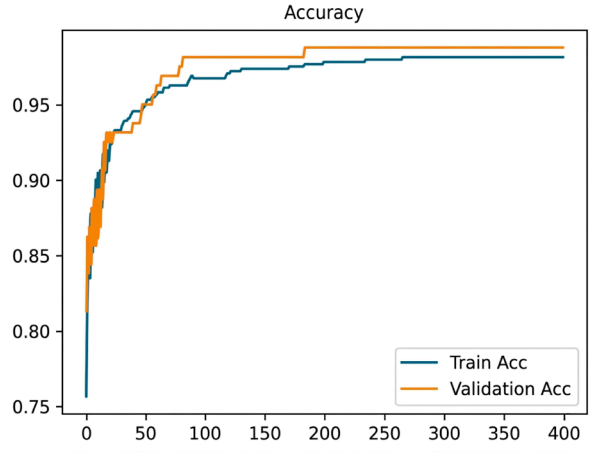

Fig. 36. Training and validation accuracy curves for fold 1 in iteration 1 using 5-fold cross-validation

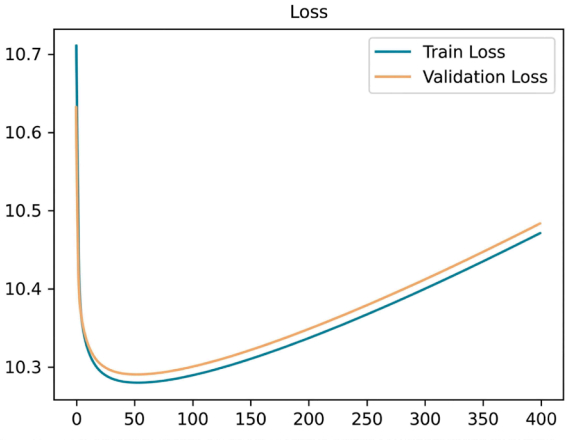

Fig. 37. Training and validation loss curves for fold 2 in iteration 1 using 5-fold cross-validation

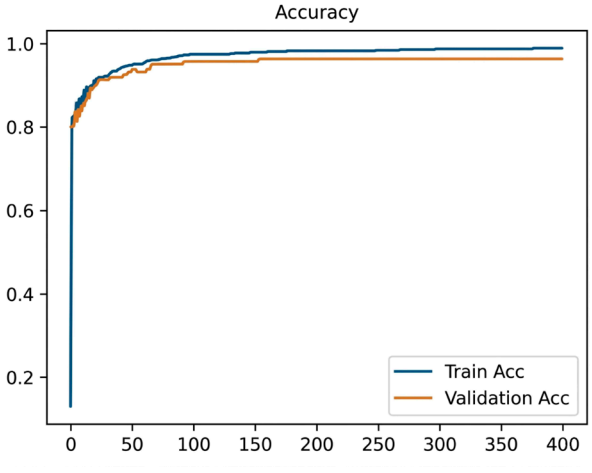

Fig. 38. Training and validation accuracy curves for fold 2 in iteration 1 using 5-fold cross-validation

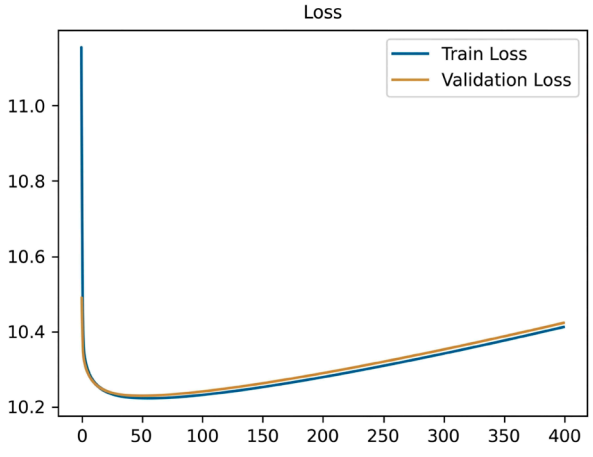

Fig. 39. Training and validation loss curves for fold 3 in iteration 1 using 5-fold cross-validation

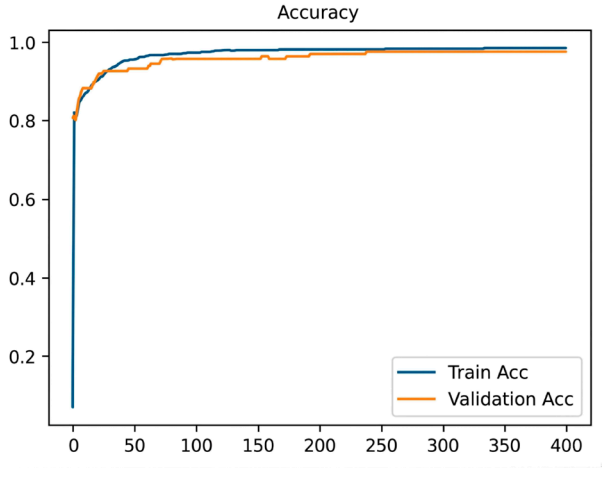

Fig. 40. Training and validation accuracy curves for fold 3 in iteration 1 using 5-fold cross-validation

Training accuracy increases gradually and is followed by consistently high validation accuracy. The training and validation curves tend to remain close throughout the training process, indicating that the method is not overfitting and is able to maintain balanced performance.

Following the evaluation process for fold 3, the evaluation for fold 4 was then conducted, as shown in Fig. 41.

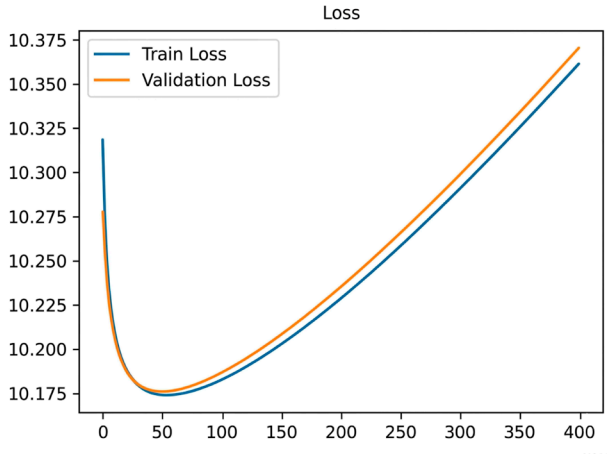

Fig. 41. Training and validation loss curves for fold 4 in iteration 1 using 5-fold cross-validation

In fold 4, the training loss and validation loss decreased gradually and in parallel throughout training, indicating a stable convergence process. The training and validation accuracy results are shown in Fig. 42.

Training and validation accuracy increased rapidly from the start until reaching high values with minimal difference, indicating the stability of the learning process and the method's ability to effectively capture data patterns.

Following the evaluation process for fold 4, the evaluation for fold 5 was then conducted, as shown in Fig. 43.

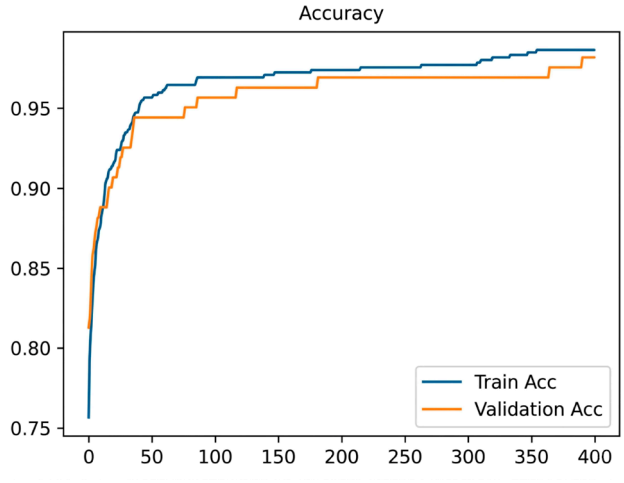

Fig. 42. Training and validation accuracy curves for fold 4 in iteration 1 using 5-fold cross-validation

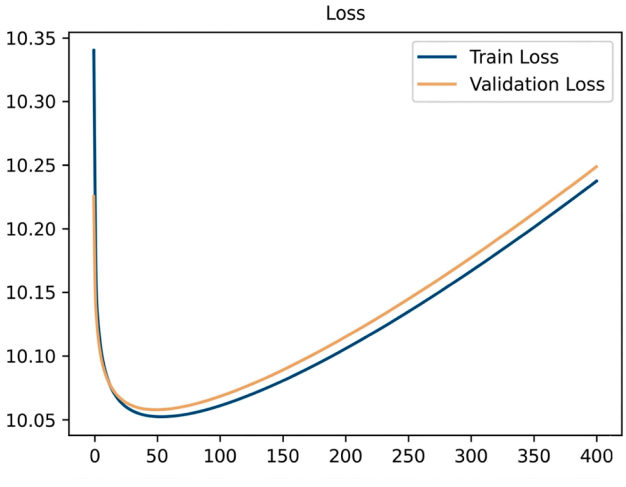

Fig. 43. Training and validation loss curves for fold 5 in iteration 1 using 5-fold cross-validation

At the start of training, the training loss was still high, but then decreased steadily as the number of epochs increased, with the validation loss following a similar pattern, indicating consistent learning. The training and validation accuracy results are shown in Fig. 44.

Training accuracy was relatively low at first, but then increased significantly until it stabilized at a high level, with validation accuracy following closely behind, indicating that the method performs robustly across data variations.

Overall, all folds exhibit a consistent learning pattern characterized by a significant increase in accuracy and a steady decrease in loss. This implies that the COST-ANN method with the first cost vector is capable of producing a stable learning process, although it has not yet fully optimized the handling of class imbalance. Next, an Iteration 2 grid search was performed, as shown in Fig. 45.

At the beginning of training, the training loss was relatively high, then decreased in parallel with the validation loss as the number of epochs increased, indicating a stable learning process. The training and validation accuracy results are shown in Fig. 46.

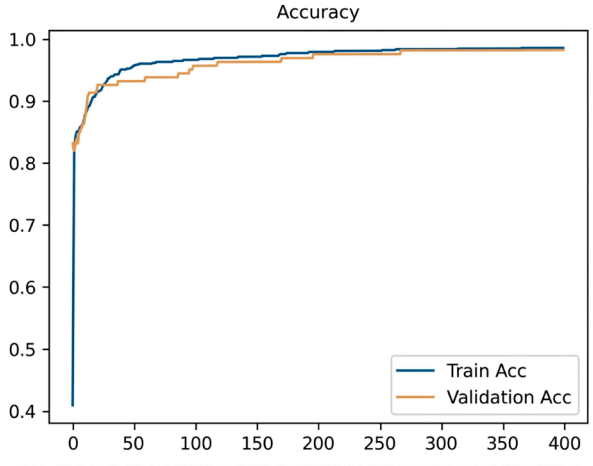

Fig. 44. Training and validation accuracy curves for fold 5 in iteration 1 using 5-fold cross-validation

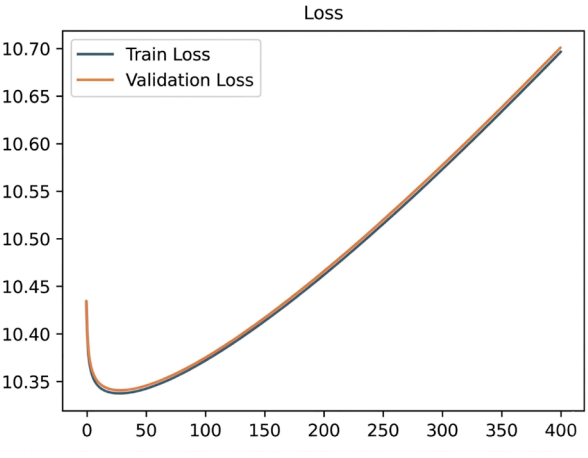

Fig. 45. Training and validation loss curves for fold 1 in iteration 2 using 5-fold cross-validation

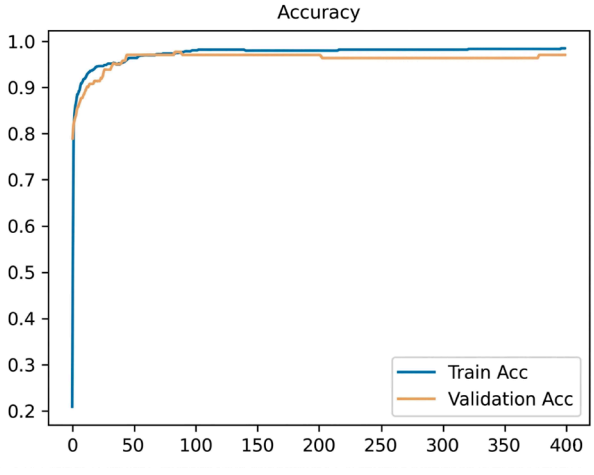

Fig. 46. Training and validation accuracy curves for fold 1 in iteration 2 using 5-fold cross-validation

Training accuracy was initially low, while validation accuracy was already quite good from the start. As the number of epochs increased, training accuracy improved significantly to the range of 97–98%, followed by validation accuracy that remained stable above 95%, indicating good generalization ability.

Following the evaluation process for fold 1, the evaluation for fold 2 was then conducted, as shown in Fig. 47.

The training loss and validation loss values decreased consistently and in tandem throughout training, indicating an effective optimization process. The training and validation accuracy results are shown in Fig. 48.

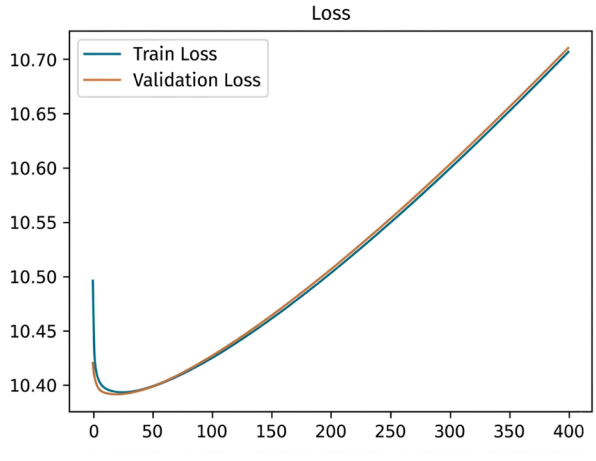

Fig. 47. Training and validation loss curves for fold 2 in iteration 2 using 5-fold cross-validation

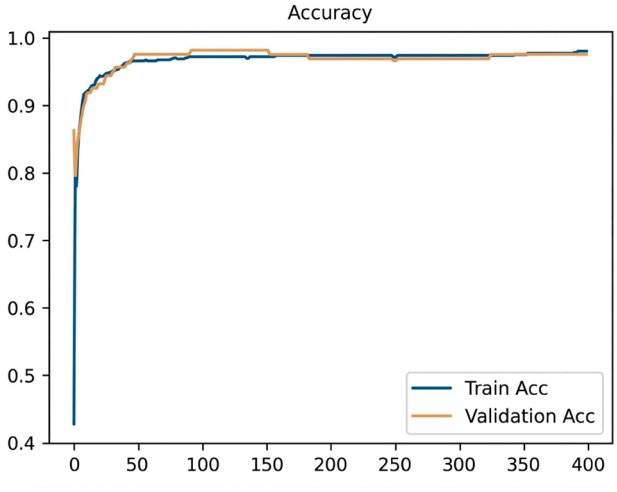

Fig. 48. Training and validation accuracy curves for fold 2 in iteration 2 using 5-fold cross-validation

Training accuracy was initially low but increased rapidly until it reached a state of convergence. Training and validation accuracy reached a high range (around 96–98%) with a stable curve pattern.

Following the evaluation process for fold 2, the evaluation for fold 3 was then conducted, as shown in Fig. 49.

Fold 3 exhibits a stable learning pattern with a rapid decrease in training loss early in the training process. The training and validation accuracy results are shown in Fig. 50.

Training accuracy increases gradually and is followed by consistently high validation accuracy. The training and validation curves tend to remain close throughout the training process, indicating that the method is able to maintain a balance in performance without overfitting.

Following the evaluation process for fold 3, the evaluation for fold 4 was then conducted, as shown in Fig. 51.

By fold 4, the method demonstrates good convergence, with a rapid increase in accuracy from the early epochs. The training loss and validation loss values decrease in parallel. The training and validation accuracy results are shown in Fig. 52.

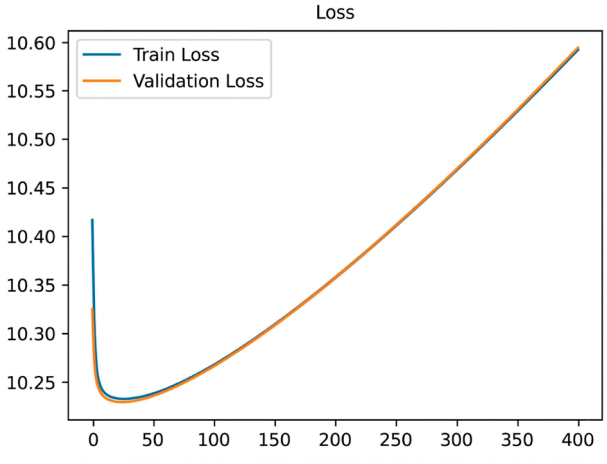

Fig. 49. Training and validation loss curves for fold 3 in iteration 2 using 5-fold cross-validation

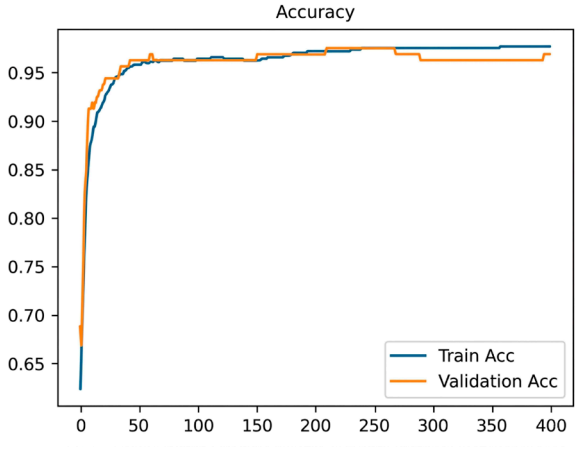

Fig. 50. Training and validation accuracy curves for fold 3 in iteration 2 using 5-fold cross-validation

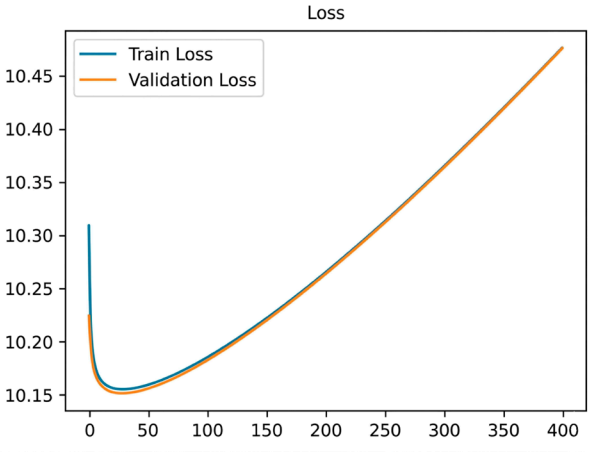

Fig. 51. Training and validation loss curves for fold 4 in iteration 2 using 5-fold cross-validation

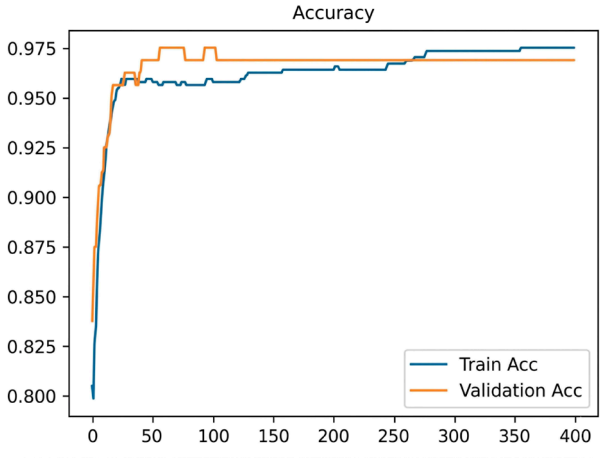

Fig. 52. Training and validation accuracy curves for fold 4 in iteration 2 using 5-fold cross-validation

The training and validation accuracies reach high values with only a small difference. This indicates that the method exhibits good learning stability and is capable of effectively capturing data patterns.

Following the evaluation process for fold 4, the evaluation for fold 5 was then conducted, as shown in Fig. 53.

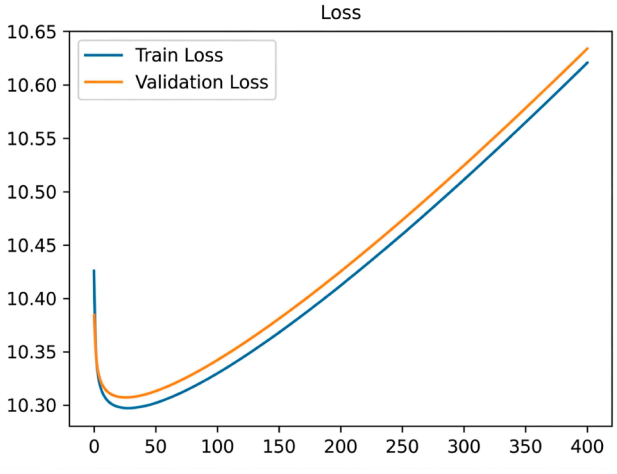

Fig. 53. Training and validation loss curves for fold 5 in iteration 2 using 5-fold cross-validation

Fold 5 exhibits a learning pattern consistent with the other folds. In the early stages, the loss value remains relatively high and the training accuracy is not yet optimal, but it improves significantly as the number of epochs increases. The training and validation accuracy results are shown in Fig. 54.

Training and validation accuracy eventually stabilize at a high range (around 96–98%) with minimal variation, indicating that the method performs robustly against data variations.

Overall, all folds in the second grid search exhibit stable learning patterns, significant accuracy improvements, and small differences between training and validation. This confirms that the second cost vector is capable of producing a more effective and balanced learning process in handling class imbalance. Next, the third iteration of the grid search is performed, as shown in Fig. 55.

At fold 1, the method exhibited unstable learning behavior from the start. The training loss and validation loss values fluctuated without a clear downward trend. The training and validation accuracy results are shown in Fig. 56.

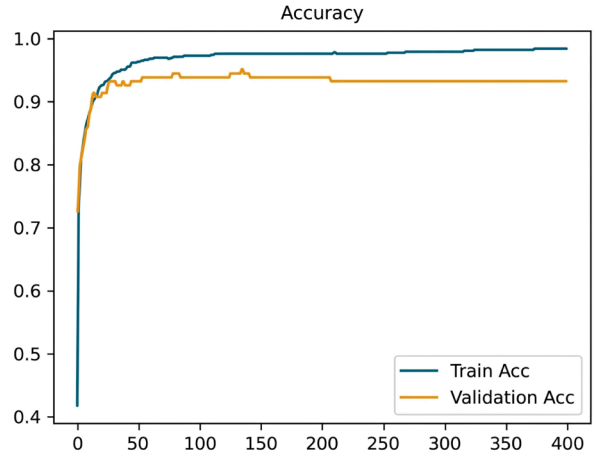

Fig. 54. Training and validation accuracy curves for fold 5 in iteration 2 using 5-fold cross-validation

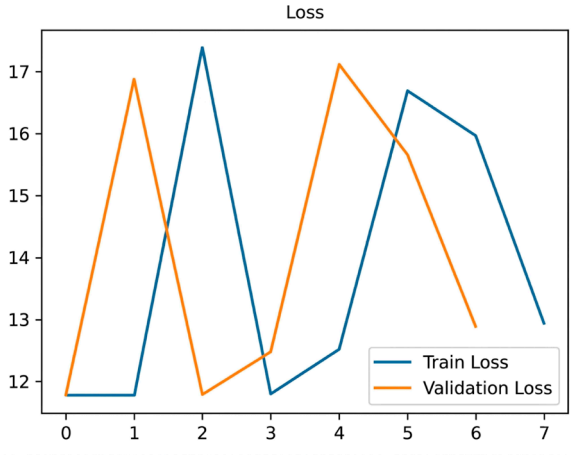

Fig. 55. Training and validation loss curves for fold 1 in iteration 3 using 5-fold cross-validation

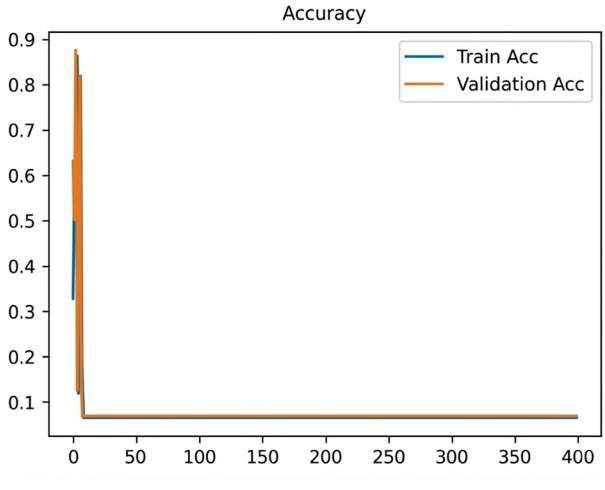

Fig. 56. Training and validation accuracy curves for fold 1 in iteration 3 using 5-fold cross-validation

The training and validation accuracy remained very low and did not improve. This indicates that the optimization process was not proceeding smoothly due to numerical instability.

Following the evaluation process for fold 2, the evaluation for fold 5 was then conducted, as shown in Fig. 57.

In fold 2, instability is more pronounced, with sharp fluctuations in the loss values at each epoch. The train loss and

validation loss curves do not demonstrate convergence but rather erratic changes. The training and validation accuracy results are shown in Fig. 58.

Accuracy remains low and stagnant, indicating that the method failed to learn the data patterns effectively.

Following the evaluation process for fold 3, the evaluation for fold 5 was then conducted, as shown in Fig. 59.

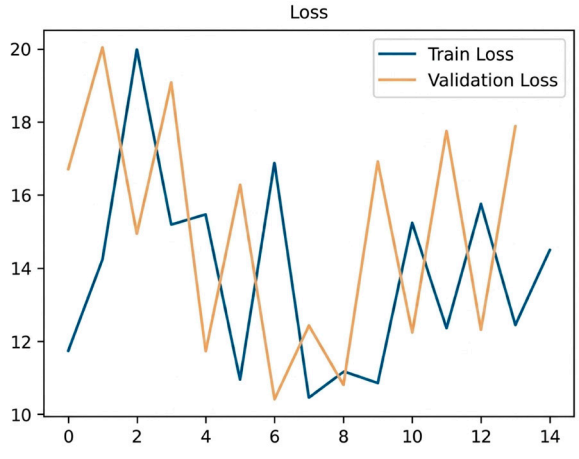

Fig. 57. Training and validation loss curves for fold 2 in iteration 3 using 5-fold cross-validation

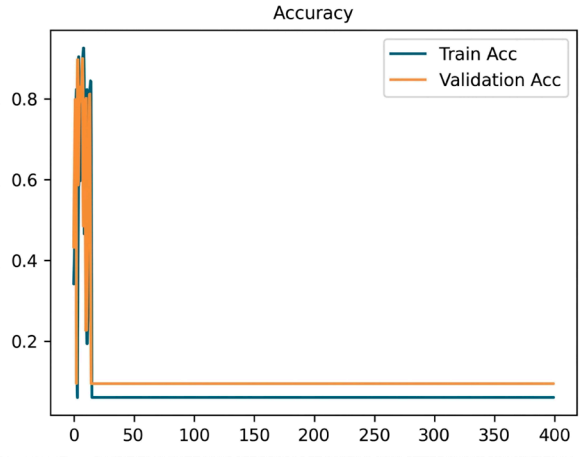

Fig. 58. Training and validation accuracy curves for fold 2 in iteration 3 using 5-fold cross-validation

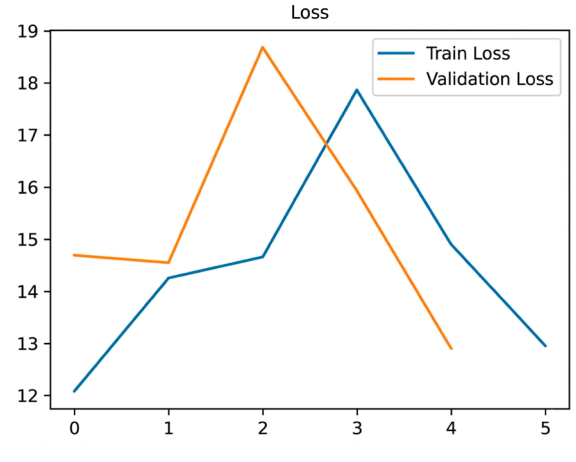

Fig. 59. Training and validation loss curves for fold 3 in iteration 3 using 5-fold cross-validation

Fold 3 exhibits a non-convergent learning pattern, in which the loss value fluctuates inconsistently throughout training. The training and validation accuracy results are shown in Fig. 60.

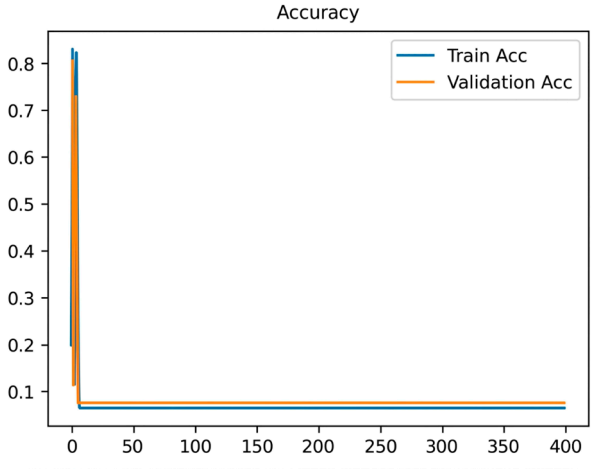

Fig. 60. Training and validation accuracy curves for fold 3 in iteration 3 using 5-fold cross-validation

The training and validation accuracy did not improve significantly and remained at very low levels. This confirms that the parameter update process is not proceeding optimally.

Following the evaluation process for fold 4, the evaluation for fold 5 was then conducted, as shown in Fig. 61.

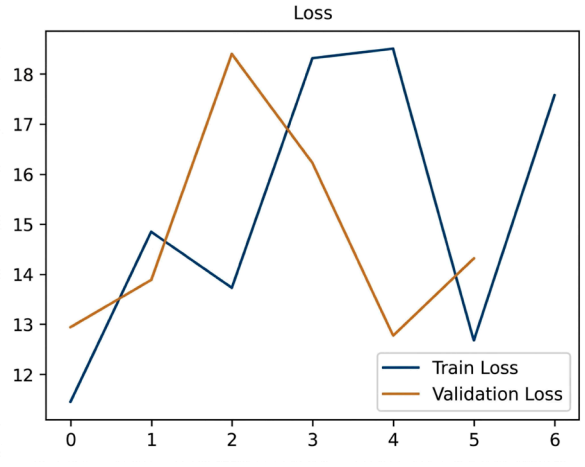

Fig. 61. Training and validation loss curves for fold 4 in iteration 3 using 5-fold cross-validation

In fold 4, the method also experienced learning failure, characterized by unstable loss values and the absence of a downward trend. The training and validation accuracy results are shown in Fig. 62.

Training and validation accuracy remained stagnant at low levels, indicating that the backpropagation method was unable to achieve effective learning. Fold 5 of 5, as shown in Fig. 63.

Fold 5 exhibits a failure pattern consistent with the other folds. The loss values fluctuate and do not converge. The training and validation accuracy results are shown in Fig. 64.

Accuracy does not improve over the course of the epoch. This indicates that the method is unable to adjust its parameters due to disturbances in the loss function.

The selection of the optimal cost vector was performed using grid search with k-fold cross-validation on three can-

didates. The results demonstrate that grid search 1 provides the most stable performance with consistent loss reduction and accuracy above 96% without overfitting. Grid search 2 also improves performance but with greater loss fluctuations, making it less stable. Meanwhile, grid search 3 failed due to the occurrence of NaN loss and low accuracy, so it was eliminated. Based on a comparison of validation loss, the best cost vector is found in iteration 1. The results of the grid search with cross-validation evaluation are shown in Fig. 65.

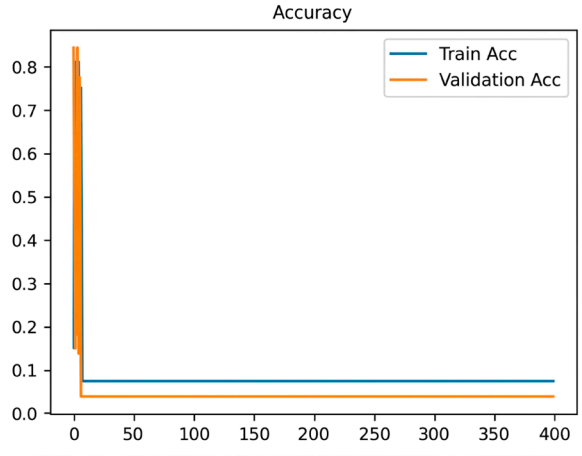

Fig. 62. Training and validation accuracy curves for fold 4 in iteration 3 using 5-fold cross-validation

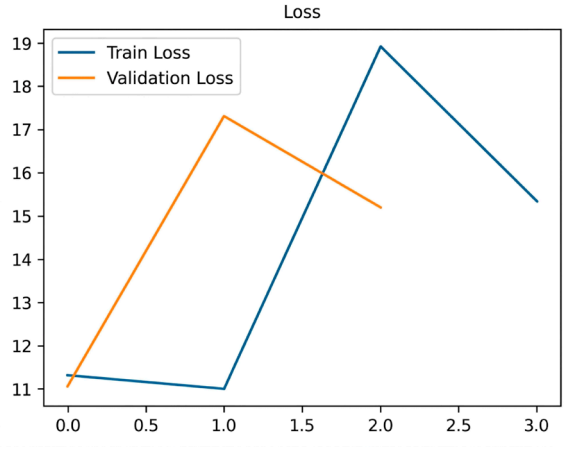

Fig. 63. Training and validation loss curves for fold 5 in iteration 3 using 5-fold cross-validation

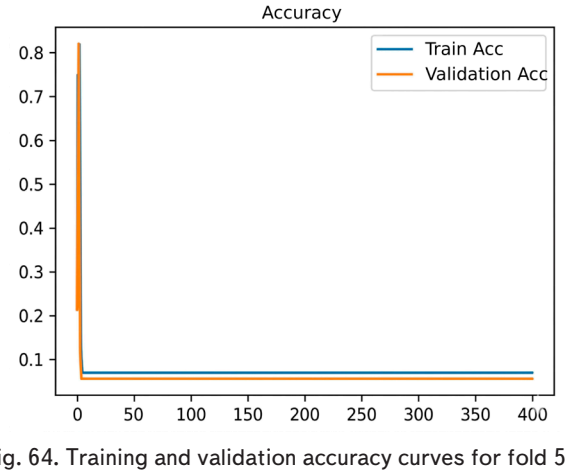

Fig. 64. Training and validation accuracy curves for fold 5 in iteration 3 using 5-fold cross-validation

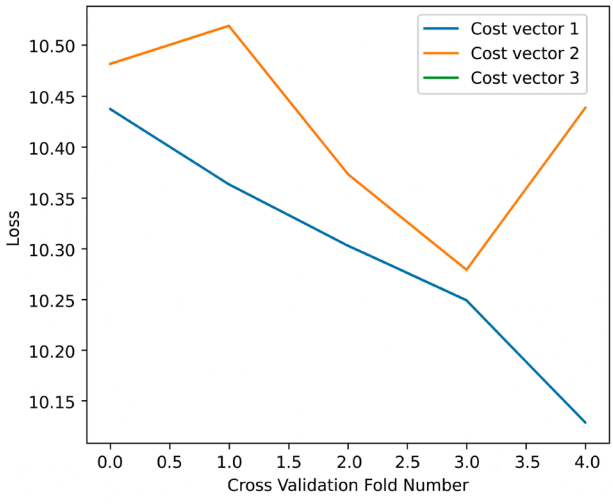

Fig. 65. Grid search with cross validation evaluation

Fig. 65 demonstrating a comparison of the average loss values resulting from k-fold cross-validation for three candidate cost vectors. Cost vector 1 produced the lowest and most stable validation loss value, while cost vector 2 showed greater fluctuations and cost vector 3 failed to produce a valid loss. Based on these results, cost vector 1 was selected as the optimal parameter in training the COST-ANN method.

5. 3. 2. Results of cost-sensitive artificial neural network with principal component analysis COST-ANN + PCA optimization

Iterative network weight updates using a cost-sensitive loss function. By integrating PCA with the COST-ANN optimization method, the method achieves faster and more stable convergence, as well as better generalization capabilities for new data.

Each output row demonstrating the progression of the train loss, test loss, train accuracy, and test accuracy values at each epoch as a concrete representation of the evaluation and parameter update steps [24], illustrated by the graph showing the shift in the COST-ANN method training results in Fig. 66.

Fig. 66 demonstrates the results of COST-ANN training using cost vector number 1 for 400 epochs. Training and testing accuracy increased rapidly in the early epochs to exceed 95% and then stabilized with a small difference, accompanied by a consistent decrease in loss. This pattern indicates good convergence without overfitting, so that cost vector number 1 produces effective learning and has excellent generalization capabilities, as shown in Fig. 67.

The results of the evaluation of the COST-ANN optimization method with PCA show significant and consistent classification performance, characterized by Sample-wise accuracy of 98.84%, area under the ROC curve (AUC) of 1.00, and F1-score of 0.99, as well as class-wise accuracy of 99.23% with recall of 0.988 and precision of 0.989. The confusion matrix shows that most classification are on the main diagonal, where the method was able to correctly classify 22 out of 23 GOOD class samples, 280 out of 282 MODERATE class samples, and 39 out of 40 UNHEALTHY class samples. This minimal classification error indicates that the method works accurately and evenly across all classes. Overall, these results prove that the integration of COST-ANN with PCA successfully produces a stable method that is not biased towards class imbalance and has excellent generalization capabilities.

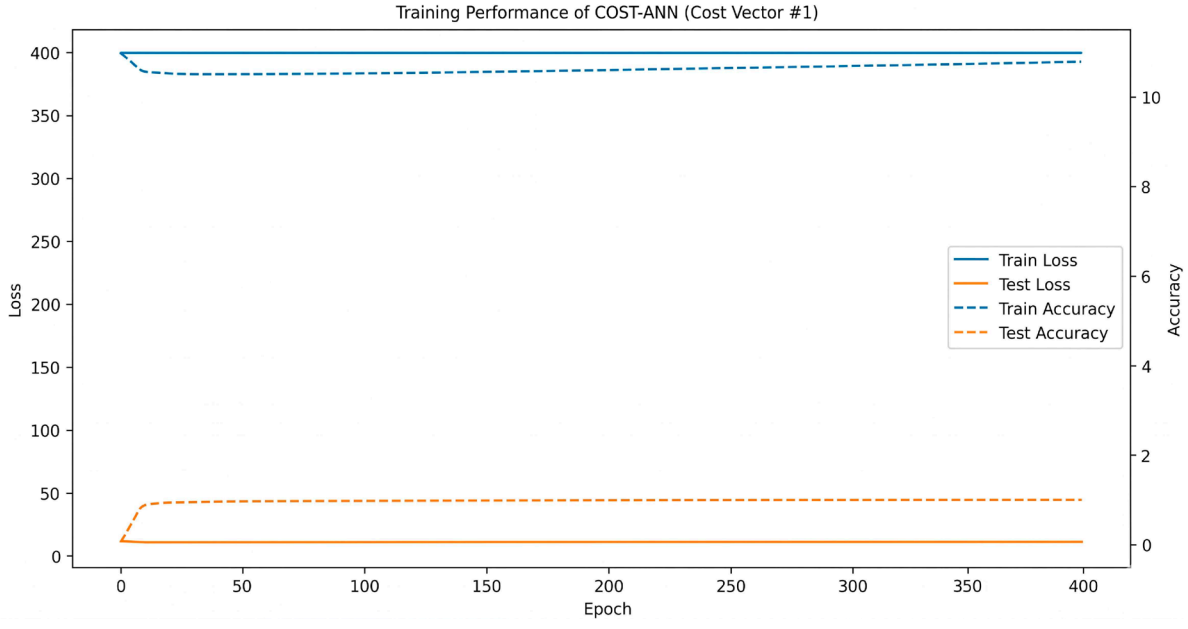

Fig. 66. Graph showing the development of the COST-ANN method training with Principal Component Analysis (PCA)

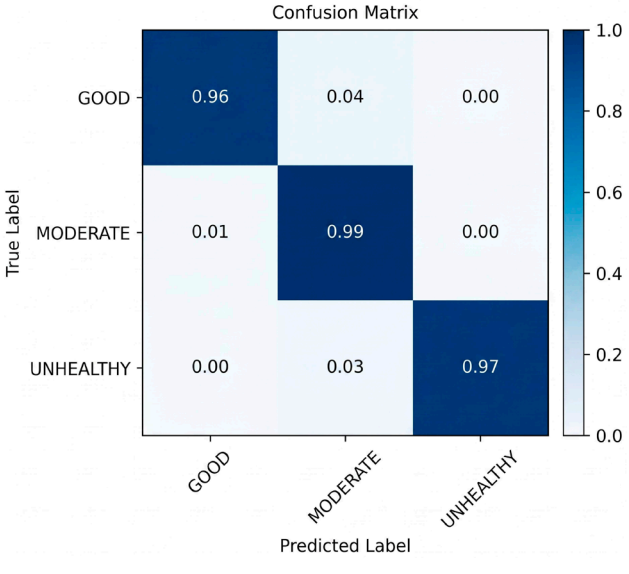

Fig. 67. Confusion matrix COST-ANN with Principal Component Analysis (PCA) and optimal cost dataset Air Pollution Standard Index (ISPU)

5. 4. Comparative evaluation of classification performance

After training, the method was evaluated on the ISPU dataset using the accuracy, precision, recall, F1-score, and AUC metrics to compare the performance of standard ANN and cost-sensitive ANN. This evaluation aimed to assess the effectiveness of the cost-sensitive method in improving classification quality and determining the most optimal method [9]. The results of the performance comparison between the two methods are presented in Table 2.

Table 2 shows a comparison of performance between the standard artificial neural network (ANN) method and the cost-sensitive artificial neural network (COST-ANN) based on five main evaluation metrics, namely accuracy, precision, recall, F1-score, and AUC. The test results show that the COST-ANN method consistently performs better than the standard ANN, with accuracy increasing from 97.97% to 98.26%, which means an increase in accuracy of 0.29%. In addition, precision increased from 98.01% to 98.30%, recall from 97.97% to 98.26%, and F1-score from 97.89% to 98.21%, indicating an overall improvement in classification quality. The AUC value also increased from 0.997140 to 0.998419, indicating that the method's discrimination ability for each class became more optimal.

A representation of the table that yields evaluation metrics commonly used in the performance analysis of classification methods on imbalanced data [25], as shown in Fig. 68.

The improvement in all of these metrics confirms that the application of cost-sensitive methods in ANN successfully reduces bias towards the majority class, increases sensitivity towards the minority class, and produces a classification method that is more balanced, stable, and reliable than standard ANN.

The comparative evaluation of ANN + PCA versus COST-ANN + PCA aims to assess the effectiveness of the cost-sensitive method in improving classification quality and to determine the most optimal method [9], as shown in Table 3 below.

Table 2

Comparison between ANN and COST-ANN

Method	Accuracy	Precision	Recall	F1-Score	AUC
ANN	0.979710	0.980139	0.979710	0.978951	0.997140
COST-ANN	0.982609	0.982966	0.982609	0.982158	0.998419

Table 3

Comparison of ANN + PCA with COST-ANN + PCA

Method	Accuracy	Precision	Recall	F1-Score	AUC
ANN + PCA	0.979710	0.979578	0.979710	0.979519	0.999006
COST-ANN + PCA	0.985507	0.985538	0.985507	0.985502	0.999100

Table 3 shows a comparison of the performance between the ANN + PCA and COST-ANN + PCA methods based on the metrics of accuracy, precision, recall, F1-score, and AUC. The evaluation results show that the application of a cost-sensitive method to PCA-based ANNs can consistently improve classification performance across all metrics. Accuracy and

F1-score increased from around 97.97% to 98.55%, with an accuracy increase of around 0.58%, while the AUC value also increased from 0.999006 to 0.999100.

Compared to previous studies discussed in Section 2, the proposed method demonstrates several advantages. Unlike methods that rely on resampling techniques such as SMOTE, this study preserves the original data distribution while improving classification balance. In contrast to methods that use static class weights, the proposed adaptive cost weighting method provides greater flexibility in handling varying levels of imbalance. Furthermore, while previous works often focus on binary classification or specific domains such as medical data, this study addresses a multi-class tabular dataset with a more generalized framework. These differences confirm that the proposed COST-ANN + PCA method offers a more robust and adaptable solution for imbalanced multi-class classification problems.

Table 3 presents the results of calculations for evaluation metrics commonly used in the performance analysis of classification methods on imbalanced data [25], which are then visualized in the form of a graph in Fig. 69.

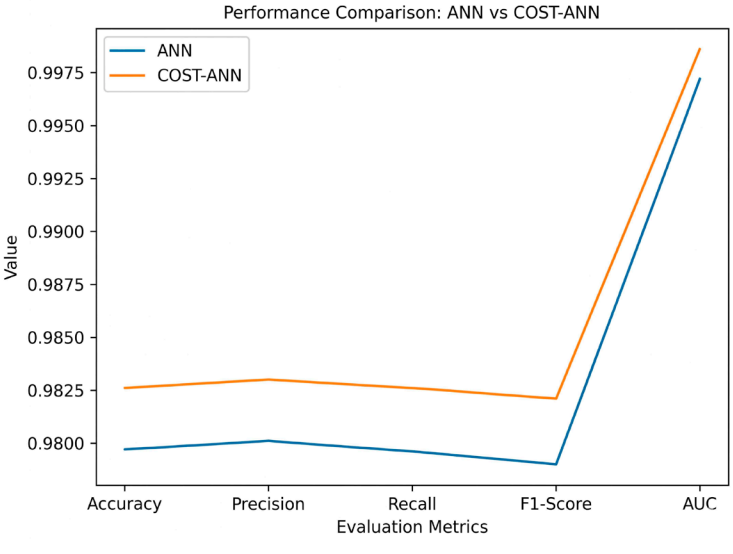

Fig. 68. Performance comparison chart between Artificial Neural Network ANN and Cost-Sensitive Artificial Neural Network COST-ANN

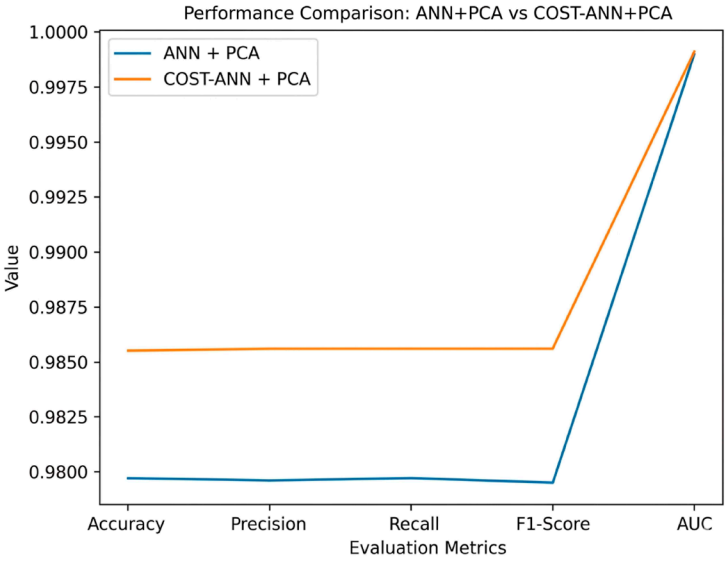

Fig. 69. Performance comparison chart of ANN + PCA vs COST-ANN + PCA

This improvement shows that weighting the optimal cost vector makes the method more sensitive to minority classes and reduces bias towards majority classes. Thus, the combination of PCA and COST-ANN produces a method that is more accurate, balanced, and has better generalization capabilities compared to ANN + PCA without cost optimization.

6. Discussion of cost-sensitive artificial neural network (COST-ANN) method results with principal component analysis (PCA) integration

The results obtained in this study are primarily attributed to the modification of the loss function in the COST-ANN method, specifically through the incorporation of class penalty weights as formulated in equations (22)–(25). This method assigns greater importance to errors in the minority class during the optimization process, thereby increasing the method's sensitivity to imbalanced data distributions. In addition, parameter optimization using grid search with k-fold cross-validation (Fig. 2–32) ensures that the selected cost vector represents the most stable configuration and yields the lowest validation loss. It should be noted that the optimal cost vector differs between the non-PCA and PCA scenarios due to differences in feature space representation. This difference indicates that the optimal cost configuration is highly dependent on the feature space representation, where the dimensionality reduction introduced by PCA alters the data distribution and consequently changes the optimal weighting scheme, leading to different convergence behaviors between the two scenarios. This is evidenced by the consistent convergence patterns observed in the best iterations, where the train loss and validation loss curves decrease in parallel and accuracy increases significantly across all folds. The integration of PCA, as formulated in equations (2)–(6), also contributes to improved method performance. PCA is capable of reducing the dimensionality and correlations among features, thereby making the learning process more stable and efficient. This is reflected in the improved performance of COST-ANN + PCA shown in Table 3 and Fig. 66, 67, where the method achieves an accuracy of 98.55% and an AUC approaching 1.00. Furthermore, the confusion matrix results (Fig. 34, 67) indicate that the method can classify all classes evenly, including the minority class, which was previously a major issue in conventional ANNs. Thus, the combination of cost-sensitive learning, cost vector optimization, and dimensionality reduction via PCA is the primary factor explaining the method's improved performance.

The proposed method has several key distinctions compared to previous studies. First, this study develops adaptive cost weighting as formulated in equations (23)–(25), which differs from previous method that generally use static weights such as inverse class frequency [35–38]. This method adjusts the weights based on the data proportion and the parameter λ, thereby providing greater flexibility in handling various levels of class imbalance. Second, this study systematically applies cost vector optimization using Grid Search combined with k-fold

cross-validation (Fig. 32, 65), unlike previous methods that use fixed weights [1], thereby yielding more robust and stable parameters. Third, the proposed method does not use data resampling techniques such as SMOTE or oversampling [15], thereby preserving the original data distribution and avoiding the risk of overfitting caused by synthetic data. Fourth, this study integrates principal component analysis (PCA) into a cost-sensitive learning framework, which has proven effective for multidimensional tabular data such as ISPU, unlike CNN-based method for time-series data [14]. Based on the comparison results in Tables 2, 3, the COST-ANN and COST-ANN + PCA methods consistently demonstrate better performance than conventional ANNs across all evaluation metrics (accuracy, precision, recall, F1-score, and AUC), indicating that the proposed method not only improves overall accuracy but also yields a better classification balance across classes.

Several inherent limitations of this study should be noted. First, the method is highly dependent on the data distribution, particularly in determining class weights within the adaptive cost weighting method equations (23)–(25). Significant changes in the data distribution can affect the cost vector values, thereby requiring a re-optimization process. Second, regarding generalization, the method was only tested on the ISPU dataset with tabular and numerical data characteristics, so further validation on cross-domain datasets is needed to ensure consistent performance. Third, the optimization process using grid search with k-fold cross-validation involves high computational complexity, particularly on large datasets or those with a greater number of parameters. Fourth, the proposed method still relies on determining the parameter λ as the weighting sensitivity controller, so that the selection of suboptimal parameter values can affect the stability and performance of the method. Fifth, the optimal results in this study were obtained within a specific range of data characteristics, so the effectiveness of the method on different types of data, such as time-series data or images, still requires further study.

Unlike the limitations, the weaknesses in this study are more related to aspects of implementation and method design. First, the parameter optimization process still uses Grid Search, which is discrete and less efficient in optimally exploring the parameter space, thus potentially failing to find the truly globally optimal cost vector combination; therefore, more advanced optimization method such as Bayesian optimization or genetic algorithms could be considered in future studies. Second, the cost vector method used remains static during the training process, so it cannot adapt to the dynamics of error distribution changes during the learning process, which may affect the method's sensitivity to minority classes at certain phases; the development of adaptive cost learning based on epochs or gradient feedback could be a solution to enhance the method's flexibility. Third, the artificial neural network (ANN) architecture used is still relatively simple, so the ability to represent complex non-linear features has not been fully utilized; integration with deeper deep learning architectures or hybrid methods has the potential to improve method performance, especially on data with higher complexity.

This study offers several opportunities for future development, accompanied by various challenges. First, the development of a dynamic adaptive cost learning method, in which cost weights are updated during the training process, has the potential to enhance method flexibility, but faces mathematical difficulties in maintaining gradient stability and loss function convergence. Second, integration with deep learning architectures such as convolutional neural networks (CNNs) or long short-term memory (LSTMs) opens up opportunities for application to spatial and time-series data, but poses methodological challenges in architectural design and complex parameter tuning. Third, application at the big data scale or in real-time systems requires high computational efficiency, thus facing experimental difficulties related to scalability, resource requirements, and computation time. Fourth, the development of theoretical studies regarding the convergence of loss functions in cost-sensitive method poses a significant mathematical challenge, as it involves formal proofs for weighted loss functions that are non-linear and complex. Fifth, validation on cross-domain datasets is crucial for testing method generalization, yet it faces methodological and experimental challenges stemming from differences in data distributions (domain shift), which can affect method stability and performance.

7. Conclusion

1. The proposed method introduces a cost-sensitive loss formulation that incorporates class-dependent penalty weights to address class imbalance. This formulation enables the method to assign higher importance to minority class errors, thereby reducing bias toward majority classes and improving classification balance.

2. The proposed method demonstrates stable and consistent learning behavior through the optimization process using grid search with cross-validation. The results show consistent convergence patterns, where both loss and accuracy remain stable across all folds. This indicates that the proposed method of handling varying levels of data imbalance while maintaining robust and reliable classification performance.

3. The integration of principal component analysis (PCA) with COST-ANN resulted in a more significant improvement in performance, with accuracy and F1-score reaching 98.55% and an AUC value of 0.9991. Compared to ANN + PCA, the COST-ANN + PCA method demonstrated an increase in accuracy of approximately 0.58% as well as improved method discrimination capability. These results indicate that dimensionality reduction via PCA enhances learning efficiency and method stability, particularly with multidimensional data, thereby leading to better generalization.

4. The comparative evaluation results show that the proposed method outperforms conventional artificial neural network methods in handling imbalanced multi-class classification problems. The proposed method superior performance across multiple evaluation metrics, with accuracy reaching 98.55%, precision 98.56%, recall 98.55%, F1-score 98.55%, and an AUC value of 0.9991. These results demonstrate that the proposed method is able to improve classification performance while maintaining balanced performance across all classes without modifying the original data distribution.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Data cannot be made available for reasons disclosed in the data availability statement.

Use of artificial intelligence

This study utilized artificial intelligence tools (ChatGPT, Grammarly) for language refinement, literature review, and code-writing assistance (Google Colab). All outputs were manually reviewed and verified by the authors to ensure their accuracy and reliability. The use of AI did not affect the scientific conclusions of this study.

Authors' contributions

Bosker Sinaga: Conceptualization, Methodology, Validation, Investigation, Formal analysis, Investigation, Resources, Data Curation, Writing – original draft, Visualization, Project management; **Yuhandri**: Conceptualization, Methodology, Validation, Writing – review & editing, Supervision; **Gunadi Widi Nurcahyo**: Conceptualization, Methodology, Validation, Writing – review & editing, Supervision.

References

1. Huang, W., Li, T., Liu, J., Xie, P., Du, S., Teng, F. (2021). An overview of air quality analysis by big data techniques: Monitoring, forecasting, and traceability. Information Fusion, 75, 28–40. https://doi.org/10.1016/j.inffus.2021.03.010
2. Karmoude, M., Munhungewarwa, B., Chiraira, I., Mckenzie, R., Kong, J., Smith, B. et al. (2025). Machine learning for air quality prediction and data analysis: Review on recent advancements, challenges, and outlooks. Science of the Total Environment, 1002, 180593. https://doi.org/10.1016/j.scitotenv.2025.180593
3. Ravindiran, G., Hayder, G., Kanagarathinam, K., Alagumalai, A., Sonne, C. (2023). Air quality prediction by machine learning models: A predictive study on the indian coastal city of Visakhapatnam. Chemosphere, 338, 139518. https://doi.org/10.1016/j.chemosphere.2023.139518
4. Fahim, A., Osman, A. M., Tarek, Z., Elshewey, A. M. (2025). Enhancing Air Quality Index Classification Based on Ensemble Machine Learning Techniques. Engineering, Technology & Applied Science Research, 15 (6), 29325–29333. https://doi.org/10.48084/etasr.13875
5. Labory, J., Njomgue-Fotso, E., Bottini, S. (2024). Benchmarking feature selection and feature extraction methods to improve the performances of machine-learning algorithms for patient classification using metabolomics biomedical data. Computational and Structural Biotechnology Journal, 23, 1274–1287. https://doi.org/10.1016/j.csbj.2024.03.016
6. Prael, F. J., Cox, J., Sturm, N., Kutchukian, P., Forrester, W. C., Michaud, G., Blank, J. et al. (2024). Machine learning proteochemometric models for Cereblon glue activity predictions. Artificial Intelligence in the Life Sciences, 6, 100100. https://doi.org/10.1016/j.ailsci.2024.100100
7. Buda, M., Maki, A., Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011
8. Abdelsattar Mohamed Saeed, M., Rasslan, A., Emad-Eldeen, A. (2024). Comparative Analysis of Machine Learning Techniques for Fault Detection in Solar Panel Systems. SVU-International Journal of Engineering Sciences and Applications, 5 (2), 140–152. https://doi.org/10.21608/svusrc.2024.279389.1198
9. Mienye, I. D., Sun, Y. (2021). Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. Informatics in Medicine Unlocked, 25, 100690. https://doi.org/10.1016/j.imu.2021.100690
10. Rezvani, S., Wang, X. (2023). A broad review on class imbalance learning techniques. Applied Soft Computing, 143, 110415. https://doi.org/10.1016/j.asoc.2023.110415
11. Pes, B., Lai, G. (2021). Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. PeerJ Computer Science, 7, e832. https://doi.org/10.7717/peerj-cs.832
12. Sangalli, S., Erdil, E., Hoetker, A., Donati, O., Konukoglu, E. (2021). Constrained Optimization to Train Neural Networks on Critical and Under-Represented Classes. arXiv. https://doi.org/10.48550/arXiv.2102.12894
13. Sleeman IV, W. C., Krawczyk, B. (2021). Multi-class imbalanced big data classification on Spark. Knowledge-Based Systems, 212, 106598. https://doi.org/10.1016/j.knosys.2020.106598
14. Zubair, M., Yoon, C. (2022). Cost-Sensitive Learning for Anomaly Detection in Imbalanced ECG Data Using Convolutional Neural Networks. Sensors, 22 (11), 4075. https://doi.org/10.3390/s22114075
15. Joloudari, J. H., Marefat, A., Nematollahi, M. A., Oyelere, S. S., Hussain, S. (2023). Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks. Applied Sciences, 13 (6), 4006. https://doi.org/10.3390/app13064006
16. Safi, S. A.-D., Castillo, P. A., Faris, H. (2022). Cost-Sensitive Metaheuristic Optimization-Based Neural Network with Ensemble Learning for Financial Distress Prediction. Applied Sciences, 12 (14), 6918. https://doi.org/10.3390/app12146918
17. Prasetiyowati, M. I., Maulidevi, N. U., Surendro, K. (2022). The accuracy of Random Forest performance can be improved by conducting a feature selection with a balancing strategy. PeerJ Computer Science, 8, e1041. https://doi.org/10.7717/peerj-cs.1041
18. Nath, D., Shahariar, G. M. (2023). Gastrointestinal Disease Classification through Explainable and Cost-Sensitive Deep Neural Networks with Supervised Contrastive Learning. arXiv. https://doi.org/10.48550/arXiv.2307.07603

19. Mustari, A., Ahmed, R., Tasnim, A., Juthi, J. S., Shahariar, G. M. (2023). Explainable Contrastive and Cost-Sensitive Learning for Cervical Cancer Classification. 2023 26th International Conference on Computer and Information Technology (ICCIT), 1–6. https://doi.org/10.1109/iccit60459.2023.10441352
20. Volk, O., Singer, G. (2024). An adaptive cost-sensitive learning approach in neural networks to minimize local training–test class distributions mismatch. Intelligent Systems with Applications, 21, 200316. https://doi.org/10.1016/j.iswa.2023.200316
21. Kim, Y.-S., Kim, M. K., Fu, N., Liu, J., Wang, J., Srebric, J. (2025). Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models. Sustainable Cities and Society, 118, 105570. https://doi.org/10.1016/j.scs.2024.105570
22. Manocchio, L. D., Layeghy, S., Gallagher, M., Portmann, M. (2025). An empirical evaluation of preprocessing methods for machine learning based network intrusion detection systems. Engineering Applications of Artificial Intelligence, 158, 111289. https://doi.org/10.1016/j.engappai.2025.111289
23. Bashir, R. N., Mzoughi, O., Shahid, M. A., Alturki, N., Saidani, O. (2024). "Principal Component Analysis (PCA) and feature importance-based dimension reduction for Reference Evapotranspiration (ET0) predictions of Taif, Saudi Arabia," Computers and Electronics in Agriculture, 222, 109036. https://doi.org/10.1016/j.compag.2024.109036
24. Razali, M. N., Arbaiy, N., Lin, P.-C., Ismail, S. (2025). Optimizing Multiclass Classification Using Convolutional Neural Networks with Class Weights and Early Stopping for Imbalanced Datasets. Electronics, 14 (4), 705. https://doi.org/10.3390/electronics14040705
25. Wang, Y., Rosli, M. M., Musa, N., Li, F. (2024). Multi-Class Imbalanced Data Classification: A Systematic Mapping Study. Engineering, Technology & Applied Science Research, 14 (3), 14183–14190. https://doi.org/10.48084/etasr.7206
26. Shoeibi, M., Nevisi, M. M. S., Salehi, R., Martín, D., Halimi, Z., Baniasadi, S. (2024). Enhancing Hyper-Spectral Image Classification with Reinforcement Learning and Advanced Multi-Objective Binary Grey Wolf Optimization. Computers, Materials & Continua, 79 (3), 3469–3493. https://doi.org/10.32604/cmc.2024.049847
27. Guo, Q., Wang, C., Xiao, D., Huang, Q. (2023). A novel multi-label pest image classifier using the modified Swin Transformer and soft binary cross entropy loss. Engineering Applications of Artificial Intelligence, 126, 107060. https://doi.org/10.1016/j.engappai.2023.107060
28. Zhao, Y. (2020). A note on new Bernstein-type inequalities for the log-likelihood function of Bernoulli variables. Statistics & Probability Letters, 163, 108779. https://doi.org/10.1016/j.spl.2020.108779
29. Narendran, A., Inuguri, A. H., Ravindra, A. R., Saga, H., C. S., V., Raj, R., B., K. (2025). Computational Approaches for Classifying Antimicrobial Peptides: A Comparative Analysis of BERT, Word2Vec, One-Hot Encoding, and Physicochemical Analysis. Procedia Computer Science, 258, 3019–3030. https://doi.org/10.1016/j.procs.2025.04.560
30. Yeung, M., Sala, E., Schönlieb, C.-B., Rundo, L. (2022). Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. Computerized Medical Imaging and Graphics, 95, 102026. https://doi.org/10.1016/j.compmedimag.2021.102026
31. Song, Z., Shi, Z., Yan, X., Zhang, B., Song, S., Tang, C. (2024). An Improved Weighted Cross-Entropy-Based Convolutional Neural Network for Auxiliary Diagnosis of Pneumonia. Electronics, 13 (15), 2929. https://doi.org/10.3390/electronics13152929
32. Han, X., Zhu, X., Pedrycz, W., Mostafa, A. M., Li, Z. (2024). A design of fuzzy rule-based classifier optimized through softmax function and information entropy. Applied Soft Computing, 156, 111498. https://doi.org/10.1016/j.asoc.2024.111498
33. Sun, Y., Zheng, J., Zhao, H., Zhou, H., Li, J., Li, F. et al. (2024). Modifying the one-hot encoding technique can enhance the adversarial robustness of the visual model for symbol recognition. Expert Systems with Applications, 250, 123751. https://doi.org/10.1016/j.eswa.2024.123751
34. Polat, G., Çağlar, Ü. M., Temizel, A. (2025). Class distance weighted cross entropy loss for classification of disease severity. Expert Systems with Applications, 269, 126372. https://doi.org/10.1016/j.eswa.2024.126372
35. He, H., Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 21 (9), 1263–1284. https://doi.org/10.1109/tkde.2008.239
36. Ling, C. X., Sheng, V. S. (2008). Cost-Sensitive Learning and the Class Imbalance Problem. Encyclopedia of Machine Learning. Available at: https://www.researchgate.net/publication/268201268_Cost-Sensitive_Learning_and_the_Class_Imbalance_Problem
37. Johnson, J. M., Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. Journal of Big Data, 6 (1). https://doi.org/10.1186/s40537-019-0192-5
38. Cui, Y., Jia, M., Lin, T.-Y., Song, Y., Belongie, S. (2019). Class-Balanced Loss Based on Effective Number of Samples. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9260–9269. https://doi.org/10.1109/cvpr.2019.00949