Received: June 30, 2023. Revised: August 16, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

The Performance Comparison between C4.5 Tree and One-Dimensional Convolutional Neural Networks (CNN1D) with Tuning Hyperparameters for the Classification of Imbalanced Medical Data

Agus Wahyu Widodo[1] **Samingun Handoyo[2,3]*** **Irfany Rupiwardani[4]**
Yusi Tyroni Mursityo[5] **Imam Nurhadi Purwanto[6]** **Heni Kusdarwati[3]**

[1]*Informatics Engineering Department Brawijaya University, Malang, Indonesia*
[2]*Electrical Engineering and Computer Science –IGP Department, National Yang-Ming Chiao Tung University, Hsinchu, Taiwan*
[3]*Statistics Department Brawijaya University, Malang, Indonesia*
[4]*Department of Environmental Health School of Health Science, Widyagama Husada, Malang, Indonesia*
[5]*Information System Department Brawijaya University, Malang, Indonesia*
[6]*Mathematics Department Brawijaya University, Malang, Indonesia*
* Corresponding author's Email: samistat@ub.ac.id

Abstract: The implementation of a CNN1D model is a big challenge because it involves many hyperparameters which have to be tuned well. The research has goals to compare the classification performance of the C4.5 tree and CNN1D models applied in the imbalanced medical dataset in six performance metrics. The oversampling method is applied to the training dataset to form the oversampling training data which is divided into 5 folds of cross-validation data for tuning hyperparameters of both models. By using the grid search method, there are obtained the optimal CNN1D hyperparameters are [256, 4] for the filter and kernel sizes, and the optimal C4.5 hyperparameters are [25,7] for the minimum number of instances in the splitting node and the depth of C4.5 tree. The models with optimal hyperparameters are trained using the oversampling training data and evaluated their performance in both of original training and testing datasets. In the training dataset, the CNN1D outperforms the C4.5 tree in all six metrics with a value of 99% except in the matthews correlation coefficient (MCC) metric with 73% besides the performance gaps of both models being very large. In the testing dataset, the CNN1D also outperforms the C4.5 tree in four metrics with a value of around 99%, but in the MCC and the area under curve (AUC) metrics, the C4.5 tree outperforms the CNN1D although their performance gaps are narrow enough. An interesting future work is to use the MCC or AUC criteria in the tuning hyperparameters of models for the classification of imbalanced class datasets.

Keywords: Convolution neural networks, Filter and kernel sizes, Hyperparameters tuning, Imbalanced class, MCC and AUC metrics.

1. Introduction

Deployment of convolution neural networks (CNN) models as a classifier model of images had reached a peak popularity in the fewer last few decades. The existence of convolutional layers that had a role in mapping features supporting the model can be more understood in the image classification stage. Even to increase model performance, Hassanzadeh et al [1] hybridized the genetic algorithm and CNN forming a new model which had outperformed other models when it was trained in 5 datasets. While Yuan et al [2] fully utilized the advantages of transformers and convolutional neural networks (CTC-Net) to form a hybrid model that had a high performance in image segmentation. The implementation of CNN for classifying instances researchers including Habite et al [3] used CNN1D to determine path location along Norway spruce timber boards by generating thousands of virtual timber

Received: June 30, 2023. Revised: August 16, 2023.

boards. Recently, Venturini et al [4] applied the CNN1D on a single-fluorescence spectra for predicting five chemical quality indicators of olive oil. The CNN1D model in both kinds of research has a very satisfactory performance. On another side, an assembling method called the C4.5 tree had been applied successfully a long time ago to classify instances with very satisfactory performance such as shown by Wati et al [5] and Gite [6] recently. Both CNN1D and C4.5 models are categorized as complex models that involve many hyperparameters which have a large influence on model performance. Tuning hyperparameters of complex models is not an easy task and must be addressed fairly ways. Tuning hyperparameters as done in [7, 8] was a trade-off that must be paid to obtain a satisfactory performance of a complex model including the model of CNN1D and the C4.5 tree.

All decision-making processes always can be brought into a binary choice that supposes to make easy decision-makers jobs. The right decision choice from a set of decision candidates will lead to a wise policy. Some real-world applications of binary choice of decisions include a financial analyst evaluating a company status that was categorized as a fraudulent or not fraudulent firm based on many features of the company profile [9, 10], a nutritionist making a decision regarding the status of a baby categorized as stunting or normal [11], also a midwife deciding what a pregnant mother will birth through surgery or normally [12]. Those researchers used various machine learning models including the C4.5 tree and CNN1D models with satisfactory performance. Decision-making in the health area is not only a crucial but also critical task because it relates directly to the survival of human life. In the last decades, hospitals have collected their medical data well so it supports developers of machine learning models with a huge dataset. Unfortunately, many cases of medical data have imbalanced classes such as in Liu et al [13], Mienye and Sun [14], and Alabbad et al [15] where an additional treatment must be given to the imbalanced data before they are used to train machine learning models. The oversampling method is one of the techniques to address imbalanced class problems which should be conducted to the training set to yield oversampling training data as done by Kovács [16], and also by Wibowo & Fatichah [17]. Furthermore, the selected model namely the one having optimal hyperparameters must be trained by using the oversampling training data.

The research goals are to implement both models of the CNN1D and C4.5 tree for the classification of imbalanced medical data and compare their performances in the six metrics where their optimal hyperparameters are obtained through 5 folds cross-validation of the oversampling training data. In each of the models, it is determined 2 kinds of hyperparameters which will be tuned by the grid search algorithm. The comparison of performance metrics is carried out in both the original training and testing data. The rest of the article is structured as outlined in the following: Section 2 provides an overview of pertinent literature in the field. In section 3, we delve into a detailed examination of a medical dataset used as a case study, accompanied by an explanation of the proposed methodology. Section 4 encompasses the presentation and discussion of our results, including the process of model development, and performance evaluation across six metrics. Concluding remarks and recommendations for future work are presented in section 5.

2. Related works

Today, most aspects of human life are supported by artificial intelligence and machine learning methods are most core models implemented there. The existence of the target feature in a dataset will lead to the type of appropriate machine learning model. If a feature target is not available, descriptive models can be built. Some popular descriptive models include clustering methods such as those implemented in [18, 19] and ranking methods such as those implemented in [20, 21]. Descriptive models do not have a standard evaluation metric so they are impossible to obtain the best one. On the other hand, if a feature target is available, predictive models can be developed whether regression models such as in [22, 23] or classification models such as in [24] depending on the scale measure of the target feature. Complex classification models including decision trees and neural networks have proven very satisfactory performance when they were compared with most other machine learning models [25, 26]. Nevertheless, the complicated models tend to suffer the overfitting problem and have various numbers of hyperparameters that need to be tuned in a systematic way. For example, decision tree hyperparameters include the tree depth, many instances in the splitting node, pruning criteria, and so on. The neural network hyperparameters include network architecture, hidden layer depth, learning algorithm, and so on.

Deployment of the C4.5 tree that was implemented in real-world applications is very popular with satisfactory performance. The model is many applied both in engineering and public health or medical fields because it permits not only categorical but also numerical attributes, deals with missing values, and applies pruning to avoid overfitting. The C4.5 tree had been applied at the planning stage to form rule bases in predicting household electricity consumption in Maluku by Tempola et al [27]. Using the C4.5 tress for Security classifications of power system operating states was achieved under vast load variations was done by Jain et al [28]. The C4.5 trees-based islanding technique was used to reduce the chip's peak and average temperatures, voltage drop, area, and wire length done by Shanthi et al [29]. The C4.5 implementations in those fields had given state-of-the-art performance. In another field, Peng et al [30] proposed C5.0 to forecast aggravation risk in patients with acute exacerbation of chronic obstructive pulmonary disease with 28 features including vital signs, medical history, comorbidities, and various inflammatory indicators selected. Jaiswal & Kumar [31] proposed Enlarge C4.5 by pre-processing the Breast cancer dataset using stochastic self-organizing map. Hamdi et al [32] combined the Gaussian mixture model and the C4.5 tree method were used to classify and predict new cases of the COVID-19 virus. The C4.5 tree and its derived model in the above research outperformed the benchmark models used in at least 2 accuracy metrics. It means enough reasons for implementing the C4.5 tree model for classifying the medical data. However, most of the C4.5 tree implementations did not conduct the hyperparameters tuning systematically. The hyperparameter values were set up by trial and error so it was not guaranteed that they were the optimal one.

The CNNs model is a generation of neural networks that had obtained more attention in the last decade because it had been implemented successfully for image classification with very satisfactory performance [33]. The wider deployment of CNN for one-dimension data classification has brought into the model called CNN1D and also has attracted many researchers. Zhang et al [34] compared 9 deep neural networks including CNN1D to forecast hourly bus passenger flow in China. Su et al [35] used a sliding window in CNN1D to learn the local information of the patient record and capture the dependence between patients and medications from both global and local levels. Landolsi et al [36] merged 3 text features which are formatting style, syntactic, and semantic features to train the CNN1D for generating the document title. Those implementing CNN1D in various fields above had yielded satisfactory performance compared to the other machine learning models used as the benchmark. However, deep neural networks model including CNN suffered from the problem of long-term dependencies which led to their feature extraction ability defect [37]. While AL-Alimi, et al [38] conducted concatenating and training a hybrid multi-size CNN kernel using the meta-learner technique to train multi-class and multi-size datasets to produce better models. Two types of important hyperparameters of the CNN model are the filter and kernel sizes. Tuning both hyperparameters in k folds cross-validation data ensured obtaining the optimal pairs of them fairly.

Numerous researchers, including Liu et al [13], have explored classification techniques for medical datasets with imbalanced data. One of their approaches involved a combination of random forest (RF) and deep neural networks (DNN). In their study, they utilized random forest regression to handle missing values in the data prior to classification. Furthermore, they employed an automated hyperparameter optimization (AutoHPO) technique using a deep neural network for enhancing stroke prediction. Mienye and Sun [14] introduced a method of addressing class imbalance called cost-sensitive learning. This technique, as opposed to resampling methods, involves adjusting the objective functions of machine learning algorithms. Their approach doesn't require altering the distribution of the imbalanced class during training, leading to more dependable performance compared to data resampling. The method combining the Synthetic minority oversampling technique (SMOTE) and the k-means algorithm to address the imbalanced class problem was proposed by Xu, et al [15]. The approach had increased significantly the C 4.5 tree performance in both the Sensitivity and Specificity indexes of accuracy metrics. Although there are many methods to address the imbalanced class, the implementation to develop the CNN1D model will lead to another problem. In the research, the oversampling method is used to resolve the imbalance class where the method is applied to the training data to form the oversampling training data which will be used to tune hyperparameters and to train the models with the optimal ones.

3. Dataset and proposed method

The dataset was taken from a Taiwan hospital in 2021. It consisted of 52159 records of the joint replacement surgery patient. The dataset is available as public data provided by the hospital that can be found at https://github.com/saminghan/SDAI_HW-3/blob/main/cnn_train.csv. The target feature called the "outcome" feature had a binary value namely not infected or infected that it was supposed to be affected by 29 features consisting of both categorical and numerical features. Table 1 presents a list of both types of predictor features.

Table 1. List of predictor feature names

The 17 categorical features	The 12 numerical features
"Psychoses", "Alcohol Abuse", "Coagulopathy", "Solid Tumor without Metastasis", "Peptic Ulcer Disease excluding bleeding", "Rheumatoid Arthritis/collagen Cancer history", "Diagnosis", "elx_index", "cci_index", "Commercial_ALBC", "Drain", "Paralysis", "SEX", "Non_commercial_ALBC", "Anemia", "Psyciatric disorder"	"CBC_Platelet", "AGE", "length of stays (LOS)", "OP_time_minute", "OP_time_hour", " Irregular heartbeat (BUN)", " Heart disease (GOT)", " Pulmonary Circulation Disorders (GPT)", " Peripheral Vascular Disorders (ALB)", " Hypertension Uncomplicated (Na)", " Paralysis (K)", " Neurological Disorders (UA)"

The target feature of the dataset has a name of "outcome" that excludes in Table 1. The "outcome" feature has a distribution of the imbalance class, which is [not infected(class_0), infected(class_1)] = [51280, 879]. The number of instances in class_0 is 98%, which means only by guessing that the probability of an instance really comes from class_0 which is 98% correct. The dataset contains 17 categorical predictor features that have various label numbers which are 11 features with 2 labels, and 2 features with 3 labels, while the remaining features have respectively 4, 6, 14, and 16 label categories. The second column of Table 1 presents 12 numerical features, which have not only various measurement units but also very heterogeneous variances. The preprocessing was done to make commensurate measures of numerical features which must be transformed into [0,1] range by using a min-max transformation.

Two model types will be developed in the study, namely the C4.5 tree and CNN1D classification models. The research focuses on the development of the CNN1D model while the implementation of the C4.5 tree has a role as a benchmark model. The stages of developing models are presented in the sequential process in Fig. 1. In essence, Fig. 1 consists of 4 main processes namely the oversampling to overcome imbalanced class, tuning hyperparameters by using 5 folds cross-validation, training models with the optimal hyperparameters by using the oversampling training data, and evaluating of model performances on both the original training and testing dataset in six performance metrics.

3.1 Tuning hyperparameters

Most machine learning models involve hyperparameters which are supposed to be set up carefully in fairways [39]. The k folds cross-validation method is a popular method to obtain objectively optimal hyperparameters that direct the machine learning model to have the best performance [40]. Fig. 2 describes how to divide the training oversampling data into 5 pairs of training and validation folds. In the beginning, the dataset is divided into the training data which is a large part of the dataset where it is usually 80% and the 20% remaining is the testing data.

In Fig. 2, Firstly, the training oversampling data is divided into 5 folds with almost the same size. Each fold has a chance to be a validation fold. In detail, the process to arrange the training and validation folds based on 5 folds is given in the following summary:

a. Pick the first fold as the validation fold and the remaining folds as the training folds.
b. Pick the second fold as the validation fold and 4 other folds as the training folds.
c. Pick the third fold as the validation fold and 4 other folds as the training folds, and so on.
d. Finally, pick the fifth fold as the validation fold and the 4 previous folds as the training folds.

Model candidates with each pair of hyperparameters are trained on all of the training folds and their performances are evaluated on the corresponding validation fold. The pair of hyperparameters that yield the best average accuracy on the 5 validation folds is picked up as the optimal one [40].

3.2 C4.5 tree model

The C4.5 tree model is a classification model which is the best-known and most widely implemented in real-world applications. Some advantages of the C4.5 tree are to permit numeric attributes, deal with missing values, and apply pruning to avoid overfitting [41]. Numerical attributes are treated through a binary split on a chosen cut-point. The way to find it is by evaluating info gain (or other measures) for every possible cut-point of the attribute and selecting the cut-point having the highest info gain. The C4.5 tree deals with missing values by using the simple method (treat a missing value as a separate value) or using the advanced method which means each data point is associated with a weight in training and splits a data

Received: June 30, 2023.   Revised: August 16, 2023.

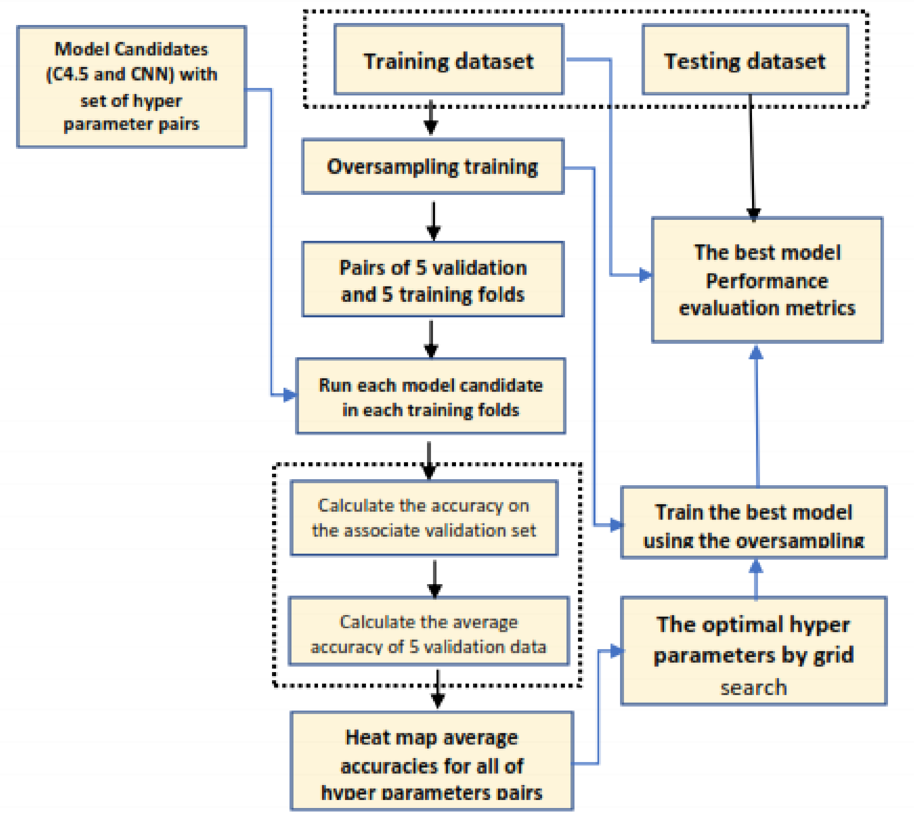

Figure. 1 The schema of computation stages in developing the C4.5 tree and CNN1D classification model

The training dataset				
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Val. 1	The training folds			
	Val. 2			
		Val. 3		
			Val. 4	
The training folds				Val. 5

Figure. 2 Divide the oversampling training data into 5 pairs of validation and training folds

point with missing values into pieces. Furthermore, the final classification step is done by considering the weights associated with each branch. The advanced method is conducted when the simple method faces a problem in resolving missing values. Two strategies for pruning the C4.5 tree are pre-pruning (stop growing a branch when information becomes unreliable) and post-pruning (take a fully-grown decision tree and discard unreliable parts) [42].

The post-pruning is done by using an error estimate. The C4.5 tree attempts to make the error estimate based on the training data itself. The idea is to consider a set of instances reaching each node and imagine that a majority class is chosen to represent that node. It's a heuristic based on some statistical reasoning that seems to work well in practice. The mathematical formula involved is given in Eq. 1. The default of a particular confidence c used by C4.5 is c = 25%. The task is to find confidence limits z such that satisfies Eq. (1) as the following.

$$Pr\left[\frac{f-q}{\sqrt{q(1-q)/N}} > z\right] = c \tag{1}$$

where N, f = E/N, q respectively is the number of samples, the observed error rate, and the true error rate. It leads to an upper confidence limit for q that upper confidence limit as a (pessimistic) estimate for the error rate e on the node is given in Eq. (2) as follows.

$$e = \frac{f+\frac{z^2}{2N}+z\sqrt{\frac{f}{N}-\frac{f^2}{N}+\frac{z^2}{4N^2}}}{1+\frac{z^2}{N}} \tag{2}$$

The + sign before the square root in the numerator of Eq. (2) is used to obtain the upper confidence limit where the z is the number of standard deviations corresponding to the confidence c, which for c = 25% is z = 0.69. The C4.5 tree uses a default confidence value set at 25% and works fairly well in most cases; more drastic pruning will occur if the default value is set to less than 25% and the true error rate of the pruned trees in the test dataset is much higher than

 DOI: 10.22266/ijies2023.1031.63

Received: June 30, 2023. Revised: August 16, 2023.

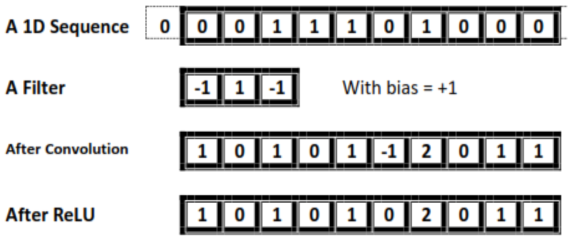

Figure. 3 A convolution layer computation involving the ReLU activation

the estimated error rate [43].

3.3 CCN1D classification model

CNNs are a special family of neural networks that contain convolutional layers that have a role as a feature map through a pooling layer. Two characteristics of CNNs are translation invariance in images which implies all patches of an image will be treated in the same manner and locality which means only a small neighbourhood of pixels will be used to compute the corresponding hidden representations [44]. A convolutional layer conducts the dot product (cross-correlation) between the input and kernel containing a bias that will be added to the result of the dot product to produce an output. The two kinds of parameter sources of a convolutional layer are the kernel and associated bias. The kernel contains weights as many as the kernel size and an associated scalar bias. When training a model based on the convolutional layer, the kernel weights typically initialize randomly, and they update through the backpropagation algorithm by minimizing the loss function by using an optimization method such as gradient descent [45]. The loss function for binary classification usually uses the cross entropy (CE) which is given in Eq. (3) as the following [46].

$$CE = -\sum_x P(x) * \log(1 - P(x) \tag{3}$$

where x and P(x) respectively represent an instance and the probability that the instance x is classified correctly.

Fig. 3 shows the process of the calculated output of the convolution layer involving a filter with a bias and RelU activation function. The filter size is 3 of [-1,1,-1] with a bias of +1. The raw named after convolution is obtained by operating the dot product between the filter and the part of the corresponding sequence, and then adding the bias value to get the convolution output. The filter is moved from the first part to the second part of the sequence using 1 stride. The first and last elements with break lines show the same padding. The last row of Fig. 3 is obtained by forwarding the convolution output to the ReLU activation function.

Table 2. Elements of a confusion matrix

Actual Class	Predicted class	
	Class_0	Class_1
Class_0	True Negative(TN)	False Negative (FN)
Class_1	False Positive (FP)	True Positive (TP)

3.4 Classification performance metrics

In general, performance metrics of a classification model are calculated by using the elements of the confusion matrix given in Table 2 as the following [47].

The MCC (matthews correlation coefficient) metric is widely used in the evaluation of a classification model performance in biomedical research and it is calculated based on elements of Table 2. On the other hand, a metric called AUC (area under ROC curve) is calculated by using a numerical integration approach. The AUC had shown as an elective metric to obtain a consensus on the best practices for the development and validation of predictive models in the medical field [48]. Both MCC and AUC have a range value between 0.0 and 1.0 representing a binary classifier performance for separating instances of the positive class (class_1) from instances of the negative class (class_0). While the formula of both metrics is given in Eqs. (4) and (5) as the following

$$\boldsymbol{MCC} = \frac{\boldsymbol{a}}{\sqrt{\boldsymbol{b}}} \tag{4}$$

Where $a = TP \times TN - FP \times FN$, and
$b = (TP + FP)(TP + FN)(TN + FP)(TN + FN)$

$$\boldsymbol{AUC} = \int_0^1 \boldsymbol{ROC(x)dx} \tag{5}$$

Other kinds of performance metrics which are very popular for evaluating classifier models are given in Eqs. (6) to (9) namely accuracy, precision, recall, and F1 score. These metrics except the F1 score cannot measure well classifier model performance of imbalanced class data. It means a classifier model having a high value of Accuracy, precision, and recall does not ensure that it is a good model [49]. Because of the issue, other performance metrics namely MCC and AUC which are talked about previously are used to complete for evaluating classifier models' performance.

$$\boldsymbol{Accuracy} = \frac{(\boldsymbol{TP+TN})}{(\boldsymbol{TP+TN+FP+FN})} \tag{6}$$

 DOI: 10.22266/ijies2023.1031.63

$$\textit{Precision} = \frac{TP}{(TP+FP)} \quad (7)$$

$$\textit{Recall} = \frac{TP}{(TP+FN)} \quad (8)$$

$$\textit{F1 score} = \frac{2 \times Recall \times Precision}{(Recall+Precision)} \quad (9)$$

4. Results and discussion

In the section, some results for the developing stages of both the C4.5 tree and CNN1D models will be presented sequentially in detail. Firstly, the issue of the imbalanced class is resolved by conducting the oversampling. The second results are both heat maps in tuning hyperparameters of models by using 5 folds cross-validation created from the oversampling training data. The last results are comparisons of model performance in both the original training and testing data in six metrics of performance.

4.1 Oversampling to the training dataset and formatting 5 folds for cross-validation

An imbalanced class will lead to the classification model yielding bad performance when it is used to predict an instance from the minority class. The over-sampling technique has the goal to increase the number of instances in the minority class label to be the same class member as the majority class label [50]. The method took instances in the minority class randomly with replacement until both classes have the same number of elements. The over-sampling was conducted in the training set and the resulting over-sampling data will be used to train both C4.5 and CNN1D classification models.

Complicated models such as the ensemble model (C4.5) or CNN1D model need to set up their hyperparameters. To make fairness in the tuning hyperparameters, the research uses the grid search with cross-validation data [51]. There are 2 hyperparameters considered in both models namely for the C4.5 model is depth tree and a minimum number of instances in a splitting node and for the CNN1D model is the kernel and filter sizes.

The cross-validation data is arranged from the over-sampling data divided into 5 folds. The candidate model was trained by using the training fold (formed by 4-folds) and the performance was evaluated with the remaining 1 fold which is called the validation data. Each fold has the opportunity to be the validation data, so there are 5 validation data and 5 training folds. Each candidate model with pairs of hyperparameters will be trained 5 times and its performance will be measured 5 times too, the average accuracy of the candidate model in 5 validation data will be used in the grid search method.

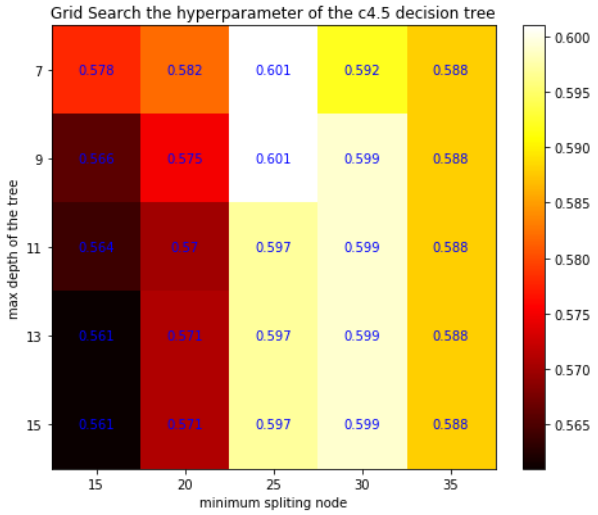

Figure. 4 The heat map of average accuracies in both of C4.5. hyperparameters

4.2 The C4.5 tree hyperparameters tuning

The C4.5 trees hyperparameters considered that suppose to have a great influence on the performance are the minimum instances of the splitting nodes and the deep maximum of the C4.5 trees that are respectively [15, 20, 25, 30, 35] and [7, 9, 11, 13, 15]. Each of all combination pairs (25 pairs) has the role as a candidate of the optimal hyperparameters to train the C4.5 tree on training oversampling folds and compute the accuracy performance on the associated oversampling validation fold. Because there are 5 possible training oversampling folds and 5 possible associated validation folds that meant each pair of hyperparameters is used 5 times to train the C4.5 tree and each model yielded will be evaluated in its accuracy performance using the associated validation fold. Furthermore, the average accuracy of the model on 5 validation folds is calculated. by averaging the accuracy of model performance on each pair of hyperparameters. The average accuracy of 25 pairs of hyperparameters combination is presented as a heat map in Fig. 4.

Based on Fig. 4, The result of the grid search method in the finding of the optimal hyperparameters of the C4.5 tree gives that the pairs of [25,6] and [25,7] produce the highest average accuracy performance which is 0.601. One of them which is the pair of [25,7] was picked up for training the C4.5 tree by using the oversampling training data and evaluating the model performances on both the training (80% dataset) and testing sets (20% dataset). After fitting the model with the optimal hyperparameters, firstly, the confusion matrices on both the training and testing sets have to be calculated because they are the raw material for obtaining all of the performance metrics. Table 3 presents confusion matrices on both the training and testing sets of the best C4.5 tree.

4.3 The CNN1D hyperparameters tuning

Developing the CNN1D model with the optimal filter and kernel sizes is started by determining some values as the optimal hyperparameters candidate. Hyperparameters of the filter and kernel sizes that are tuned by 5 folds cross-validation are [8, 16, 32, 64, 128, 256] and [2, 3, 4, 5, 6, 7] for possible values of filter and kernel sizes respectively. The kernel size values are selected by the following arithmetic sequence with 8 differences but the filter sizes follow the same sequence with 1 difference. Each combination pair of them (36 pairs) will be used to train the CNN1D in the training oversampling folds (union of 4 folds) and calculate the accuracy performance on the associated validation fold. The average accuracy performance in all folds and pairs of filter and kernel values is presented as a heat map in Fig. 5.

Fig. 5 shows that the average accuracy values increase gradually in the ordinate of filter size and decrease gradually in the axis of kernel size. The increasing of filter size value made the increasing the average accuracy, but the increasing the kernel size value made the decreasing the average accuracy. The best pair of the filter and kernel sizes are 4 and 256 where the average accuracy of the CNN1D model is 0.949, and there were some hyperparameter pairs having the average accuracy performance which was not a significant difference. Furthermore, the CNN1D with the hyperparameters pair of filter and kernel sizes of [256, 4] is trained by using the full oversampling training data. Both the confusion matrix and model performance will be presented in the next section for both the original training and testing sets.

4.4 The C4.5 tree and CNN1D performance

The C4.5 tree with [25,7] hyperparameters of the min. the number of instances on the splitting mode and the max. deep of tree has the confusion matrix in the first part of Table 3, while the second one is the confusion matrix of the CNN1D with [256,4] hyperparameters of the filter and kernel sizes number. Both models are evaluated on the training and testing dataset where the training dataset consists of 36766 instances and the testing dataset consists of 9192 instances. In the training dataset, the C4.5 tree classifies correctly 28345 instances but the CNN1D classifies correctly 35637 instances of class 0. While

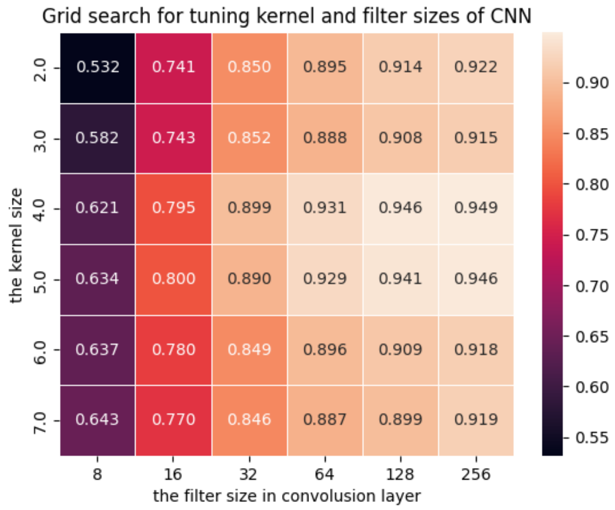

Figure. 5 The heat map of CNN1D average accuracies in various values of kernel and filter sizes

class 1 instances are classified correctly as 413 and 611 instances by the C4.5 tree and CNN1D respectively. The result shows that the CNN1D performance is better than the C4.5 tree. On the other hand, in the testing dataset, as many as 7051 and 8753 instances of class 0 are classified correctly by the C4.5 tree and the CNN1D respectively. While the class 1 instances, the C4.5 tree classifies correctly 73 instances, and the CNN1D classifies correctly 8 instances. The result is slightly different from the previous one in the training dataset. The comparative performance of both models in six metrics is presented in Table 4 as the following.

Table 4 presents the performance of the best C4.5 tree as a benchmark model and the best CNN1D in both the original training and testing data. The first column is the performance metrics names used for model evaluation, the second column contains the C4.5 tree and CNN1D performance on the original training data, and the last one is column contains both models' performance on the original testing data. Firstly, concerning performance metrics in the original training data, the C4.5 tree has a lower performance than the CNN1D in all of six metrics. The performance gap between both models is a very large of significant difference. All performance metrics of CNN1D are around 99% except the MCC metrics of around 73% while the C4.5 tree has moderate performance in metrics of accuracy, recall, and AUC which are 78.22%, 73%, and 72.99% respectively, and other metrics have values of 52%, 48%, and 14.11% respectively for metrics of precision, F1 score, and MCC. It should be noticed that the original training data here is an imbalanced class, 80% part of the dataset, which is not the data which used to train the models.

Received: June 30, 2023. Revised: August 16, 2023.

Table 3. Confusion matrices of both classification models

Actual value	C4.5 tree Classification				CNN1D Classification			
	Prediction training		Prediction testing		Prediction training		Prediction testing	
	Class_0	Class_1	Class_0	Class_1	Class_0	Class_1	Class_0	Class_1
Class_0	28345	7810	7051	1972	35637	518	8753	270
Class_1	198	413	96	73	0	611	161	8

Table 4. The comparison both C4.5 tree and CNN1D performance

Performance Metrics	Training dataset		Testing dataset	
	C4.5 tree	CNN1D	C4.5 tree	CNN1D
Accuracy	0.7822	0.9859	0.7750	0.95311
Precision	0.5200	0.9900	0.5100	0.9600
Recall	0.7300	0.9900	0.6100	0.9500
F1-score	0.4800	0.9900	0.4700	0.9600
MCC	0.1411	0.7304	0.0689	0.0137
AUC	0.7299	0.9928	0.6067	0.5087

In the testing data which is 20% part of the dataset, the CNN1D has performance metric values of around 96% in 4 popular metrics namely accuracy, precision, recall, and F1 Score where the C4.5 tree has values on that metrics are 77.50%, 51%, 61%, and 47% respectively. The result shows that the CNN1D outperforms the C4.5 tree in the 4 popular metrics where the result is supported by previous research [46] that the CNN1D outperforms other models in all of the performance metrics used in the research. Nevertheless, an anomaly condition occurs that the C4.5 tree has a better performance than the CNN1D on the metrics of MCC and AUC whose values respectively of [6.89%, 60.67%] and [1.37%, 50.87%] for the C4.5 tree and CNN1D. The result needs a deeper exploration including by Lee [41] find the AUC is a recommendation metric to access the C4.5 tree performance but Chicco and Jurman [48] advised replacing the AUC with the MCC metric in the binary classification problem.

As previously talking that the oversampling method was conducted on the training data and used the training oversampling data to train models with optimal hyperparameters. The evaluation models on the training dataset show the result that the CNN1D outperforms the C4.5 tree in six performance metrics with extreme gaps in both MCC and AUC values which are the CNN1D has 99% AUC and 73% MCC values, while the C4.5 tree has 73%AUC and 14.11 MCC values. The results are not surprising and were supported by previous works in [35] and [36] that the CNN1D outperforms other machine learning models. Besides, the oversampling method also can improve both models' performance which is similar to the works in [14, 15]. Unfortunately, in the testing data, the CNN1D performance in both metrics of MCC and AUC is lower than the C4.5 tree although their performance gaps are narrow enough. The occurrence proved that the CNN1D trained by oversampling data does not predict the positive class well. The contradictory result leads to a challenge in the selection of the metric accuracy when tuning hyperparameters that did not use the accuracy value but it could use the MCC or AUC values [48]. It is also possible the cross entropy loss function is not suitable for the training of a CNN1D model in the imbalance class dataset.

5. Conclusion

The total number of instances is almost to be doubled by conducting the oversampling method on the extremely imbalanced class dataset. The oversampling training dataset has 2 roles, the first one is divided into 5 folds of cross-validation data for tuning hyperparameters, and the second role is to train the C4.5 tree and CNN1D with the optimal hyperparameters. By grid search method, the C4.5 tree has the optimal hyperparameters of 7 depth of the tree and 25 minimum number of instances in the splitting node while the CNN1D has hyperparameters of 4 kernel size and 128 filter size. The C4.5 tree with selected hyperparameters above is trained by using the oversampling training dataset which is also used to train the CNN1D by setting other hyperparameters namely 200 epochs, 1000 mini-batch size, and adaptive momentum gradient descent optimization method. The performance of both models is evaluated by six metrics in both the original training dan testing data. In the training data, the CNN1D has almost perfect performance with around 99% in 5 metrics and a 73% value in the MCC metric. The CNN1D outperforms the C4.5 tree in all performance metrics. In the testing data, the CNN1D outperforms the C4.5

tree in metrics of accuracy, precision, recall, and F1 score with a value of around 99%. Nevertheless, the C4.5 tree has better performance than the CNN1D in the MCC and AUC metrics although their performance gap is narrow enough. This result gives a fact that the CNN1D trained by oversampling training data does not predict well the instance from the positive class yet. The challenge in future works is to use the MCC or AUC criteria for tuning hyperparameters or to develop another loss function else the cross-entropy loss in training the CNN1D for classification of an imbalanced class dataset.

Conflicts of interest

The authors declare no conflict of interest related to this research project.

Author contributions

The contributions of each author are the following: Conceptualization was done by Widodo and Handoyo, methodology was done by Handoyo and Rupiwardhani; software was done by Mursityo and Purwanto, validation was done by Kusdarwati, Purwanto, and Widodo, formal analysis was done by Purwanto, investigation and resources were done by Widodo data curation was done by Kusdarwati and Rupiwardhani, writing—original draft preparation was done by Widodo, writing—review and editing was done by Handoyo; visualization was done by Mursityo, supervision was done by Handoyo, project administration was done by Rupiwardhani, funding acquisition was done by Widodo.

References

[1] T. Ssanzadeh, D. Essam, and R. Sarker, "EvoDCNN: An evolutionary deep convolutional neural network for image classification", *Neurocomputing*, Vol. 488, pp. 271-283, 2022.

[2] F. Yuan, Z. Zhang, and Z. Fang, "An effective CNN and Transformer complementary network for medical image segmentation", *Pattern Recognition*, Vol. 136 p. 109228, 2023.

[3] T. Habite, O. Abdeljaber, and A. Olsson, "Determination of pith location along Norway spruce timber boards using one-dimensional convolutional neural networks trained on virtual timber boards", *Construction and Building Materials*, Vol. 329, p. 127129, 2022.

[4] F. Venturini, M. Sperti, U. Michelucci, A. Gucciardi, V. M. Martos, and M. A. Deriu, "Extraction of physicochemical properties from the fluorescence spectrum with 1D convolutional neural networks: Application to olive oil", *Journal of Food Engineering*, Vol. 336, p. 111198, 2023.

[5] M. Wati, H. R. Hatta, A. D. Saputri, A. Septiarini, and M. Jamil, "Implementation of the C4. 5 Decision Tree Algorithm Method for Selection of Facial Mask Skin Care Products", In: *Proc. of International Conf. on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, pp. 402-406, 2022.

[6] P. Gite, K. Chouhan, K. M. Krishna, C. K. Nayak, M. Soni, and A. Shrivastava, "ML Based Intrusion Detection Scheme for various types of attacks in a WSN using C4.5 and CART classifiers", *Materials Today*, Vol. 80, pp. 3769-3776, 2023.

[7] D. Passos and P. Mishra, "A tutorial on automatic hyperparameter tuning of deep spectral modeling for regression and classification tasks", *Chemometrics and Intelligent Laboratory Systems*, p. 104520, 2022.

[8] S. Nematzadeh, F. Kiani, M. T. Afshar, and N. Aydin, "Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases", *Computational Biology and Chemistry*, Vol. 97, p. 107619, 2022.

[9] S. Handoyo, Y P. Chen, G. Irianto, and A. Widodo, "The Varying Threshold Values of Logistic Regression and Linear Discriminant for Classifying Fraudulent Firm", *Mathematics and Statistics*, Vol. 9, No. 2, pp. 135 – 143, 2021.

[10] A. W. Widodo, S. Handoyo, G. Irianto, N. W. Hidajati, D. S. Susanti, and I. N. Purwanto, "Finding the Best Performance of Bayesian and Naïve Bayes Models in Fraudulent Firms Classification through Varying Threshold", *International Journal of Emerging Technology and Advanced Engineering*, Vol 12, No. 12, pp. 94–106, 2022.

[11] Marji and S. Handoyo, "Performance of Ridge Logistic Regression and Decision Tree in the Binary Classification", *Journal of Theoretical & Applied Information Technology*, Vol. 100, No. 13, pp. 4698-4709, 2022.

[12] A. E. Denoble, S. A. Goldstein, L. E. Wein, C. A. Grotegut, and J. J. Federspiel, "Comparison of severe maternal morbidity in pregnancy by modified World Health Organization Classification of maternal cardiovascular risk", *American Heart Journal*, Vol. 250, pp. 11-22, 2022.

[13] T. Liu, W. Fan, and C. Wu, "A hybrid machine learning approach to cerebral stroke prediction

based on the imbalanced medical dataset", *Artificial Intelligence in Medicine*, Vol. 101, p. 101723, 2019.
[14] I. D. Mienye and Y. Sun, "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data", *Informatics in Medicine Unlocked*, Vol. 25, p. 100690, 2021.
[15] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, "A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data", *Information Sciences*, Vol. 572, pp. 574-589, 2021.
[16] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets", *Applied Soft Computing*, Vol. 83, p. 105662, 2019.
[17] P. Wibowo and C. Fatichah, "Pruning-based oversampling technique with smoothed bootstrap resampling for the imbalanced clinical dataset of Covid-19", *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 9, pp. 7830-7839, 2022.
[18] Marji, S. Handoyo, I. N. Purwanto, and M. Y. Anizar, "The Effect of Attribute Diversity in the Covariance Matrix on the Magnitude of the Radius Parameter in Fuzzy Subtractive Clustering", *Journal of Theoretical and Applied Information Technology*, Vol. 96, No. 12, pp. 3717-3728, 2018.
[19] S. Handoyo, A. Widodo, W. H. Nugroho, and I. N. Purwanto, "The Implementation of a Hybrid Fuzzy Clustering Public Health Facility Data", *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 8, No. 6, pp. 3549-3554, 2019.
[20] I. N. Purwanto, A. Widodo, and S. Handoyo, "System for Selection Starting Lineup of A Football Players by Using Analytical Hierarchy Process", *Journal of Theoretical & Applied Information Technology*, Vol. 97, No. 1, pp. 19-31, 2018.
[21] O. Soesanto, Y. T. Muesityo, E. Darmanto, Zuraidah, N. W. Hidajati, and S. Handoyo, "Implementation of The Fuzzy Analytical Hierarchy Process and Data Envelopment Analyses for Assessment of The Capability of Farmer Groups in Indonesia", *Journal of Theoretical and Applied Information Technology*, Vol. 101, No. 6, pp. 2375–2384, 2023.
[22] H. N. Utami, Candra, and S. Handoyo, "The Effect of Self Efficacy And Hope on Occupational Health Behavior in East Java of Indonesia", *International Journal of Scientific & Technology Research*, Vol. 9, No. 2, pp. 3571-3575, 2019.
[23] H. Kusdarwati and S. Handoyo, "Modeling Treshold Liner in Transfer Function to Overcome Non Normality of the Errors", In: *Proc. of International Conf. On Basic Science*, Malang, Indonesia, Vol. 546, No. 5, p. 052039, 2019.
[24] S. Handoyo, N. Pradianti, W. N. Nugroho, and Y. J. Akri, "A Heuristic Feature Selection in Logistic Regression Modeling with Newton Raphson and Gradient Descent Algorithm", *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 3, pp. 119-126, 2022.
[25] C. Jin, F. Li, S. Ma, and Y. Wang, "Sampling scheme-based classification rule mining method using decision tree in big data environment", *Knowledge-Based Systems*, Vol. 244, p. 108522, 2022.
[26] H. N. Duc and T. V. Duc, "Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification", *Automation in Construction*, Vol. 148, p. 104767, 2023.
[27] F. Tempola, M. Muhammad, A. K. Maswara, and R. Rosihan, "Rule Formation Application based on C4. 5 Algorithm for Household Electricity Usage Prediction", *Trends in Sciences*, Vol. 19, No. 3, pp. 2167-2167, 2022.
[28] S. K. Jain, S. Agrawal, P. K. Shukla, and A. Jain, "A decision tree C4. 5-based voltage security events classifier for electric power systems", *International Journal of Engineering Systems Modelling and Simulation*, Vol. 13, No. 4, pp. 268-276, 2022.
[29] J. Shanthi, D. G. N. Rani, and S. Rajaram, "A C4. 5 decision tree classifier based floorplanning algorithm for System-on-Chip design", *Microelectronics Journal*, Vol. 121, p. 105361, 2022.
[30] J. Peng, C. Chen, M. Zhou, X. Xie, Y. Zhou, and C. H. Luo, "A machine-learning approach to forecast aggravation risk in patients with acute exacerbation of chronic obstructive pulmonary disease with clinical indicators", *Scientific Reports*, Vol. 10, No. 1, pp. 1-9, 2020.
[31] A. Jaiswal and R. Kumar, "Breast cancer diagnosis using Stochastic Self-Organizing Map and Enlarge C4.5", *Multimedia Tools and Applications*, Vol. 82, No. 12, pp. 18059-18076, 2023.
[32] M. Hamdi, L. H. Jaghdam, B. E., Elnaim, and A.

A. Elhag, "Forecasting and classification of new cases of COVID-19 before vaccination using decision trees and Gaussian mixture model", *Alexandria Engineering Journal*, Vol. 62, pp. 327-333, 2023.

[33] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros, and B. J. Fernandes, "A multimodal approach using deep learning for fall detection", *Expert Systems with Applications*, Vol. 168, p. 114226, 2021.

[34] Y. Zhang, X. Wang, J. Xie, and Y. Bai, "Comparative analysis of deep-learning-based models for hourly bus passenger flow forecasting", *Transportation*, pp. 1-26, 2023.

[35] Y. Su, Y. Shi, W. Lee, L. Cheng, and H. Guo, "TAHDNet: Time-aware hierarchical dependency network for medication recommendation", *Journal of Biomedical Informatics*, Vol. 129, p. 104069, 2022.

[36] M. Y. Landolsi, L. Hlaoua, and L. B. Romdhane, "Hybrid method to automatically extract medical document tree structure", *Engineering Applications of Artificial Intelligence*, Vol. 120, p. 105922, 2022.

[37] Y. Jin, L. Hou, and Y. Chen, "A time series transformer-based method for the rotating machinery fault diagnosis", Neurocomputing, Vol. 494, pp. 379-395, 2022.

[38] D. A. L. Alimi, M. A. A. Qaness, Z. Cai, A. Dahou, Y. Shao, and S. Issaka, "Meta-learner hybrid models to classify hyperspectral images", *Remote Sensing*, Vol. 14, No. 4, p. 1038, 2022.

[39] T. Yan, S. L. Shen, A. Zhou, and X. Chen, "Prediction of geological characteristics from shield operational parameters by integrating grid search and K-fold cross-validation into stacking classification algorithm", *Journal of Rock Mechanics and Geotechnical Engineering*, Vol. 14, No. 4, pp. 1292-1303, 2022.

[40] J. Li, F. Gao, S. Lin, M. Guo, Y. Li, H. Liu, and Q. Wen, "Quantum k-fold cross-validation for nearest neighbor classification algorithm. Physica A", *Statistical Mechanics and its Applications*, Vol. 611, p. 128435, 2023.

[41] J. S. Lee, "AUC4.5: AUC-based C4. 5 decision tree algorithm for imbalanced data classification", *IEEE Access*, Vol. 7, pp. 106034-106042, 2019.

[42] S. Sundaramurthy and P. Jayavel, "A hybrid grey wolf optimization and particle swarm optimization with C4. 5 approach for prediction of rheumatoid arthritis", *Applied Soft Computing*, Vol. 94, p. 106500, 2020.

[43] A. Joshuva, R. S. Kumar, S. Sivakumar, G. Deenadayalan, and R. Vishnuvardhan, "An insight on VMD for diagnosing wind turbine blade faults using C4. 5 as feature selection and discriminating through multilayer perceptron", *Alexandria Engineering Journal*, Vol. 59, No. 5, pp. 3863-3879, 2020.

[44] L. Qu, H. Wu, T. Yang, L. Zhang, and Y. Sun, "Dynamic Hand Gesture Classification Based on Multichannel Radar Using Multistream Fusion 1-D Convolutional Neural Network", *IEEE Sensors Journal*, Vol. 22, No. 24, pp. 24083-24093, 2022.

[45] R. F. R. Junior, I. A. D. S. Areias, M. M. Campos, C. E. Teixeira, L. E. B. D. Silva, and G. F. Gomes, "Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals", *Measurement*, Vol. 190, p. 110759, 2022.

[46] D. Huang, S. Li, N. Qin, and Y. Zhang, "Fault diagnosis of high-speed train bogie based on the improved-CEEMDAN and 1-D CNN algorithms", *IEEE Transactions on Instrumentation and Measurement*, Vol. 70, pp. 1-11, 2021.

[47] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating trust prediction and confusion matrix measures for web services ranking", *IEEE Access*, Vol. 8, pp. 90847-90861, 2020.

[48] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification", *BioData Mining*, Vol. 16, No. 1, pp. 1-23, 2023.

[49] G. Canbek, T. T. Temizel, and S. Sagiroglu, "PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics", *SN Computer Science*, Vol. 4, No. 1, pp.1-13, 2022.

[50] F. Shen, X. Zhao, G. Kou, and F. E. Alsaadi, "A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique", *Applied Soft Computing*, Vol. 98, p. 106852, 2021.

[51] A. Özdemir, K. Polat, and A. Alhudhaif, "Classification of imbalanced hyperspectral images using SMOTE-based deep learning methods", *Expert Systems with Applications*, Vol. 178, p. 114986, 2021.