APEM journal

Advances in Production Engineering & Management
Volume 18 | Number 2 | June 2023 | pp 237–249
https://doi.org/10.14743/apem2023.2.470

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Enhancing automated defect detection through sequential clustering and classification: An industrial case study using the Sine-Cosine Algorithm, Possibilistic Fuzzy *c*-means, and Artificial Neural Network

Nguyen, T.P.Q.[a,*], Yang, C.L.[b], Le, M.D.[c], Nguyen, T.T.[d], Luu, M.T.[e]

[a]Faculty of Project Management, The University of Danang – University of Science and Technology, Danang, Vietnam
[b]Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan (ROC)
[c]Faculty of Transportation Mechanical Engineering, The University of Danang – University of Science and Technology, Danang, Vietnam
[d]Faculty of Computer Science and Engineering, University of Technology, Vietnam
[e]Nghia Nippers Corporation, Vietnam

A B S T R A C T

Most existing inspection models solely classify defects as either good or bad, focusing primarily on separating flaws from perfect ones. The sequential clustering and classification technique (SCC) is used in this work to not only identify and categorize the defects but also investigate their root causes. Conventional clustering techniques like *k*-means, fuzzy *c*-means, and self-organizing map are employed in the first stage to find the defects in the finished products. Then, a novel clustering method, that combines a sine-cosine algorithm and possibilistic fuzzy *c*-means (SCA-PFCM), is proposed to classify the detected defects into multiple groups to identify the defect categories and analyze the root causes of failures. In the second stage, the ground truth labels taken from the clustering technique are used to construct an automated inspection system using back propagation neural networks (BPNN). The proposed approach is applicable for detecting and identifying the causes of errors in manufacturing industry. This study applies a case study in nipper manufacture. The SCA-PFCM algorithm can detect 97 % of defects and classify them into four types while BPNN shows a predicted accuracy of up to 96 %. Additionally, an automated inspection system is developed to reduce the time and cost of the inspection process.

A R T I C L E   I N F O

Keywords:
Back Propagation Neural Network;
Clustering;
Classification;
Combined SCA-PFCM;
Defect detection;
Nipper manufacturing;
Possibilistic Fuzzy *c*-means (PFCM);
Root cause analysis;
Sine-Cosine Algorithm (SCA)

**Corresponding author:*
ntpquyen@dut.udn.vn
(Nguyen, T.P.Q)

Article history:
Received 24 April 2023
Revised 12 June 2023
Accepted 23 June 2023

1. Introduction

The rapid rise of both information and operational technology is accelerating the transition from traditional manufacturing to smart factories. A smart factory typically relies on modern information and communication technology, with all of the factory's components being smartly integrated and operated [1]. A smart factory needs to optimize production conditions with a minimum of resources and time. Defects are one of the primary causes of high production costs. Therefore, the inspection system for defect detection is crucial in smart manufacturing, which necessitates the quick detection of defects and categorizes defect types. Effective defect categorization aids in determining and analyzing the type of failure that occurs based on the process conditions [2]. Besides, the defect types can be used to diagnose equipment failure and analyze its root causes. Moreover, an effective inspection system can raise customer satisfaction levels by preventing them from obtaining substandard goods.

Many manufacturing factories still operate the visual and sampling inspection manually. Sampling inspections cannot catch all the defects like a thorough inspection can. Manual inspection may lead to inconsistent results, expensive costs, and time-consuming [3]. Enterprises expect quick defect detection and root cause investigation to provide timely regulation of production line failures [4]. A more effective and reliable automated defect inspection system is required to overcome the drawbacks of sampling and manual inspection in manufacturing.

Generally, a data analysis framework to detect the defects and investigate their causes is necessary. However, the majority of the research that has been done so far has solely focused on identifying defects or defect prediction models, which can result in dividing the finished products into two groups: 1) defects, and 2) non-defective ones [5]. In practice, the manager may not only need to detect failure or an abnormal state in the manufacturing process but also identify the causes of defects in the routine execution of the production plan to make timely adjustments and corrections [6]. This process is known as root cause analysis which aims to determine the underlying cause of an issue and the measures required to solve it.

A novel model of sequential clustering and classification-based genetic algorithm (NSGAII-SCC) has recently been proposed by Yang and Quyen to investigate the hidden structure of data and to identify the features correlated with the explored patterns [7]. The NSGAII-SCC framework is applied to analyze the point of sale (POS) data for a chain of bakeries in China [8]. The bakery store is partitioned into several clusters using a clustering technique and classification is then used to investigate the factors that contribute to the partition process of each store cluster. Kuo *et al.* presented an extension of the NSGAII-SCC by combining a deep learning and multi-objective sine-cosine algorithm (Deep MOSCA-SCC) [9]. The Deep MOSCA-SCC algorithm employed a deep clustering technique that combined auto-encoder and k-means to improve clustering performance. Generally, the SCC approach is useful for defect detection and classification, analyzing its root causes as well as establishing an automatic detect defection model in the above analysis.

Thus, this study focuses on developing a model-based SCC method to detect the defects and analyze their root causes. In the first stage, the proposed model-based SCC method first employs clustering methods to detect the defects in the finished product. Thereafter, the defects are classified into several groups with similar properties to identify the defect categories. Most of the research used traditional clustering techniques like k-means algorithm[10] or fuzzy c-means (FCM) algorithm [11] for industrial applications since they are simple to implement and interpret the result. However, the clustering result was also comparable. In the first stage, several conventional clustering techniques, i.e., k-means, FCM, self-organizing map (SOM) [12], and DBSCAN [13], are employed to detect the defects from the flawless ones since this process is not very complicated due to the predetermined number of clusters k is known as two labels (defect or non-defect). To determine the defect categories and analyze the root causes of failures, a novel clustering method that combines a sine-cosine algorithm and possibilistic fuzzy c-means (SCA-PFCM), is proposed. In the second stage, a back propagation neural network (BPNN) [14] is employed to develop an automatic defect detection model based on the ground truth labels adopted from the clustering technique. In contrast to existing SCC methods, the proposed inspection model-based SCC contributes by combining innovative clustering and classification techniques, namely SCA-PFCM and BPNN, whereas the original NSGAII-SCC suggested an integration of traditional k-means and decision tree classifier. The research result is then applied to an inspection system of a nipper manufacturing factory in Vietnam.

The paper is arranged as follows. The review of sequential clustering and classification method is shown in Section 2. Section 3 describes the production and inspection process in nail nipper manufacturing. The methodology is presented in Section 4. Section 5 illustrates the result analysis of a case study. The concluding remarks and future research direction come in Section 6.

2. Methods and materials

2.1 Review of sequential clustering and classification method

Since clustering is frequently carried out without knowledge regarding the membership of data instances to predetermined labels, it is commonly referred to as a technique for unsupervised learning [15]. The goal of clustering is exploratory the structure of the dataset. The clustering process divides objects in a given dataset into distinct groups or clusters, with each cluster consisting of objects that are similar to each other in that cluster but dissimilar to objects in other clusters. On the contrary, classification is a supervised approach that assigns categories or labels to data records based on previously collected information [16]. The classification process consists of two steps. The first step is to develop a training model based on the classification rules using the given data, which contains a set of attributes and their corresponding outcomes. The training model is then used in the second step to forecast the labels for incoming unknown data. The complexity of the large number of features utilized as a classifier is what makes these processes challenging [17].

Yang and Nguyen [7] proposed the SCC framework, which combines clustering and classification sequentially on two different datasets. The SCC framework can investigate the hidden structure of data and discover the features that are relevant to the explored patterns. Fig. 1 illustrates the SCC framework. From the original dataset, the target interests were identified and these features were then used for discovering data patterns by the clustering algorithm. The remaining dataset contained the relevant features that are correlated with the target features. A clustering algorithm was implemented on the dataset containing target features to explore the data patterns. The result of clustering algorithm was embedded into the training process of the classification model. Two popular clustering algorithms, k-means, and hierarchical clustering, were used to perform clustering. The classification algorithm employed decision trees [18], artificial neural networks (ANN) [19], k-nearest neighbor (KNN) [20], and support vector machine (SVM) [21]. As a result, there are eight clustering-classification algorithm combinations built from two clustering methods and four classification methods. Besides, as combining heuristic techniques and machine learning become more prevalent [22, 23], the SCC framework combined with NSGAII to deal with two objective functions from clustering and classification tasks. Because the number of clusters (k) is not predetermined, the SCC framework implemented the various values of k (from 2 to 10) and finally selected the solution based on the Pareto front. For each value of k, multiple solutions are selected from the first Pareto front to balance two objective functions of clustering and classification tasks. Performance was superior to the others when k-means clustering and decision tree classification were combined.

The Deep MOSCA-SCC algorithm [9] is also an SCC approach that employed deep clustering to enhance the clustering compactness and classification accuracy. Herein, MOSCA [24], which is a simple and easy-to-implement method, was applied to exploit the optimal result for the SCC approach. The utilization of MOSCA in the Deep MOSCA-SCC algorithm was expected to speed up computation compared to NSGAII. However, the experimental results demonstrated that there was no benefit in Deep MOSCA-SCC computational time due to the use of autoencoder, which increased computational time.

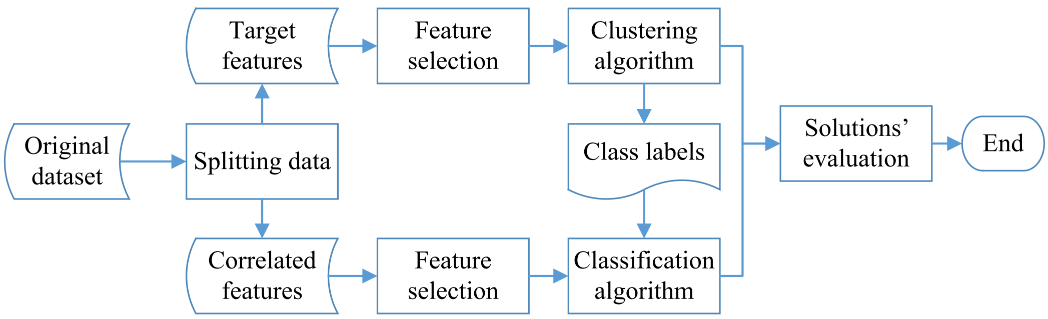

Fig. 1 The SCC approach

2.2 Proposed methodology

The manual examination procedure necessitates a significant amount of human labor and expenditure. Industry 4.0 is having an impact on nail manufacturing, with automated defect detection reducing inspection time and human labor. Automatic visual inspection, which plays a key part in the quality inspection process of smart manufacturing, is a promising technology in this scenario. The SCC technique is used for algorithm development of an automatic visual inspection in nipper manufacturing in this study.

This study utilizes the SCC method [7] to not only detect the defects from the flawless ones but also classify the defects into several groups with similar attributes for root cause analysis in the first stage. An automated defect detection system is developed in the second stage. The result of the first stage is used to train the classification model for automatic defect detection. The methodology framework is shown in Fig. 2.

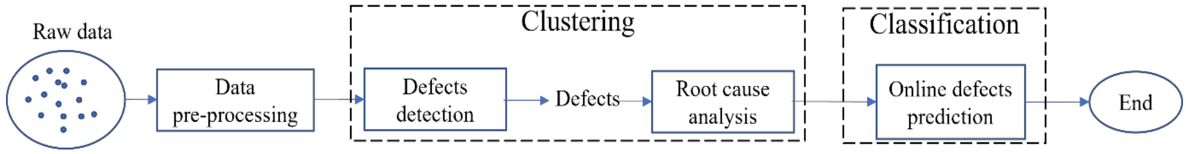

Fig. 2 Methodology Framework

Stage 1: Defect detection and root cause analysis using clustering technique

Detecting defects from the finished products

In this stage, the clustering techniques are employed for defect detection and root cause analysis in nipper manufacturing. The dataset is collected from a nipper factory in Vietnam. The Department of quality control provides historical data on finished items, which is only divided into two categories: defective products and non-defective products. Thus, this study first uses the clustering technique to detect the defects from the flawless ones. The number of clusters is defined as $k = 2$ and the class labels of all data instances are predetermined based on the historical data. k-mean, FCM, SOM, and DBSCAN are employed to classify the defects and non-defected items in the first stage since they are in the group of popular clustering methods applied for industry applications [25, 26]. Clustering accuracy ($Clus\_Acc$) is used to evaluate the result of this step. The $Clus\_Acc$ is calculated as follows [27]:

$$Clus\_Acc = \frac{\sum_{i=1}^{k} b_i}{n}, \tag{1}$$

where b_i is the number of instances that are classified in a corrected cluster, n is the number of total data instances.

Proposed SCA-PFCM for identifying the defect categories

The defects revealed in the first step are classified into several types of defects that have similar properties to analyze the root causes of defects. Herein, k is unknown. Thus, this study proposes a novel clustering method, i.e., SCA-PFCM, which not only automatically determines the optimal k but also simultaneously partitions the clusters to reveal data patterns corresponding to the selected k. The proposed SCA-PFCM inherited the approach of identifying the optimal number of clusters in an automatic fuzzy clustering method proposed by Nguyen and Kuo [28], which deals with automatic clustering for categorical data. However, the dataset from the nipper industry contains numerical features. Thus, the distance measure in the clustering procedure is changed to be appropriate for the data features. Besides, SCA is employed since it is a simple and straightforward algorithm with competitive performance as compared with GA, and PSO [29]. The sine and cosine functions are utilized to update the particle's position in the SCA algorithm as follows:

$$X_i^{(t+1)} = X_i^t + r_1 \times \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right|,\ r_4 < 0.5 \tag{2}$$

$$X_i^{(t+1)} = X_i^t + r_1 \times \cos(r_2) \times \left| r_3 P_i^t - X_i^t \right|,\ r_4 \geq 0.5 \tag{3}$$

where X_i^t *and* $X_i^{(t+1)}$ is the position of individual i at iteration t and $t+1$, respectively. P_i^t is the optimal position at the current iteration t. r_1 is a control number that is calculated as:

$$r_1 = \mu - t\frac{\mu}{T} \tag{4}$$

while r_2 is a number randomly chosen between 0 and 2π, whereas $r_3, r_4 \in (0,2)$. μ is a constant.

The SCA-PFCM algorithm's process is explained as follows:

Step 1: Identify the maximum k (k_{max})
A local density-based approach named RECOME [30] is employed to determine k_{max}. The cluster center typically has higher local density and is surrounded by neighbors with lower local densities. If the number of high-density centers is discovered, the k_{max} can be obtained. The procedure to determine k_{max} is shown in Fig. 3.

Step 2: Set up parameters for SCA and PFCM algorithms.

Step 3: Initialization: Randomly generate the initial population. A solution representation consists of two parts. The first part is used to define the k. A control element C_p is used to determine k in each particle where $k_{min} \leq C_p \leq k_{max}$. k_{min} is set at 2. To determine C_p, we randomly generate a vector $C = C_1, C_2, \dots, C_j, \dots, C_{k_{max}}$ in the range of [0, 1]. C_p is determined by counting the elements in C that are greater than 0.5: C_p = count (C|C_j ≥ 0.5). The second part of a particle is the cluster center corresponding to the value of k in the first part.

Step 4: Calculate fitness: The PFCM algorithm is implemented for each particle in the population. The fitness value is obtained through the objective function of PFMC algorithm.

Step 5: Identify the optimal solution P_i^t at iteration *t.*

Step 6: Compute r_1 using Eq. 4 and randomly generate r_2 r_3, and r_4.

Step 7: Update the cluster centers using Eq. 2 and Eq. 3.

Step 8: Return to step 4 until the stopping condition is met.

Input: a given dataset X, $N(x)$ is the K nearest neighbors set of x in X.

Output: k_{max}

- For each instance $x \in X$, and $N(x)$ is the K nearest neighbors set of x in X:
  - Calculate K nearest Neighbor Kernel Density (NKD):
    $$\rho(x) = \theta \sum_{y \in N(x)} \exp\left(-\frac{d(x,y)}{\sigma}\right)$$
  - Caculate relative K nearest Neighbor Kernel Density (RNKD):
    $$\rho^*(x) = \frac{\rho(x)}{\max_{y \in N(x) \cup \{x\}} \rho(y)}$$
- Define core instances: $O = \{x | x \in X, \rho^*(x) = 1\}$.
- Determine Higher Density Nearest-neighbor (HDN)-$\pi(x)$:
  $$\pi(x) = argmin\ \{d(x,y)\}, where\ \rho(y) > \rho(x), y \in X.$$
- Construct a directed graph $G = (X, A)$, where $A = \{\langle x, \pi(x)\rangle | x \in X \backslash O\}$.
- Find atom cluster: for a core instance C_o in O
  - The atom cluster $C_a = \{C_o\} \cup \{x \in XO | x\ is\ connected\ to\ C_o\ in\ G\}$.
- k_{max} is defined by number of atom clusters

Fig. 3 Procedure to determine k_{max} [28]

The result in this stage, which includes the class labels and the number of clusters for defects, is used to train the model in the second stage.

Stage 2: Automatic defect detection using back propagation neural network (BPNN)

BPNN [14] is a popular branch of ANN that contains a multi-layer feedforward network trained by the error BP technique. A BPNN consists of three different layer types: input, hidden, and output layers. The hidden layer acts as a link for signal forward propagation connecting the input and output layers. If the output layer is unable to produce the required results, the error BP will transmit an error signal from the output nodes back via the hidden layer to the input nodes before repeating the signal forward propagation.

This study employs BPNN to develop an automated defect prediction model. The labels that are revealed by clustering techniques in the first stage are used to train the BPNN model. The number of neurons in the input layer is identified by the number of data features that significantly impact the output. The output layer nodes depend on the defect categories that are partitioned in the first stage. It is calculated by multiplying the number of defect types by one to represent the defect categories and good products. The performance of the BP network is significantly influenced by the number of hidden layers and hidden layer nodes. A single hidden layer or two hidden layers can be considered in the design of BPNN model depending on its predictive performance [31]. Besides, better performance can be achieved with more hidden layer nodes, although a lengthy training period may result. Typically, an empirical formula is used to estimate the number of hidden layer nodes. The empirical formula is displayed as follows [32]:

$$H = \sqrt{I + O} + a, \tag{5}$$

where I, H, and O are the number of nodes in the input, hidden, and output layers, respectively, a is a constant number in the range [0, 10]. Cross-entropy is selected as the loss function, which is calculated as follows [33]:

$$E = -\sum_{i=1}^{N}(y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - \hat{y}_i)), \tag{6}$$

where y_i is the target value, $\hat{y}_i$ is predicted value. The structure of BPNN that contains two hidden layers for automated defect prediction is shown in Fig. 4.

The proposed method is applicable for detecting and identifying the causes of errors in industrial manufacturing. There are two ways that the input data can be gathered from the industry: sensor signal data or image data. The proposed method in its current form can work well for signal data. If the input is the image data, it can be preprocessed to extract the features from the image before the proposed method is implemented. This paper analyzes a case study in nipper manufacturing. An automatic defect detection system is developed in Section 3.3 in which the product images are taken on the production line. The user can adjust the model to match the type of input data.

The nipper manufacturing industry in Vietnam, which is selected as a case study for this research, will be covered in the next subsection, along with the production and inspection processes.

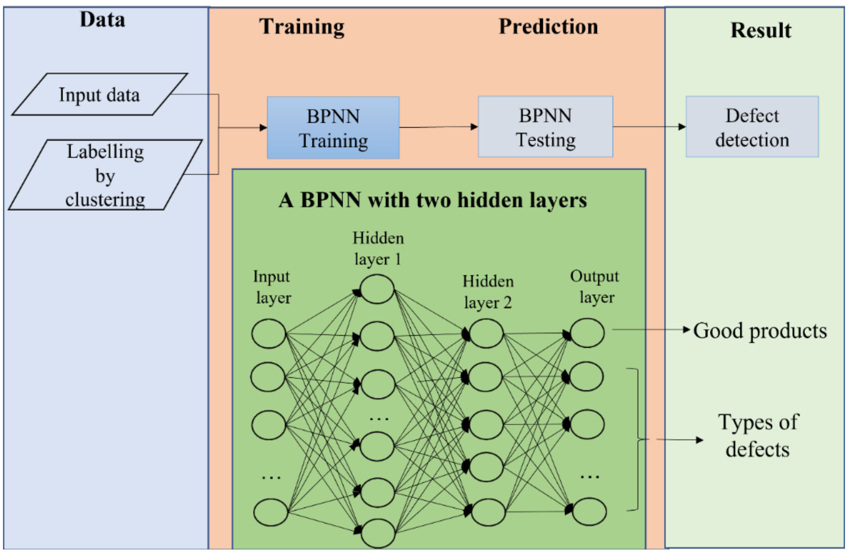

Fig. 4 The structure of BPNN model for automated defect prediction

2.3 Production and inspection processes in nipper manufacturing

This section introduces the production and inspection processes in nipper manufacturing. There are two types of nippers: cuticle nippers and nail nippers. The cuticle nipper usually has a smaller jaw and handle, as compared to the nail nipper. Besides, the cuticle nipper also has a lighter weight than the nail nipper. A nipper consists of five main components: jaw, joint, spring, spring pin, and handles. The jaw is designed with exact blade alignment and precision sharpening. For the cuticle nipper, there is different jaw size such as J12, J14, and J16 which are corresponding to the blade sizes are 4.5-5.5 mm, 5.5-6.5 mm, and 6.5-7.5 mm, respectively. In contrast, the nail nipper only has one jaw size of 6.5-8.0 mm. The joints aim to transmit the cutting force from the handles to the two jaws when the user squeezes them. Box joint is designed to warranty it is durable. Springs are used to assist (push and elastic) the user when squeezing the two handles. The pin is used to hold the spring with the handle. Besides, the company name or logo is also etched on the handle.

The manufacturing process of a nipper is described in Fig. 5. There are four main stages in the whole production process. The first stage is to make a mold from the mold blueprint. In the pressing stage, several processes, such as cutting, stamping for shaping, bending the blade edges, annealing mill, and grinding the surface, are continuously carried out. The output of this stage is a shaped nipper. The process is continued in the pre-processing stage which includes the stamping and drilling assembly, thermal process, shaping of the jaw, polishing of the handle, and plating. Thereafter, the preliminary nipper is transferred to the finishing stage with the following processes: installing springs, printing laser, polishing the jaw, sharpening the blade edge, and cleaning the finished product. There is an inspection process to check product quality at each stage. Then, the inspection process is applied to the finished product. The criteria for inspection consist of sharpness, the shape of the jaw, logo printing, length of the blade, and handle length. The inspection process is performed manually and is costly. Moreover, the number of products per day is also huge. Thus, sampling inspection is currently used with a sample of 10 % to 20 % of finished products randomly taken for inspection. The sampling inspection process cannot guarantee the quality of all the finished products.

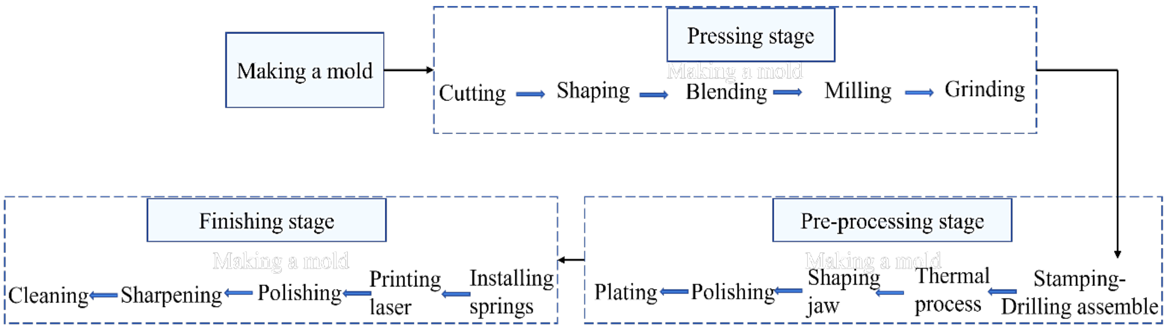

Fig. 5 The manufacturing process of a nipper

3. Case study: Result and discussion

3.1 Data collection and parameter setting

The experiment uses the historical data gathered from a nipper manufacturing factory in Vietnam. According to the company's current production needs, a total of 15000 nippers of various types must be produced per day. As a result, the amount of data recorded day by day becomes very huge to process. This study analyzes three different types of nail nippers, coded D555, D401, and D363, respectively. D555 includes 84320 data instances, whereas D401 and D363 have 48384 and 33690 data instances. Each data instance has the 12 features such as nipper length, handle, blade length, width and thickness of nipper handles, thickness of two nipper bearings, and so on.

In the first stage, k-means, FCM, SOM, and DBSCAN are used to classify the defects from the flawless ones. The number of clusters is predetermined as $k = 2$. Clustering accuracy is employed to evaluate the result. Then the discovered defects are used to perform clustering to determine the defect types by using the proposed SCA-PFCM algorithm. To compare the optimal k obtained from the SCA-PFCM, the elbow method-based k-means is employed in which the sum of squared error (SSE) is considered as a performance indicator. SSE is calculated as follows:

$$SSE = \sum_{j=1}^{k} \sum_{x \in C_j} dist(x, m_j)^2 \tag{7}$$

The parameter setting for BPNN model will be discussed in more detail in the next subsection after getting the result of the first stage.

3.2 Result analysis

First-stage experimental findings

Clustering methods are first used to detect the defects in the finished product. The k-means, FCM, SOM, and DBSCAN algorithms are implemented on the three datasets. Each algorithm is executed 30 times for each tested dataset. Table 1 shows the clustering accuracy based on the average values of 30 runs and the standard deviation. It is not much different from the clustering accuracy by k-means and FCM. The DBSCAN algorithm is slightly better since its accuracy dominates the other algorithms for all tested data. Thus, the clustering result of DBSCAN is used in the next step. Table 2 shows the number of defective and non-defective products detected by DBSCAN method.

Table 1 Clustering accuracy of the process to classify defects from flawless ones

Dataset	k-means	FCM	SOM	DBSCAN
D555	0.912 ± 0.001	0.904 ± 0.003	0.896 ± 0.003	0.913 ± 0.001
D401	0.938 ± 0.001	0.940 ± 0.002	0.913 ± 0.002	0.942 ± 0.002
D363	0.953 ± 0.001	0.951 ± 0.001	0.948 ± 0.002	0.965 ± 0.001

Table 2 Clustering results in terms of class labels

Dataset	Number of instances		
	Total	Non-defective	Defective
D555	84320	46873	37447
D401	48484	28245	20239
D363	33690	18970	14720

To analyze the root cause of failures, the defects are classified based on their features. The number of clusters, which is represented by how many types of defects, is unknown. Elbow method-based k-means clustering is applied. Fig. 6 shows the SSE plot for various k in the D555, D401, and D363 datasets, respectively. The result is quite consistent since the optimal k is selected at $k = 4$, which means that there are 4 types of defects.

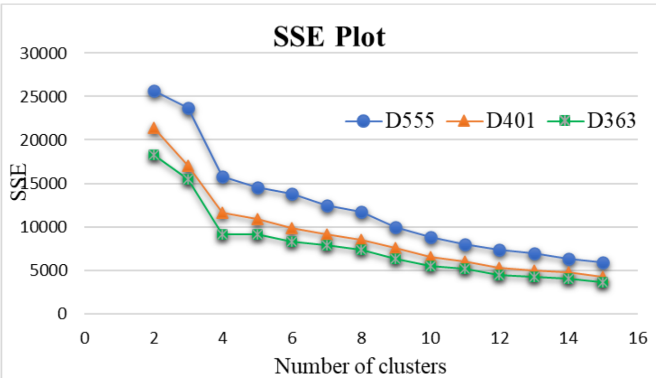

Fig. 6 Identify the number of defect types based on the elbow method

Besides, the number of defect types is also determined using the proposed SCA-PFCM. The result is quite similar to the elbow-based k-means since four types of defects are determined. However, the performance in terms of SSE of the SCA-PFCM is smaller than that of the elbow-based k-means, as shown in Table 3. It means that the clustering result provided by SCA-PFCM is more compact. The number of defect types and the label of each defect instance obtained from the

SCA-PFCM are used in the second stage. The defect types can be listed as follows. The first type of defect is untight jaws since there is a clearance between two blades of the nipper that exceeds the gap standard. In the blade sharpening process, the craftsman leaves a gap that is allowed according to the standard. If they sharpen it a lot, there is a gap between the two blades of the nipper, and the length of the blade may be shorter. Most of the defects derive from this cause. The second and third types of defect are large gaps in the upper and lower gills of the nipper caused by the assembly process of the box joint. The four defect type belongs to the length of the nipper blade as it can be shorter or longer than its standard length due to the blade grinding process.

Table 3 Comparison of SCA-PFCM and elbow-based k-means in terms of SSE

Dataset	Optimal k	SSE	
		SCA-PFCM	Elbow method
D555	4	14233.89	15784.78
D401	4	10602.53	11649.34
D363	4	8321.94	9157.63

Experimental results in the second stage

The number of defect types and the label for each defect are employed to train the model in the second stage. This stage aims to not only detect the defects from the finished products but also classify the defects into certain groups that are identified in the first stage. Hence, there are five labels representing four types of defects and good product and their corresponding number of instances are presented in Table 4.

The parameter for BPNN model with a single hidden layer (denoted as BPNN-1) is set up as follows. First, the input layer nodes equal to 12 which are the significant non-linear relationship variables to represent product features. The output layer nodes are 5 including the defect types and good product. Based on Eq. (5), the number of neurons in the hidden layer is from 4 to 14. After several trial experiments, the hidden layer nodes are selected as 10, initial weight $w = 0.2$, learning rate $r = 0.1$, and inertia weight $c = 0.6$. Logsig and purelin are the activation functions for the hidden and output layers, respectively [31]. Regarding the BPNN model with two hidden layers (BPNN-2), the input and output layers are set up similarly to the BPNN-1 model. The number of neurons in the first hidden layer is also from 4 to 14 as determined by Eq. (5). This study eliminates half of the constant number in Eq. (5), which now ranges from 0 to 5, to shorten the computation time. Thus, there are 4 to 9 neurons in the second hidden layer. The activation functions and other parameters are similar to the BPNN-1 model. After a series of experimental trials, the first and second hidden layers are chosen to have 12 and 6 neurons, respectively, based on their predictive performance. Table 5 shows the selection of key parameters for BPNN models.

Table 4 Data properties for classification

Label	Types of defect	Number of instances		
		D555	D401	D363
1	Non-defective	46873	28245	18970
2	Untight jaws	19326	11017	6793
3	Large gap in the upper gill of nipper	8768	3790	3278
4	Large gap in the lower gill of nipper	6293	4032	3480
5	Length of nipper blade	3060	1400	1169

Table 5 Setting for BPNN model

Model	Number of neurons			Activation function	
	Input layer	Hidden layer	Output layer	Hidden layer	Output layer
BPNN-1	12	10	5	logsig	purelin
BPNN-2	12	12-6	5	logsig	purelin

To evaluate how well the BPNN model predicts, the confusion matrix is utilized [34]. Four predictive performance metrics are obtained from the confusion metrics, as follows:

$$Accuracy = (\text{TP+TN})/(\text{TP+FP+TN+FN}) \tag{8}$$

$$Recall = \text{TP}/(\text{TP+FN}) \tag{9}$$

$$Precision = TP/(TP + FP) \tag{10}$$

$$F\_measure = 2 \cdot Recall \cdot Precision/(Recall + Precision) \tag{11}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. Table 6 shows the experimental result in terms of the confusion matrix of dataset D555 implemented by the BPNN-1 model. The overall accuracy is 94.6 % which is calculated by taking the average of the correctly predicted values in each class. The overall error rate correspondingly is 5.4 %. Besides, Kappa coefficient [35] is also used to evaluate the agreement between predicted values and truth class values. The kappa coefficient is computed as 93.2 % to represent a perfect agreement. Similarly, the confusion matrix of dataset D555 implemented by the BPNN-2 model is displayed in Table 7. The overall accuracy, overall error rate, and Kappa coefficient achieved by BPNN-2 are 96 %, 4 %, and 94.8 %, respectively, which are relatively higher than those achieved by BPNN-1 model. Table 8 summarizes the result comparison of the BPNN-1 and BPNN-2 models in the three tested datasets. The BPNN-2 model is significantly better than the BPNN-1 model in terms of accuracy and Kappa coefficient. Besides, the computational time on the product group of each tested dataset is also presented in Table 8. The BPNN-2 takes more time for the training model. Therefore, compared to the BPNN-1, its computing time is significantly longer. However, the result in Table 8 is shown for the product group that includes numerous objects. The highest computational times, while taking into account the average computational time for each product, are 0.052 and 0.045 for BPNN-2 and BPNN-1, respectively. The average computation time per product provided by the two models does not differ considerably while the accuracy of BPNN-2 is significantly better. Thus, the BPNN-2 model is used to develop an automatic defect inspection system presented in the next subsection.

Table 6 Confusion matrix of dataset D555 implemented by the BPNN-1 model

BPNN-1 Dataset D555		Actual				
		Non-defective (%)	Untight jaws (%)	Large gap in the upper gill (%)	Large gap in the lower gill (%)	Over-length of nipper blade (%)
Predicted	Non-defective	100	0	0	0	0
	Untight jaws	0	100	2	13	0
	Large gap in the upper gill	0	0	88	0	0
	Large gap in the lower gill	0	0	8	87	2
	Over-length of nipper blade	0	0	2	0	98

Table 7 Confusion matrix of dataset D555 implemented by the BPNN-2 model

BPNN-2 Dataset D555		Actual				
		Non-defective (%)	Untight jaws (%)	Large gap in the upper gill (%)	Large gap in the lower gill (%)	Over-length of nipper blade (%)
Predicted	Non-defective	100	0	0	0	0
	Untight jaws	0	100	0	10	0
	Large gap in the upper gill	0	0	93	0	0
	Large gap in the lower gill	0	0	5	90	3
	Over-length of nipper blade	0	0	2	0	97

Table 8 Result comparison

Dataset	D555		D401		D363	
	BPNN-1	BPNN-2	BPNN-1	BPNN-2	BPNN-1	BPNN-2
Overall accuracy (%)	94.6	96	93.5	95.2	92.8	96.3
Overall error rate (%)	5.4	4	6.5	4.8	7.2	3.7
Kappa Coefficient (%)	93.2	94.8	93.1	94.7	92.1	95.9
Computational time (s)	2440.5	2916.2	1795.5	2091.9	1529.2	1751.6

3.3 Automated defect inspection system

The BPNN-2 model whose results were validated in Section 5.2.2 is used in the automatic inspection system for nipper manufacturing. To transfer from manual inspection to the automatic inspection process, an advanced sensing system with smart camera sensors is installed. The system includes two industry cameras that are installed perpendicular to the conveyor. One camera is used to check the overall sizes of a nipper. Another one is put closer for enlarging to check the criteria related to the nipper's jaw. The cameras take pictures of the product from different angles. Then the system will analyze the images to extract the related feature to evaluate the product quality. There are a total of 12 features extracted from the image captured by from sensor camera, such as the nipper length, handle, blade length, the clearance between two nipper blades, width, and thickness of nipper handles, the thickness of two nipper bearings, and so on. Besides, the optical illumination system is installed under a transparent conveyor and shines in the opposite direction through the conveyor towards the camera.

The automatic visual inspection system in nipper manufacturing is displayed in Fig. 7. The system is operated as follows. Operator 1 puts the nippers on the conveyor in the correct direction. The system will adjust to make the nipper on the conveyor alignment. The sensor camera takes pictures automatically when the nipper passes through the inspection position. The image is transmitted to the processing system and the product features are extracted.

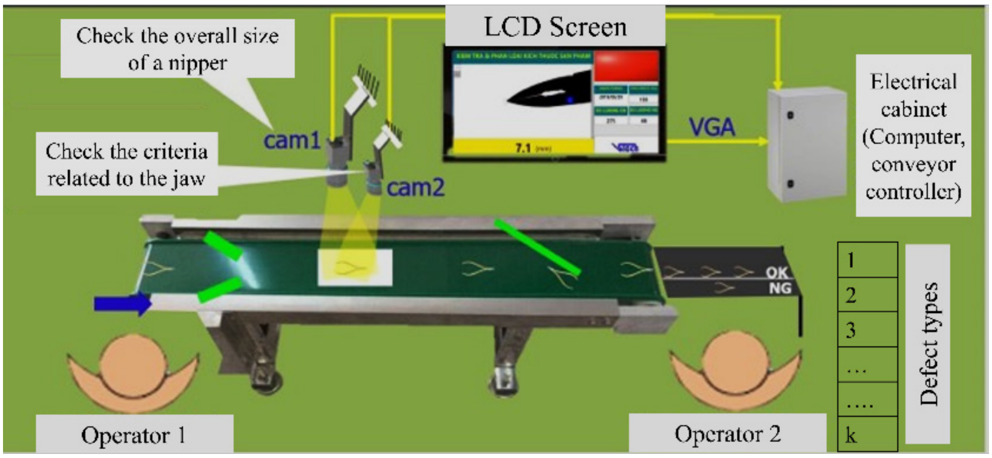

Fig. 7 The automatic inspection system in nipper manufacturing

The previous system at the company only detects a nipper to a defect or non-defective product. If a tested nipper is determined as a non-defective one, the product sorting mechanism is not active. Thus, the nipper will automatically transfer to the finished product tray. Otherwise, the product sorting mechanism is active to push a nipper on the other side of the conveyor and transfer it to the defect tray. However, this study considers not only detecting the defects but also finding the root cause of defects. Thus, the defects are classified into four groups based on their features as the result of the experiment. A classification model based on the clustering result in the first stage is developed to classify the tested nipper and determine exactly what type of defect the tested nipper is. The result is not only displayed on the screen but also has a sound warning to remind Operation 2 of the types of defects. Operator 2 has a responsibility to take the classified defect into the correct tray of defect types.

4. Conclusion

Defect detection is critical for quality control in the company. By detecting the defects in the finished products, the company can prevent sending out defects to customers. Classifying the defects into different categories based on their features also helps to determine the defect's causes. The management can utilize this information to make better decisions about addressing the defects or eliminating the factors that caused defects. Thus, the SCC method is utilized in this study to perform these works for the inspection process in the nipper factory in Vietnam. The DBSCAN clustering method is used to detect the defects in the finished products. Besides, a novel SCA-PFCM algorithm is proposed for defect categories. Then BPNN is employed to make an automatic inspection system. The result showed that the clustering and classification accuracy are all high to prove that the model is robust. The defects in the nipper were detected and classified into four different types based on the features of the defective products. Based on this result, an automatic defect detection model-based visual inspection was developed to help the factory improve the inspection process in terms of quality, time, and cost.

There are some limitations in the proposed method that can be enhance for further research. First, the clustering stage is critical important and affect the result of the classification stage since its result is embedded in the classification process. Thus, improving the clustering result is necessary. The proposed SCA-PFCM only considered within cluster distance as the objective function in the clustering process. Future research can investigate mul-objective function

This research can be extended to detecting the failures in the whole manufacturing process, not only in the finishing stage, as the recommendation by the manager. Besides, developing an algorithm to improve the quality of combining clustering and classification is also necessary.

Acknowledgment

This work was funded by Vingroup Joint Stock Company (Vingroup JSC) and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2020.DA19. The support is much appreciated. The authors also thank Nghia Nippers Corp. for providing a dataset and support during the project.

References

[1] Yun, J.P., Shin, W.C., Koo, G., Kim, M.S., Lee, C., Lee, S.J. (2020). Automated defect inspection system for metal surfaces based on deep learning and data augmentation, *Journal of Manufacturing Systems*, Vol. 55, 317-324, doi: 10.1016/j.jmsy.2020.03.009.

[2] Wang, T., Chen, Y., Qiao, M., Snoussi, H. (2018). A fast and robust convolutional neural network-based defect detection model in product quality control, *International Journal of Advanced Manufacturing Technology*, Vol. 94, No. 9, 3465-3471, doi: 10.1007/s00170-017-0882-0.

[3] Ko, T., Lee, J.H., Cho, H., Cho, S., Lee, W., Lee, M. (2017). Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data, *Industrial Management & Data Systems*, Vol. 117, No. 5, 927-945, doi: 10.1108/IMDS-06-2016-0195.

[4] Yang, Z.J., Du, X.J., Chen, F., Chen, C.H., Tian, H.L., He, J.L. (2018). Change-point estimation for repairable systems combining bootstrap control charts and clustering analysis: Performance analysis and a case study, *Advances in Production Engineering & Management*, Vol. 13, No. 3, 307-320, doi: 10.14743/apem2018.3.292.

[5] Božič, J., Tabernik, D., Skočaj, D. (2021). Mixed supervision for surface-defect detection: From weakly to fully supervised learning, *Computers in Industry*, Vol. 129, Article No. 103459, doi: 10.1016/j.compind.2021.103459.

[6] Chongwatpol, J. (2015). Prognostic analysis of defects in manufacturing, *Industrial Management & Data Systems*, Vol. 115, No. 1, 64-87, doi: 10.1108/IMDS-05-2014-0158.

[7] Yang, C.-L., Quyen, N.T.P. (2018). Data analysis framework of sequential clustering and classification using non-dominated sorting genetic algorithm, *Applied Soft Computing*, Vol. 69, 704-718, doi: 10.1016/j.asoc.2017.12.019.

[8] Yang, C.-L., Nguyen, T.P.Q. (2022). Sequential clustering and classification approach to analyze sales performance of retail stores based on point-of-sale data, *International Journal of Information Technology & Decision Making,* Vol. 21, No. 3, 885-910, doi: 10.1142/S0219622022500079.

[9] Kuo, R.J., Setiawan, M.R., Nguyen, T.P.Q. (2022). Sequential clustering and classification using deep learning technique and multi-objective sine-cosine algorithm, *Computers & Industrial Engineering*, Vol. 173, Article No. 108695, doi: 10.1016/j.cie.2022.108695.

[10] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 24, No. 7, 881-892, doi: 10.1109/TPAMI.2002.1017616.

[11] Bezdek, J.C., Ehrlich, R., Full, W. (1984). FCM: The fuzzy *c*-means clustering algorithm, *Computers & Geosciences*, Vol. 10, No. 2-3, 191-203, doi: 10.1016/0098-3004(84)90020-7.

[12] Vesanto, J., Alhoniemi, E. (2000). Clustering of the self-organizing map, *IEEE Transactions on Neural Networks,* Vol. 11, No. 3, 586-600, doi: 10.1109/72.846731.

[13] Ienco, D., Bordogna, G. (2018). Fuzzy extensions of the DBScan clustering algorithm, *Soft Computing*, Vol. 22, No. 5, 1719-1730, doi: 10.1007/s00500-016-2435-0.

[14] Goh, A.T.C. (1995). Back-propagation neural networks for modeling complex systems, *Artificial Intelligence in Engineering*, Vol. 9, No. 3, 143-151, doi: 10.1016/0954-1810(94)00011-S.
[15] Nguyen, T.P.Q., Kuo, R.J., Le, M.D., Nguyen, T.C., Le, T.H.A. (2022). Local search genetic algorithm-based possibilistic weighted fuzzy *c*-means for clustering mixed numerical and categorical data, *Neural Computing and Applications*, Vol. 34, No. 20, 18059-18074, doi: 10.1007/s00521-022-07411-1.
[16] Tan, P.N., Steinbach, M., Kumar, V. (2016). *Introduction to data mining,* Pearson Education, New Delhi, India.
[17] Zhang, Z.L., Wang, Y.F., Li, Y. (2019). Inventory control model based on multi-attribute material classification: An integrated grey-rough set and probabilistic neural network approach, *Advances in Production Engineering & Management*, Vol. 14, No. 1, 93-111, doi: 10.14743/apem2019.1.314.
[18] Zhang, Y.D., Liao, L., Yu, Q., Ma, W.G., Li, K.H. (2021). Using the gradient boosting decision tree (GBDT) algorithm for a train delay prediction model considering the delay propagation feature, *Advances in Production Engineering & Management*, Vol. 16, No. 3, 285-296, doi: 10.14743/apem2021.3.400.
[19] Tao, X., Zhang, D., Ma, W., Liu, X., Xu, D. (2018). Automatic metallic surface defect detection and recognition with convolutional neural networks, *Applied Sciences,* Vol. 8, No. 9, Article No. 1575, doi: 10.3390/app8091575.
[20] Keller, J.M., Gray, M.R., Givens, J.A. (1985). A fuzzy K-nearest neighbor algorithm*, IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 15, No. 4, 580-585, doi: 10.1109/TSMC.1985.6313426.
[21] Gao, H.N., Shen, D.H., Yu, L., Zhang, W.C. (2020). Identification of cutting chatter through deep learning and classification, *International Journal of Simulation Modelling*, Vol. 19, No. 4, 667-677, doi: 10.2507/IJSIMM19-4-CO16.
[22] Wang, Y.D., Lu, X.C., Shen, J.R. (2021). Improved Genetic Algorithm (VNS-GA) using polar coordinate classification for workload balanced multiple Traveling Salesman Problem (mTSP), *Advances in Production Engineering & Management*, Vol. 16, No. 2, 173-184, doi: 10.14743/apem2021.2.392.
[23] Zhang, L., Ma, J., Liu, X., Zhang, M., Duan, X., Wang, Z. (2022). A novel support vector machine model of traffic state identification of urban expressway integrating parallel genetic and *c*-means clustering algorithm, *Tehnički Vjesnik – Technical Gazette,* Vol. 29, No. 3, 731-741, doi: 10.17559/TV-20211201014622.
[24] Tawhid, M.A., Savsani, V. (2019). Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems*, Neural Computing and Applications*, Vol. 31, No. 2, 915-929, doi: 10.1007/s00521-017-3049-x.
[25] Chaouni Benabdellah, A., Benghabrit, A., Bouhaddou, I. (2019). A survey of clustering algorithms for an industrial context, *Procedia Computer Science,* Vol. 148, 291-302, doi: 10.1016/j.procs.2019.01.022.
[26] Gocken, T., Yaktubay, M. (2019). Comparison of different clustering algorithms via genetic algorithm for VRPTW, *International Journal of Simulation Modelling*, Vol. 18, No. 4, 574-585, doi: 10.2507/IJSIMM18(4)485.
[27] Kuo, R.J., Zheng, Y.R., Nguyen, T.P.Q. (2021). Metaheuristic-based possibilistic fuzzy k-modes algorithms for categorical data clustering, *Information Sciences*, Vol. 557, 1-15, doi: 10.1016/j.ins2020.12.051.
[28] Nguyen, T.P.Q., Kuo, R.J. (2019). Automatic fuzzy clustering using non-dominated sorting particle swarm optimization algorithm for categorical data, *IEEE Access*, Vol. 7, 99721-99734, doi: 10.1109/ACCESS.2019.2927593.
[29] Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems, *Knowledge-Based Systems*, Vol. 96, 120-133, doi: 10.1016/j.knosys.2015.12.022.
[30] Geng, Y.A., Li, Q., Zheng, R., Zhuang, F., He, R., Xiong, N. (2018). RECOME: A new density-based clustering algorithm using relative KNN kernel density, *Information Sciences*, Vol. 436-437, 13-30, doi: 10.1016/j.ins.2018.01.013.
[31] Cui, Y., Liu, H., Wang, Q., Zheng, Z., Wang, H., Yue, Z., Ming, Z., Wen, M., Feng, L., Yao, M. (2022). Investigation on the ignition delay prediction model of multi-component surrogates based on back propagation (BP) neural network, *Combustion and Flame*, Vol. 237, Article No. 111852, doi: 10.1016/j.combustflame.2021.111852.
[32] Dong, Y., Fu, Z., Peng, Y., Zheng, Y., Yan, H., Li, X. (2020). Precision fertilization method of field crops based on the Wavelet-BP neural network in China, *Journal of Cleaner Production*, Vol. 246, Article No. 118735, doi: 10.1016/j.jclepro.2019.118735.
[33] Sadowski, P. (2016). Notes on backpropagation, from http://web.khu.ac.kr/~tskim/MLPR%2025-3%20Back-propagation%20with%20CE%20(Handout).pdf, accessed 5 April 2023.
[34] Wang, K.-J., Hao, F.-J., Lee, Y.-X. (2022). A multiple-stage defect detection model by convolutional neural network, *Computers & Industrial Engineering*, Vol. 168, Article No. 108096, doi: 10.1016/j.cie.2022.108096.
[35] Chicco, D., Warrens, M.J., Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment, *IEEE Access*, Vol. 9, 78368-78381, doi: 10.1109/ACCESS.2021.3084050.