This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Generating Multi-Codebook Neural Network by Using Intelligent Gaussian Mixture Model Clustering Based on Histogram Information for Multi-Modal Data Classification

M. Anwar Ma'sum [1], Member, IEEE, Noverina Alfiany[1], Member IEEE, and Wisnu Jatmiko[1], Senior Member IEEE

[1]Faculty of Computer Science Universitas Indonesia, Kampus UI Depok, Jawa Barat Indonesia ID

Corresponding author: M. Anwar Ma'sum, Noverina Alfiany, and Wisnu Jatmiko (e-mail: muhammad_anwar@ cs.ui.ac.id, noverina.alfiany@ui.ac.id, wisnuj@cs.ui.ac.id).

This work is supported by Universitas Indonesia PUTI Q1 Research Grant 2020-2021 Grant Number NKB-3885/UN2.RST/HKP.05.00/2020.

ABSTRACT One of the open challenges in machine learning is multi-modal data classification. A classifier model needs to be enhanced to deal with multi-modal data. This study is proposed to develop multi-codebook neural networks using intelligent Gaussian mixture model clustering for multi-modal data classification. The intelligent Gaussian mixture model clustering is developed in this study prior to the development of multi-codebook models. The method analyzes the gradient of input data histogram to find the number of generated mixtures and cluster the data. The proposed multi-codebook neural network model has three variants based on the rules to fund the number of clusters. The experiment result showed that the proposed three variants of multi-codebook models performed well in the synthetic and benchmark datasets. The proposed model improved the original method by 24.14%, 15.97%, and 3.71% accuracy and 0.3510, 0.0487, and 0.2031 kappa for synthetic datasets, benchmark datasets, and overall datasets respectively. By using the ANOVA test, we have proved that all three variants of the proposed multi-codebook neural network were proved to have significant improvements compared to the original version.

INDEX TERMS multi-modal, classification, multi-codebook, neural networks, intelligent clustering, Gaussian mixture model, histogram.

I. INTRODUCTION

Classification is one of the supervised approaches in machine learning that's used to predict the output class label (category) given input data. The classifiers predict the class label of the input data based on its attributes which are called features. The classification methodology is widely applied in numerous areas e.g. life science, business, human-interaction science, engineering, medical, biology, robotics, accounting and finance, education, and many other scientific disciplines. Within the intelligent robotics field of research, machine learning is utilized as the core of system artificial intelligence modules. For example, autonomous robots utilized machine learning for object detection and tracking, path planning, field mapping, and robot navigation [1]–[3]. One of the open challenges in machine learning is multi-modal classification [4].

Multi-modal data is a set of data produced by a data capture from heterogeneous sources. Heterogenous may be interpreted as different types of sources or multiple sources that have the same types of data, but it has different characteristics. An example of multiple types of data capture is capturing users' reviews in e-commerce by using star values (numeric) and users' opinions written in review form (text). An example of a single type of data capture that produces multi-modality is user preference analysis of two compared products in e-commerce based on users' age. For example, a gadget-A has been bought by users aged from 25 to 40. While the gadget-B has been bought by users aged below 25 (0-25) and above 40 (40-60). From the users' age distribution point of view, we can say that class-A (gadget-A) has unimodal distribution as the value of users' age distributed only in one range are (25-40). While class-B (gadget-B) has multi-modal distribution as the users' age is distributed in two areas i.e. (0-25) and (40-60). In this study, we focused on multi-modality that is produced by single types of data captures. Addressing this type of multi-modality is important as usually we don't expect multi-modal data from measurement, but unintentionally we ended up dealing with multi-modal data even though we capture a single type of data [4]. More detailed information about the motivation, objectives, and focus of this study is explained in the following paragraphs.

The analysis of multi-modal data is studied widely in numerous types of data e.g. text, image, video, medical image, and exceptional data characteristics e.g large-scale data [5]–[11]. The higher variability of multi-modal data compared to unimodal data becomes a unique challenge that requires special treatment to learn [12], [13]. In many cases, a regular classification model can't fit the multi-modal data properly. As the result, the performance of the classifier dropped by a significant margin. Therefore, to deal with multi-modal data classification, a classification model needs to be enhanced to maintain its performance. Multi-modal data also elevate the non-linearity of the data itself. A unimodal data that is linearly separable can't stay linearly separable if we add another modality to its features. While unimodal data that already need a nonlinear classifier to seperate its classes will need a classifier with higher non-linear capability when the data produces another modality in its features. Therefore, multi-modality is in line with non-linearity.

Improving a classifier's performance in multi-modal data classification can be conducted by several strategies i.e. ensemble learning, incremental learning, and clustering [14]–[17]. The clustering approach utilizes the clustered data to build several prototypes (reference vectors) of the models before the training phase. During the training phase, the prototypes are updated accordingly following the update rule of the base model. The clustering approach adds computation cost before training, not within the training phase as applied in the incremental learning approach. This study is focused on enhancing the model by using the clustering approach. In the previous study, clustering confirmed that it can enhance the neural network model [14-15]. However, there is still room for improvement e.g. finding the optimum number of clusters, finding the better clusters' members, and improving the accuracy margin.

This study is a continuation of the previous study to develop a neural network model for multi-modal data classification. In the previous study, we proposed an enhanced version of learning vector quantization (LVQ)-based neural networks by generating multiple reference vectors (codebook) using the clustering approach [14], [15]. In the study, we utilized K-Means clustering, intelligent K-Means clustering based on anomalous patterns, and a parametric version of Gaussian mixture model (GMM) clustering [18]–[20]. In the previous study, we also tried to develop a variant of intelligent K-Means clustering to enhance the neural network's performance [21]. In the previous study, we applied two variants of intelligent K-Means clustering to generate the multi-codebook neural network models [15]. The models worked excellently in the dataset where all features have multimodal distribution. However, in the dataset where only a few features have multimodal distribution, the models still need improvements. In some cases (datasets) the proposed method outperformed the base neural network model but, in some cases, (datasets) the proposed models were outperformed by the base model. We need to develop a multi-codebook neural network that has better performance than the base model (single codebook) both on the datasets where all features have multi-modal distribution and, in the datasets, where a few features have multi-modal distribution. On the other hand, our previous study shows that multi-codebook neural networks using standard (parametric) GMM performed well in normal distribution data [14]. However, we need to find the best cluster number. Rather than finding the best cluster by brute force experiments, in this study, we proposed an intelligent Gaussian mixture model clustering where we do not need to specify the number of mixtures/clusters. The number of mixtures is approximated by using histogram analysis of the input dataset. Then we proposed a multi-codebook neural network by using intelligent GMM clustering for multi-modal classification. The main contributions of this paper are as follows: (1) Propose a new intelligent Gaussian mixture model clustering based on histogram information (2) Propose a new multi-codebook neural network by using intelligent Gaussian mixture model clustering (3) Propose three variants of the multi-codebook neural network mentioned in point 2. As an additional contribution, we have applied the multi-codebook based on intelligent GMM clustering in 2 neural network models. Then we continued our study by comparing the proposed method with commonly used classical machine learning i.e. Naive Bayes, support vector machine (SVM) by using radial basis function (RBF) kernel, multi-layered perceptron (MLP), Bagging tree, and random forest as the representation of ensemble learning [22]–[27]. Afterward, we compared the proposed well-known deep learning method, especially the convolutional neural network (CNN) [28]. Last, we evaluate the improvement of the proposed method by using ANOVA statistical testing to prove its significant improvements compared to the original methods.

In this study, we applied the intelligent Gaussian mixture model clustering into fuzzy neuro generalized vector quantization (FNGLVQ) as the base classifier. FNGLVQ is a simple neural network that consists of one input layer, one output layer, and one hidden layer. FNGLVQ is developed from generalized vector quantization (GLVQ) by using a fuzzy approach [29]. The method utilizes a fuzzy triangle membership function (min, mean, max) instead of one single value on its reference vector. The main idea of the method is to enhance the performance of GLVQ in high overlapping data. GLVQ is an enhancement of the LVQ2.1 model to decrease its misclassification error during the training phase [30], [31]. The method uses stochastic gradient descent

(SGD) optimization to minimize its error from time to time. In this study, we have developed multi-codebook FNGLVQ by using intelligent GMM as the main proposed classifier. However, we also have developed multi-codebook GLVQ by using an intelligent GMM classifier alongside the main proposed method.

This paper is written into eight sections. This first section discusses the introduction of the study. The second section discusses the previous study on multi-modal and multi-codebook approaches related to this study. The third section discusses the idea of multi-codebook fitting for multi-modal data classification. The fourth section explains FNGLVQ which is utilized as the base classifier within the proposed methodology. Section five discusses the proposed intelligent Gaussian mixture model clustering continued by multi-codebook FNGLVQ by using intelligent GMM clustering in section six. Section seventh discusses the experiment result and analysis. The last (eighth) section is the conclusion of the study.

II. RELATED WORKS

Classification in multi-modal data becomes a challenge within the field of machine learning. many studies are conducted to face the challenges of multi-modal information. As mentioned in the previous section, there are three approaches for enhancing a classification model to deal with multi-modal data i.e. ensemble learning, incremental learning, and clustering. Ensemble learning is an approach to enhance a classifier's performance by combining several based models called weak classifiers into one unified strong classifier. The combination method to unify the base models can be conducted by using simple (uniform) voting, weighted voting, bootstrap aggregating (bagging), boosting, and other variations [26-27]. In ensemble learning, we have $h_1, h_2, h_3, \ldots h_n$ weak classifiers a strong classifier $H=f(h_1, h_2, h_3, \ldots h_n)$. Weak classifiers can be any classifier we have already known like decision tree, SVM, perceptron, Naïve Bayes, etc. While the strong classifier is the union of all weak classifiers. The result of the strong classifier can be computed by using simple voting (majority vote) or weighted voting where each weak classifier has its weight in contributing to the classification result. Some of the ensemble learning such as random forest use bootstrapping method to resample its training data so its weak classifiers have different training (resample) data. Incremental learning is an approach to enhance a classifier's model by gradually improving its structure during the training phase [15]. The improvement of its structure can be adding/deleting layers, adding/deleting neurons in a layer, or other operations. By using the approach, the model is expected to have a more fitting structure at the end of the training phase compared to at the beginning of the training phase. In incremental learning, let's say initially we have $W=\{W_1, W_2, W_3\}$ as the weight of a classifier. After T epochs, the structure of the wight can be restructured into $W=\{\{W_{11}, W_{12}\}, W_2, W_3\}$, then after another U epochs, it evolves into $W=\{\{W_{11}, W_{12}\}, W_2, \{W_{31}, W_{32}, W_{33}\}\}$, and so on. The clustering approach utilizes groups of data (clusters) to form multiple prototypes (codebooks) for the classifier model. An enhancement of a classifier by using the clustering approach is explained in section 6.

In real applications, the study and analysis of multi-modal data can be found in various research areas. Soleymani et al conducted a survey relating to multi-modality in sentiment analysis issues [6]. Corneanu et al analyzed multi-modal information to enhance a classification model in expression recognition [5]. Oskouie conducted a study concerning multimodal feature extraction and fusion for linguistics mining of football match video [8]. Kumar et al conducted a survey relating to multi-modality and multi-dimensionality in medical imaging data such as magnetic resonance imaging (MRI) and computed tomography scan (CT-Scan) [7]. Zhang et al studied multi-modality in Alzheimer's disease analytics to enhance the model's ability in detecting the disease [32]. Cauwenberghs et al used an incremental learning approach to boost the support vector machine (SVM) classifier [33]. Molina et al additionally developed an analogous approach for adenocarcinoma disease classification [34]. Huang et al enhanced extreme learning (ELM) by using a similar approach [35]. As for clustering approaches enhancement, many studies were conducted before. Ma'sum et al used GMM, K-Means, and IK-Means to boost LVQ primarily based on neural networks [14]. In another research, Ma'sum et al combined unsupervised extreme learning machines (US-ELM) with intelligent clustering to improve neural networks algorithm performance [36]. The different approach for enhancing classifier performance was conducted by using an ensemble learning approach as utilized by Krawczyk et al and Ortiz et al [17][37]. during this study, we tend to solely discuss intelligent clustering especially intelligent Gaussian mixture model (IGMM) clustering that is proposed in this study.

As mentioned before, this study uses FNGLVQ because of the base classifier. FNGLVQ is an improved version of GLVQ wherever GLVQ is improved from LVQ. LVQ could be a shallow neural network classifier that uses the winner take al principle [29]. The winner is the class (category) that its reference vector is nearest (closest/ most similar) to the input vector. During the training and testing phase, LVQ utilizes the winner vector as the main process. Originally, LVQ is proposed by Kohonen et al, It is the supervised version of the self-organizing map (SOM). A few years after its initial development, LVQ has been developed into many variants e.g. LVQ1, LVQ2, LVQ2.1, and LVQ3. The variants of LVQ were developed by employing a limitation on the update rule equation known as a window for change conditions within the learning method. LVQ2.1 was then increased by adding improvement procedures leading to GLVQ [31]. The improvement is applied throughout the learning phase on GLVQ. Therefore, the error rate is reduced from iteration to iteration. In several cases, the performance of GLVQ is superior to any or all variants of LVQ. GLVQ was enhanced by Setiawan et al by adding a fuzzy triangle membership function on its reference vector/codebook [29]. FNGLVQ was originally utilized for arrhythmia classification. The method analyzed an electrocardiogram signal taken from a patient, then predict the heart condition of the patient. Another similar development of the LVQ classifier is conducted by Kusumoputro et al that applied the fuzzy triangle membership function in LVQ, instead of in LVQ2.1 or GLVQ. The development resulting a new method named Fuzzy Neuro Learning Vector quantization (FNLVQ) [38]. Later, FNLVQ was improved by using an optimization approach to gain better performance. The method was an improvement by using a nature-based optimization method called particle swarm improvement (PSO) leading to FNLVQ-PSO [39]. Another study to modify GLVQ was conducted by using a multi-layered approach. The enhanced method was named AMGLVQ which was proposed to integrate feature extraction and classification as one method [40].

One of the rapid developments of neural network models is categorized as deep learning. Deep learning uses complex (deep) architecture to increase the separation ability of the model. Many of the deep learning models were applied in classification with so many labels e.g labels 1000 One of the well-known approaches in deep learning method is convolutional neural networks where the models have convolutional layers along with fully connected layers [41]. One of the popular models of CNN named as VGG (adapted from the visual geometry group) model [28]. The method has very deep convolutional layers that are proposed to solve image classification with 1000 classes and millions of instances. Another notable development of CNN is residual neural networks (ResNet), densely connected convolutional networks (DenseNet) Neural architecture Search Networks (NASNet), and MobileNet [42]–[45]. All of the explained models above are proposed to solve image classification problem. We also have utilized the well-known deep learning model to compare their performance with the proposed methods. The deep learning models require minimal (32,32,1) size data, as the models originally were proposed for the image classification. In this study, we resized the data dimension to satisfy the model requirements.

Meanwhile, clustering algorithms have been developing rapidly. The vast development of deep learning also takes an impact on clustering method development. A few of the recent development in clustering methods are deep-learning-inspired clustering. Bo et al proposed a structural deep clustering method that uses structure as used in deep learning such as convolution and graph [46]. Lim et al proposed deep clustering with variational autoencoder [47]. While Yang et al proposed a robust deep clustering by using adversarial learning [48]. The other development of clustering methods is non-parametric clustering. Ni et al proposed a nonparametric clustering by using the Bayesian principle [49]. The method was proposed for classification along with clustering. D'Errico et al proposed non-parametric density peak clustering that was applied for high-dimensional datasets [50]. The other development of clustering is multi-view clustering. It is a type of clustering to learn an informative and consistent representation among different views where the label of the datasets is not utilized in learning. Lin et al proposed incomplete multi-view clustering via contrastive prediction (COMPLETER) [51]. Huang et al proposed partially view-aligned clustering [52]. While Peng et al proposed multi-view clustering without parameter selection (COMIC) [53].

III. MULTI-CODEBOOK FITTING FOR MULTI-MODAL DATA CLASSIFICATION

As mentioned in the introduction section, multi-modal data may be data wherever its attributes are spread in many areas. The illustration of multimodal data is shown in figure 1. The figure shows the scatter plot and distribution of 2 classes (categories), let's say A (orange) and B (blue) in two cases. The figure shows that in case-1 the attribute of class-A is spread in 2 areas, whereas the attribute of class-B is spread only in one area. The figure shows that class-A has multi-modal distribution whereas class-B has unimodal distribution.

The figure also shows that the attribute of class B is in the middle of the attribute of class-A spreads in two areas. The fuzzy neural network model will fit the distribution by using a fuzzy triangle. The symbol a, b, and c represents the fuzzy membership value (min, mean, max) for each triangle. The figure shows that in single-codebook fitting, the model generates one fuzzy triangle for each class. While in multi-codebook fitting, the model generates two fuzzy triangles for class-A. The figure shows that multi-codebook fitting produces less inter-class overlapping area than single-codebook fitting. The similarity between the 2 classes shown by the yellow line emphasizes that the overlapping in single-code fitting is higher than in multi-codebook fitting as h12 is higher than h23 and h24. The figure also shows the other condition where both class-A and class-B have a multi-modal distribution that is shown in case-2. The model generates two fuzzy triangles for each class in multi-modal fitting mode, whereas the model generates one codebook for each class in the single-codebook fitting model. The figure shows that in case-2 as well, the multi-codebook fitting produces less inter-class overlapping than the single-codebook fitting. The similarity line also shows the same idea, it is shown by the yellow line h35, h35, and h46 have a lower height than h12. In a formal definition, suppose that a neural network is represented as (I, W, O) where I is the input layer, W the is hidden layer, and O is the output layer. In the case of FNGLVQ, the single-codebook hidden layer model is represented as $W=\{W_1, W_2, W_{1,\dots} W_m\}$. W_i is the codebook or reference vector for class i, $W_i = (w_{i1}, w_{i2}, w_{i3 \dots} w_{in})$, $w_{ij} = (w_{ij\_min}, w_{ij\_mean}, w_{ij\_max})$ where m is the number of classes, and n is the number of features. While in the multi-codebook version of FNGLVQ, the model is represented as $W=\{\{W_{11}, W_{12}, W_{13,\dots} W_{1c1}\}, \{W_{21}, W_{22}, W_{23,\dots} W_{2c2}\}, \dots \{W_{m1}, W_{m2}, W_{m3,\dots} W_{1cm}\}\}$ where $\{W_{i1}, W_{i2}, W_{i3,\dots} W_{1c1}\}$ is a set of codebooks or reference vectors of class i. Each class may have multiple codebooks. The number of the codebook of class i is notated as ci i=1, 2, 3, …, m, where c1, c2,…, and cm may have different values. In the multi-codebook version, W_{ic} has a similar representation as W_i in the single-codebook version, $W_{ic} = (w_{ic1}, w_{ic2}, w_{ic3,\dots} w_{icn})$, $w_{icj} = (w_{icj\_min}, w_{icj\_mean}, w_{icj\_max})$ where n is the number of features, c is the c-th codebook of class i.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

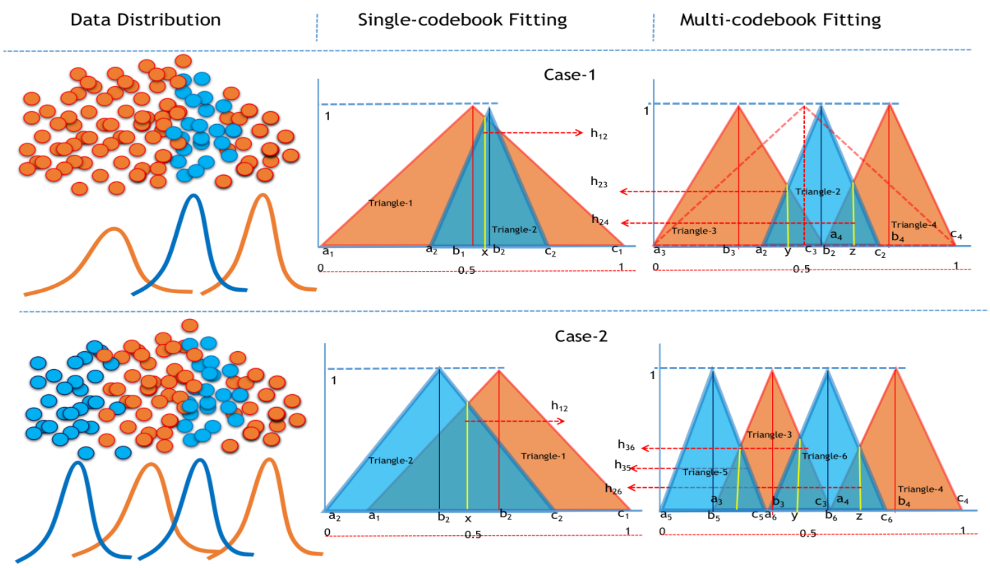


FIGURE 1. **Single-codebook vs multi-codebook fiting for multi-modal data classification**

The architecture of a single codebook is illustrated in figure 2, while the architecture of a multi-codebook is shown in figure 4. The representation for a multi-codebook version of other base models e.g. GLVQ and LVQ follows the representation of the base models themselves.

IV. FUZZY-NEURO GENERALIZED LEARNING VECTOR QUANTIZATION (FNGLVQ)

Fuzzy-Neuro Generalized Learning Vector Quantization (FNGLVQ) is an improved version of Generalized Learning Vector Quantization (GLVQ) by applying fuzzy representation for its reference vector (codebook). Each neuron in FNGLVQ has three values (min, mean, max) while in GLVQ each neuron only has one value. The fuzzy approach is applied in FNGLVQ to deal with high overlapping data. FNGLVQ is proposed by Setiawan et al, which is originally aimed for arrhythmias classification based on electrocardiogram (ECG) signal. [3]. GLVQ is the improvement of LVQ2.1 (a variant of LVQ) by using stochastic gradient descent to minimize its misclassification error during the training phase. LVQ2.1 is a variant of LVQ that utilized a limitation called window in its update rule. The idea of applying the window is to prevent the abnormal update of the model. The other modification that was initiated in LVQ2.1 is using two central vectors for update named winner vector (usually denoted as w1) and runner up vector (usually denoted as w2)

The architecture of the FNGLVQ classifier is illustrated in Figure. 2. The figure shows that the vector x is the associate degree input for the rule. within the FNGLVQ, every class has a weight called a reference vector or codebook illustrated by triangle shapes. Each class i has n weights ($w_{i1}, w_{i2}, w_{i3},\ldots w_{in}$), where n represents the number feature of the datasets. that the range of options of the dataset. Please note that w_{i1} and w_{i2} during this context do not represent the winner and runner-up vectors, but reference vector for feature-1 and feature-2. Each reference vector w_{ij} represented as tuple of three values ($w_{ij\text{-min}}, w_{ij\text{-mean}}, w_{ij\text{-max}}$) as adapted from the fuzzy triangle membership function. During the training phase, the method takes an input vector x, then the model computes the distance between $x = (x_1, x_2, x_3,\ldots x_n)$, and the reference vector of each class i $w_i = (w_{i1}, w_{i2}, w_{i3},\ldots w_{in})$, where i is the index of class, and n is the number of features. The model then finds the winner (w_1) and runner-up vector (w_2). Please note that index 1 and 2 in this case are used for denoting winner and runner-up vectors, not class labels. The winner vector is the nearest (most similar) vector from the same class label, whereas the runner-up vector is the nearest (most similar) vector from a different class label. The winner and runner-up vector will be updated by using the rule explained in equations (1) to (20). The architecture of FNGLVQ is illustrated in Figure 2. The figure shows that the vector x is the associate degree input for the rule. within the FNGLVQ, every category has weight illustrated by a triangle form within the figure. Each class i has n weights $w_i = (w_{i1}, w_{i2}, w_{i3},\ldots w_{in})$, where n is the range of options of the dataset. Please note that w1 and w2 during this context area unit the weight notation of feature-1 and feature-2 of their class, not the symbols of the winner and runner-up vectors as mentioned before. Each weight vector w_{ij} has 3 values ($w_{ij\_min}, w_{ij\_mean}, w_{ij\_max}$) because the technique uses fuzzy membership. within the learning method, the strategy computes the space between input vectors and also the weights of every class. because each data has n features, the measurement of the distances is conducted for all features, then the mean of the distances is calculated. Then the FNGLVQ method finds the winner category and runner-up classes supported the distance of the categories to the input vector. Last, the method updates the values of the winner vector and runner-up vectors.

GLVQ classifier applied misclassification error measurements as written within the equation (1), where d1, and d2 as the distance between input and winner vector and the distance between input and runner-up vector

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

respectively. FNGLVQ adapts the approach from GLVQ. In the training phase, the FNGLVQ technique uses similarity value instead of distance that is adapted from fuzzy similarity measurements (μ) to determine the winner vector and runner-up vector. Therefore, the value of the distance is represented by using similarity (μ) resulting in d = 1-μ.

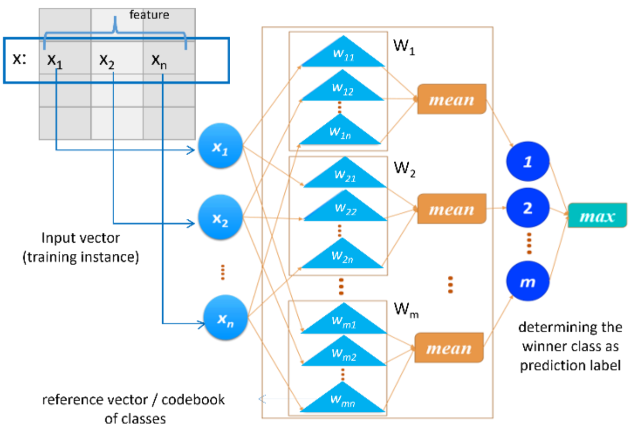


FIGURE 2. **Architecture of Fuzzy Neuro Generalized Learning Vector Quantization (FNGLVQ).**

Substituting the similarity into equation 1 resulting new misclassification error formula as shown in equation 2. The formula in equation 2 is used by FNGLVQ in its training process.

$$\varphi(x) = \frac{d_1 - d_2}{d_1 + d_2} \qquad (1)$$

$$\varphi(x) = \frac{\mu_2 - \mu_1}{2 - \mu_2 - \mu_1} \qquad (2)$$

The similarity value in FNGLVQ is computed by the equation below.

$$\mu = h(x, w_{min}, w_{mean}, w_{max}) = \begin{cases} 0 & , if \;\; x \leq w_{min} \\ \dfrac{x - w_{min}}{w_{mean} - w_{min}}, & if \; w_{min} \leq x \leq w_{mean} \\ \dfrac{w_{max} - x}{w_{max} - w_{mean}}, & if \; w_{mean} \leq x \leq w_{max} \\ 0 & , if \;\; x \geq w_{max} \end{cases} \qquad (3)$$

FNGLVQ uses the cost function as defined in GLVQ to minimize the misclassification error during the training process. The cost function (S) is defined in the equation below

$$S = \sum_{n=1}^{N} f(\varphi(x)) \qquad (4)$$

The symbol f represents a monotonically increasing function e.g. sigmoid function and N is the number of iterations (epoch) in the training phase.

$$w_i(t+1) \leftarrow w_i(t) - \alpha \frac{\delta s}{\delta w_i(t)}, \text{where } i = \; 1{,}2 \qquad (5)$$

The derivation of the cost function (S) to the weight (w) is defined by the equation below:

$$\frac{\delta s}{\delta w_i} = \frac{\delta s}{\delta \varphi} \cdot \frac{\delta \varphi}{\delta \mu} \cdot \frac{\delta \mu}{\delta w_i} \qquad (6)$$

FNGLVQ method uses a fuzzy triangular function for its reference vectors (codebooks). Therefore, the update rule of the FNGLVQ in equation 6 is derived from the three conditions below.

- If $w_{mean} < x \leq w_{mean}$

$$w_1(t+1) \leftarrow w_1(t) - \alpha \cdot \frac{\delta f}{\delta \varphi} \cdot \frac{2.(1-\mu_2)}{(2 - \mu_1 - \mu_2)^2} \cdot \left(\frac{x - w_{min}}{(w_{mean} - w_{min})^2}\right) \qquad (7)$$

$$w_2(t+1) \leftarrow w_2(t) + \alpha \cdot \frac{\delta f}{\delta \varphi} \cdot \frac{2.(1-\mu_1)}{(2 - \mu_1 - \mu_2)^2} \cdot \left(\frac{x - w_{min}}{(w_{mean} - w_{min})^2}\right) \qquad (8)$$

- If $w_{mean} < x < w_{max}$

$$w_1(t+1) \leftarrow w_1(t) + \alpha \cdot \frac{\delta f}{\delta \varphi} \cdot \frac{2.(1-\mu_2)}{(2 - \mu_1 - \mu_2)^2} \cdot \left(\frac{w_{max} - x}{(w_{max} - w_{mean})^2}\right) \qquad (9)$$

$$w_2(t+1) \leftarrow w_2(t) - \alpha \cdot \frac{\delta f}{\delta \varphi} \text{x} \frac{2.(1-\mu_1)}{(2 - \mu_1 - \mu_2)^2} \cdot \left(\frac{w_{max} - x}{(w_{max} - w_{mean})^2}\right) \qquad (10)$$

- If $x \leq w_{min}$ and $x \geq w_{max}$

$$w_i(t+1) \leftarrow w_i(t) \qquad , i = 1{,}2 \qquad (11)$$

where w_1 (winner class) is the nearest (most similar) reference vector (codebook) from the same class as input vector $C_x = C_{w1}$, and.

$$w_{min} \leftarrow w_{mean}(t+1) - \big(w_{mean}(t) - w_{min}(t)\big) \qquad (12)$$

$$w_{max} \leftarrow w_{mean}(t+1) - \big(w_{mean}(t) - w_{min}(t)\big) \qquad (13)$$

The symbol α represents the learning rate in FNGLVQ that is computed by equation below.

$$\alpha(t+1) = \alpha_0 \cdot \left(1 - \frac{t}{t_{max}}\right) \qquad (14)$$

FNGLVQ defines additional rules to adjust w_{min} and w_{max} to gain better performance as follows.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

If ($\mu 1 > 0$ or $\mu 2 > 0$), and $\varphi < 0$ then the FNGLVQ will increase fuzzy triangular width by using the equation below.

$$w_{min} \leftarrow w_{mean} - (w_{mean} - w_{min}).(1 + (\beta.\alpha)) \quad (15)$$

$$w_{max} \leftarrow w_{mean} + (w_{max} - w_{mean}).(1 + (\beta.\alpha)) \quad (16)$$

Where α is learning rate and β is the fuzzy triangle adjustment factor. Both α and β have values in range [0,1].

If input data is predicted into the wrong class ($\phi \geqslant 0$), then the method will decrease the triangular width using the equations below.

$$w_{min} \leftarrow w_{mean} - (w_{mean} - w_{min}).(1 - (\beta.\alpha)) \quad (17)$$

$$w_{max} \leftarrow w_{mean} + (w_{max} - w_{mean}).(1 - (\beta.\alpha)) \quad (18)$$

If $\mu 1 = \mu 2 = 0$, then fuzzy triangular vectors must be adjusted using equation below. γ is constant, the γ value used in this research is 0.1.

$$w_{min} \leftarrow w_{mean} - (w_{mean} - w_{min}).(1 - (\gamma.\alpha)) \quad (19)$$

$$w_{max} \leftarrow w_{mean} + (w_{max} - w_{mean}).(1 + (\gamma.\alpha)) \quad (20)$$

From the derivation above, we summarize three important points of the FNGLVQ algorithm. First, computing the membership function of the input vector to the winner vector ($\mu 1$) and runner-up vector ($\mu 2$) by using equation 3. Second, adjusting the position of the winner vector and runner-up vector regarding the input vector by using equations 7-14. Third, adjusting the size of the winner and runner-up vector by using equations 15-20. The combination of the equations will lead to model convergence during the training process. The **training process** of the FNGLVQ is conducted by the following steps:

(0). Given the training set consisting of m instances of the training set (x1, x2, ... xm). Each instance is a vector of n elements, where n is the features of each instance
(1). Initiate the codebook (weight/reference vector) of each class (w) by using all training set instances or a random selection of training sets for the respective class.
(2). For each instance of the training data, train the weights of the winner and runner-up vector by using equation 3 and equation 7-20
(3). Repeat step 2 until the T epoch (iteration) as defined

The **testing process** of the FNGLVQ classifier is conducted by measuring the similarities between the input vector and the codebooks of the classes using equation (3). Then the method takes the class whose nearest/ most similar codebook to the input as the predicted class.

V. PROPOSED: INTELLIGENT GAUSSIAN MIXTURE MODEL (IGMM) CLUSTERING BASED ON HISTOGRAM INFORMATION

Clustering is a supervised approach in machine learning that is utilized to group a set of data into several groups called clusters. Clustering learns from the data without labels as It is an unsupervised method, and clusters the data only based on its attributes (feature). There are well-known clustering methods such as K-Means, Gaussian Mixture Model (GMM), and DBScan. Each method has a unique idea to cluster the data.

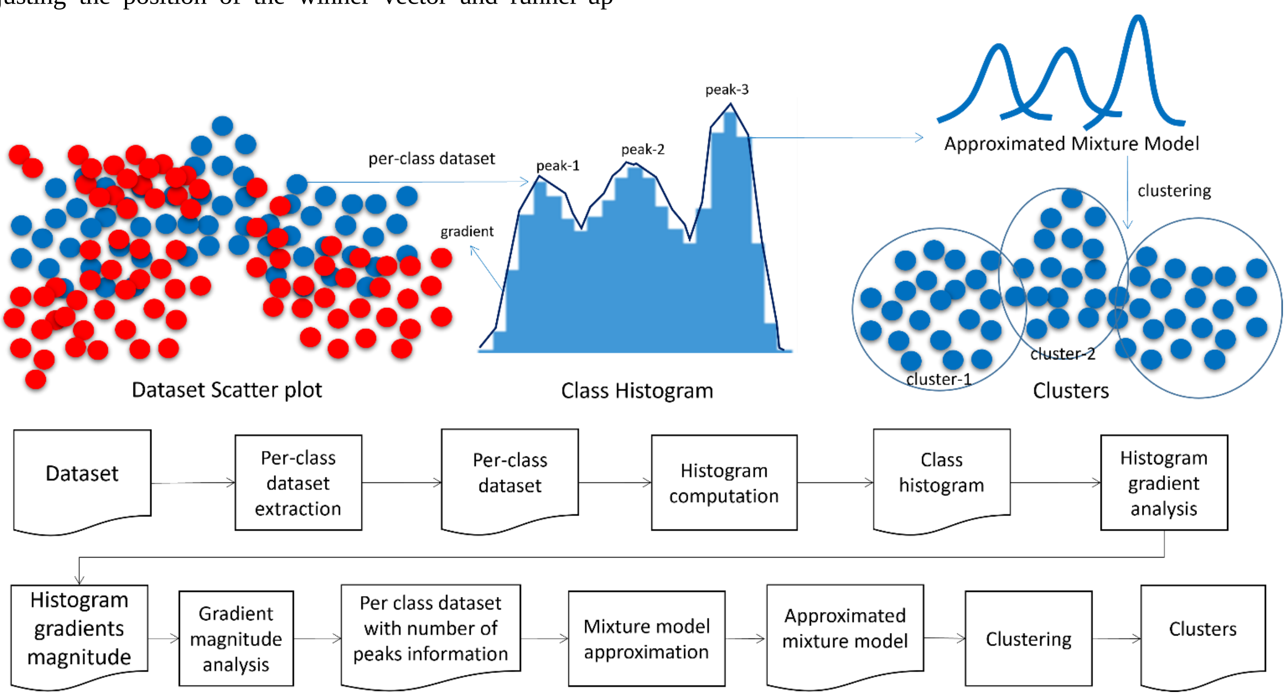


FIGURE 3. **Intelligent GMM based on Histogram Information**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

K-Means use the distance of each instance to centroids to determine the cluster where the instance will be grouped, GMM uses the highest probability if each instance to the mixture models to cluster, while the DBScan uses the density metrics. Normally GMM requires the number of mixture model/cluster (K) to cluster the data, however, finding the best K of all possible K requires high-cost experiments. Therefore, in this study, we proposed a new intelligent GMM that automatically determines the number of clusters.

Algorithm 1: Intelligent GMM Clustering based on Histogram Information

```
01: procedure IGMM-HistogramInformation
02:   Given f1, f2, f3, ….fm as features/attributes
03:   maxPeak=0 as the current maximum peak found
04:   for c=1:numberOfClasess do
05:     peaki=ApproximateHistogramPeak (fi)
06:     If peaki > maxPeak then
07:       maxPeak = peaki
08:   end for
09:   do GMM clustering with K = maxPeak
10:
11: procedure ApproximateHistogramPeak (feature)
12:   sign_1 = 0, sign_2 = 0, numPeak = 0
13:   histVal = histogram(feature) // array of histogram
      values
14:   for i =2 : sizeOf(histVal)-1 do
15:     sign_2 = sign_1
16:     If histVal(i) > histVal(i-1) then
17:       sign_1 = +1
18:     else If histVal(i) < histVal(i-1) then
19:       sign_1 = -1
20:     else
21:       sign_1 = 0
22:     if sign_2 == 1 and sign_1 == +1 then
23:       numPeak = numPeak + 1
24:   end for
25:   return numPeak
```

In this study, we developed an intelligent GMM clustering by using histogram information. The main idea of the proposed clustering is to find the number of the cluster by analyzing the data histogram. The flow diagram of the proposed intelligent GMM is shown in figure 3. Given a dataset, the method will extract the instances of each class. Then the method computes the histogram of the dataset. In the next step, the method analyzes the gradient of the histogram data. The gradient is the change of value (frequency) from one bin to the next bin in the histogram. By this mechanism, the method can find the number of peaks in the data. A peak is found when the gradient of the histogram changes from positive (rising) to negative (falling). The number of peaks (let's say k) is imported as the number of mixtures. Then, the method generates k Gaussian mixtures and forms clustering. Finally, the method produces k clusters of the dataset. The steps of intelligent GMM clustering based on histogram information are written in algorithm 1.

VI. PROPOSED METHOD: MULTI-CODEBOOK FUZZY NEURO GENERALIZED LEARNING VECTOR QUANTIZATION BY USING INTELLIGENT GAUSSIAN MIXTURE MODEL BASED ON HISTOGRAM INFORMATION (MC-FNGLVQ-IGMM-HIST)

A. ARCHITECTURE

The architecture of multi-codebook FNGLVQ busing intelligent GMM clustering based on histogram information is adapted from the architecture of the original FNGLVQ. The multi-codebook FNGLVQ has 3 layers i.e. input layer, hidden (middle) layer, and output layer. The input layer acquires an input sample for the classifier, the hidden layer contains the reference vectors (codebooks) for every class, whereas the output layer computes to find the winner vector whether for codebook updates(training) or prediction (testing). the most distinction between the multi-codebook FNGLVQ and the original FNGLVQ is in its hidden layer. the original version of FNGLVQ has one reference vector (codebook) for each class, whereas the multi-codebook version of FNGLVQ supports many codebooks for all classes. the number of codebooks between one class and another class may completely different. for instance, classes A and B have 4 codebooks, however, class C has 3 codebooks, and class D has only 2 codebooks. In a class is also possible to have 1 codebook. The architecture of the multi-codebook FNGLVQ by using intelligent GMM clustering based on histogram information is shown in figure 4.

In this study, we have proposed 3 variants of multi-codebook FNGLVQ by using intelligent GMM based on histogram information. The variants are developed based on lines 03 to 08 in algorithm 1. The versions are listed below.

1. Version 1 (V1): The number of mixtures k is assigned by the value of the maximum number of peaks from all class features.

$$k = \max(peak(fi)),\ \ i=1,2,...,n \qquad (21)$$

2. Version 2 (V2): The number of mixtures k is assigned by the value of the mean number of peaks from all class features.

$$k = \text{mean}(peak(fi)),\ \ i=1,2,...,n \qquad (22)$$

3. Version 3 (V3): The number of mixtures k is assigned by the value of the maximum number of peaks plus the mean number of peaks from all class features, then divided by 2.

$$k = (\max(peak(fi))+\text{mean}(peak(fi)))/2\ \ i=1,2,...,n \qquad (23)$$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

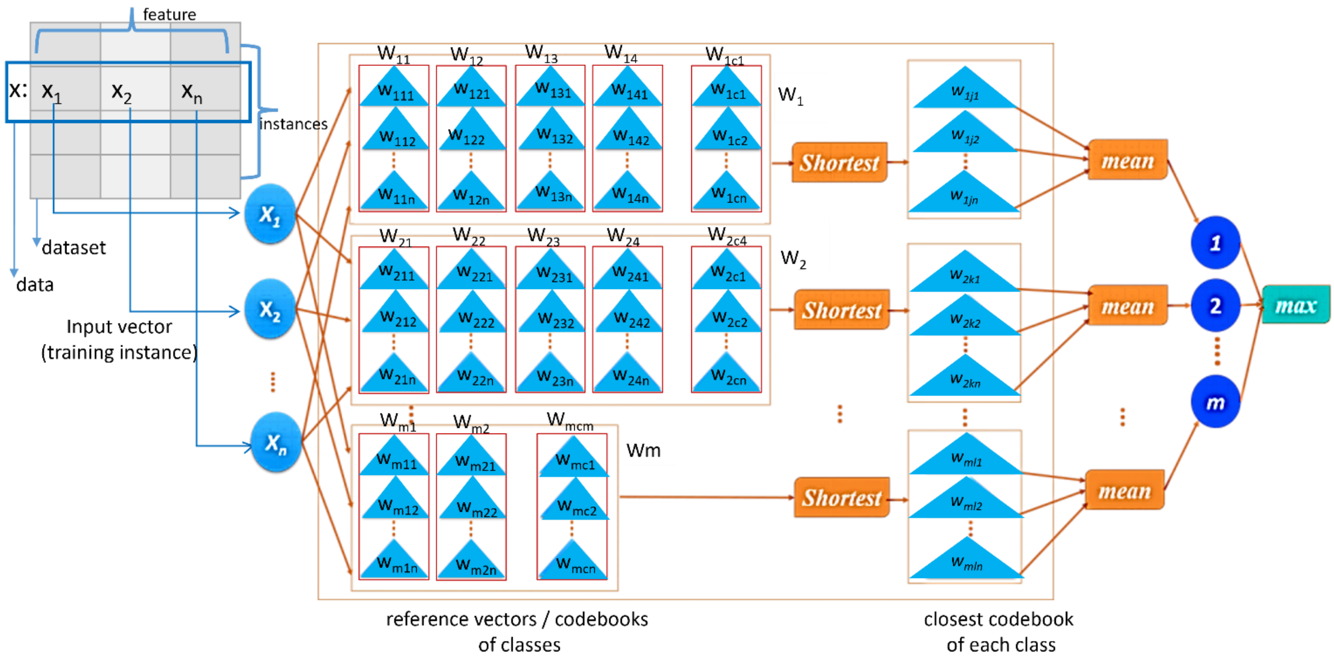


FIGURE 4. **Architecture of Multi-codebook FNGLVQ**

B. ALGORITHM

As mentioned in the previous sub-sections the idea of the proposed method is to generate multiple codebooks by using the clustering approach. In multi-codebook FNGLVQ by using intelligent GMM based on histogram information, the clustering is conducted before the training phase. Before the training phase, the method separates the data based on class labels. Then, each class data is clustered into several (k) clusters, where the k is automatically generated by the IGMM algorithm. In this study, we proposed three variations to compute the k value as explained in the previous section. Then the method generates a reference vector (codebook) from each cluster. The method will compute a tuple (min, mean, max) of the cluster members and use the tuple as a reference vector. This process is called fuzzification. If a class has k clusters, then the method will generate k codebooks for the class.

The process is repeated for all class labels of the dataset. Therefore, all classes have initial codebooks before the training phase. The next step is training the multi-codebook FNGLVQ IGMM based on the histogram model with the generated codebooks and given training samples. The training mechanism in the multi-codebook FNGLVQ IGMM Hist is the same as the training mechanism in the original FNGLVQ. During the training phase, the proposed method will find the winner and runner-up vector (codebook) from all available codebooks. Then the winner and runner-up codebook will be updated based on the three conditions as explained in section 4.

The testing process in the proposed multi-codebook FNGLVQ is also the same as the training process in the original version. The method will predict the input class label based on the winner codebook (reference vector) that is the closest/most similar codebook to the input. The steps of the proposed multi-codebook FNGLVQ are written in algorithm 2.

Algorithm 2: Multi-codebook Fuzzy Neuro Generalized Vector Quantization By Using Intelligent GMM Clustering (MC-FNGLVQ-IGMM-HIST)

01: **procedure** MC-FNGLVQ-CLUSTERING
02: **given** $x_1, x_2, x_3, \ldots x_n$ as training samples
03: **given** $f_1, f_2, f_3, \ldots f_m$ as features of training samples
04: **given** $c_1, c_2, c_3, \ldots c_y$ as class labels of training samples
05: **for** $c_i = c_1{:}c_y$ **do**
06: denote **k** as number of clusters
07: denote $f_c = f_1{:}f_m$ where $c_c = c_i$
08: do clustering to f_c, where k is a number of clusters as the result of intelligent Gaussian mixture model clustering (iGMM)
09: **for** j=1:k **do** //k=number of cluster
10: denote k_j as member of cluster-j
11: find min, mean, meax of k_j
12: Use the min, mean, max to generate FNGLVQ codebook
13: **end for**
14: **end for**
15: **for** $x_i = x_1{:}x_n$ **do**
16: train x by using FNGLVQ method
17: **end for**
18 Repeat step 15-17 until the number of iteration (epoch) is satisfied

TABLE 1
DETAIL INFORMATION OF DATASETS

No	Dataset	Information	#C	#F	#I
1	SyntA_i1	[2peak,5class,2feature]	5	2	10000
2	SyntA_i2	[2peak,5class,2feature]	5	2	10000
3	SyntA_i3	[2peak,5class,2feature]	5	2	10000
4	SyntA_i4	[2peak,5class,2feature]	5	2	10000
5	SyntB_i1	[2peak,2class,5feature]	2	5	4000
6	SyntB_i2	[2peak,3class,5feature]	3	5	6000
7	SyntB_i3	[2peak,4class,5feature]	4	5	8000
8	SyntB_i4	[2peak,5class,5feature]	5	5	10000
9	SyntB_i5	[3peak,2class,5feature]	2	5	6000
10	SyntB_i6	[3peak,3class,5feature]	3	5	9000
11	SyntB_i7	[3peak,4class,5feature]	4	5	12000
12	SyntB_i8	[3peak,5class,5feature]	5	5	15000
13	Pinwheel	[54]	5	2	5000
14	Glass	UCI	6	9	214
15	Ionosphere	UCI	2	33	351
16	Breast Cancer Coimbra	UCI	2	9	116
17	Wall	UCI	3	2	5456
18	Segment	UCI	7	19	2310
19	Ecoli	UCI	4	7	336
20	Odor	[39]	12	8	2400

VII. EXPERIMENT RESULT AND ANALYSIS

A. DATASET

In this study, we utilized synthetic datasets and benchmark datasets to evaluate the performance of the proposed methods. The synthetic dataset is a computer-generated dataset that has normal distribution and multimodality. class, The benchmark is taken from the UCI machine learning repository and datasets that were used in the previous study. The synthetic dataset is used to confirm the idea of a multi-codebook will fit the multi-modal distribution better than a single codebook. The synthetic dataset consists of two packages named Synthetic A (Synt_A) and Synthetic B (Synt_B). Bot of those sets consists of datasets with overlapping conditions between classes. However, datasets in the Synt_A package have higher overlapping than the datasets in the Synt_B package. The Synt_A package has 4 instances of datasets, where all of them have 2 peaks, 5 classes, and 2 features. While Synt_B package has 8 instances datasets where they have a different number of peaks, classes, and features. The distribution of Synt_A datasets is shown in figure 5. Benchmark datasets consist of 8 instances, 6 of them were taken from the UCI machine learning dataset, while two of them were taken from the referred previous study. The aim of evaluating the proposed methods by using benchmark datasets is to examine whether the performance of the proposed methods is better than the original version in real case data. The benchmark data may have multi-modality in its features and may not. More detailed information on the dataset utilized in this paper is delineated in Table 1. The symbols #C, #F, and #I within table 1 represent the number of classes, features, and instances respectively.

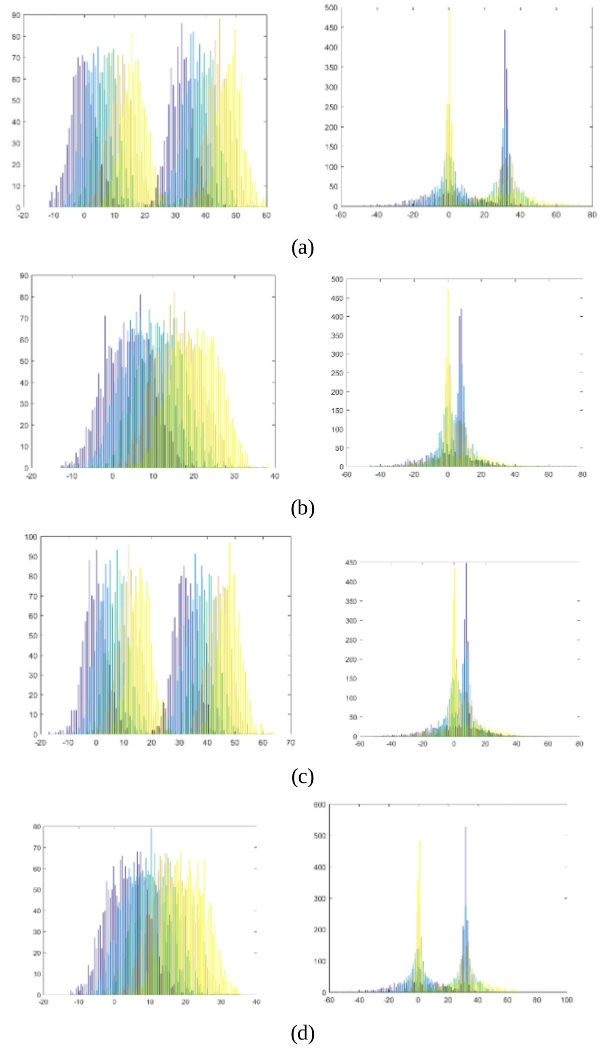


FIGURE 5. **Distribution of Synt_A dataset (a) instance 1, (b) instance 2, (c) instance 3, (d) instance 4**

B. EXPERIMENT SCENARIO AND SETUP

In this paper, we evaluated the three variations of the proposed method by using synthetic and benchmark datasets that are explained in the previous sub-section. The synthetic dataset has two packages named Synt_A and Synt_B. Synt_A consists of 4 instances of 2-peaks 5-classes datasets. Whereas the Synt_B package consists of 4 datasets that vary both in the number of peaks and the number of classes. The benchmark dataset has 8 instances that are taken from the UCI repository and previous research. These three packages' datasets are utilized to evaluate the performance of the multi-codebook FNGLVQ compared to the original version of FNGLVQ. We conducted three experiment scenarios, one scenario by using one package dataset respectively.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

The first scenario was conducted by using the Synt_A package that is conducted to confirm that the idea of multi-codebook architecture improves the performance of the base neural network. The second scenario was conducted to elaborate on the first experiment scenario. The second scenario was conducted in various peaks and a number of class datasets that are included in the Synt_B package. This scenario was conducted to observe whether the multi-codebook version was still superior compared to the original version. The third scenario was conducted by utilizing benchmark datasets. The scenario was conducted to confirm if the proposed methods achieve better performance in a real case problem compared to the original version. As mentioned in the previous section, we developed three variants of the multi-codebook neural network by using the intelligent Gaussian mixture model based on histogram information. As for the base classifier, we utilized two single hidden layered neural networks i.e. FNGLVQ and GLVQ. The evaluation metrics to evaluate the performance of the proposed method are accuracy and kappa coefficient. The formula of accuracy and kappa is written in the equation below.

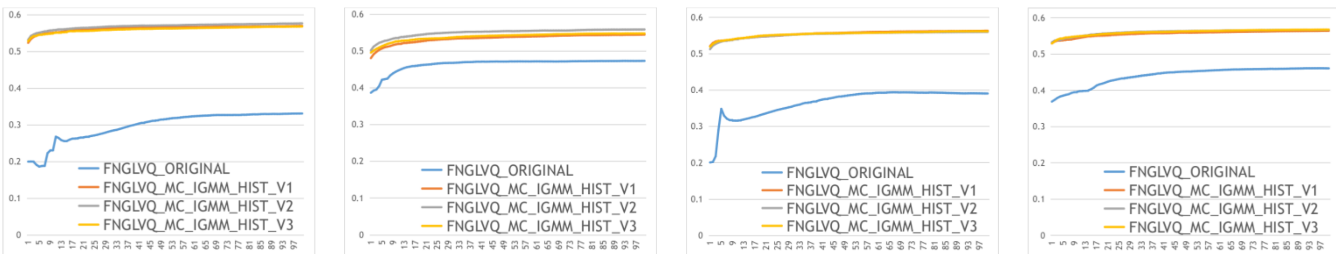
$$Accuracy = \left(\frac{C}{A}\right) = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \quad (24)$$

$$Kappa = \left(\frac{P0-Pe}{1-Pe}\right) \quad (25)$$

Accuracy is the ratio of true predicted samples to whole samples. In binary classification, true predicted samples consist of true positives (TP) and true negative (TN), while the whole samples consist of false positive (FP) and false negative (FP) alongside TP and TN. While in multi-class classification, accuracy is defined as the ratio of correctly predicted samples (C) to all samples (A). Accuracy is used as primary measurement in classification, however, It has weakness in imbalanced-class class classification. Therefore, we utilized kappa to confirm that is a more reliable representative in imbalanced-class classification besides representing the accuracy. Kappa is defined as the ratio of difference of the relative observed agreement among raters (p0) and the hypothetical probability of chance agreement (pe) to 1 minus the hypothetical probability of chance agreement (pe). In the classification problem, we treat class labels as observers for kappa measurements. The value of kappa is in the range of -1 to +1, whereas the value of accuracy is in the range of 0 to +1 (100%). For example, when we have 90 instances of class A and 10 instances of class B, then the classifier predicts all instances (100) as class A. The accuracy metrics will show 0.9 (90%) accuracy, while the kappa measurement will show 0.00 in the range of -1 to 1. In that case, the accuracy interprets that the model has good performance (90%) but the kappa interprets that the model has bad performance.

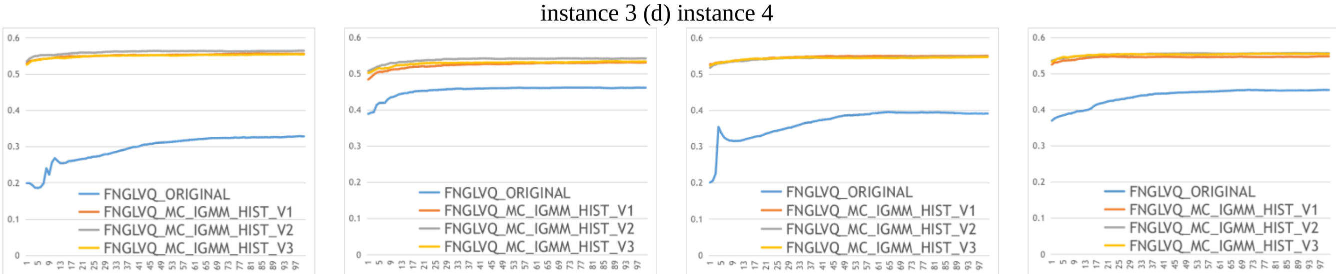
The GLVQ, FNGLVQ and their multi-codebook version were run by using learning rate (alpha) = 0.05, coefficient beta = 0.00005, gamma = 0.00005, and delta = 0.1. The coefficients are represented in equations (34) and (35). The methods were run for 100 iterations (epoch). The parameter is chosen as the recommendation for the methods based on previous studies [15]. All experiments were run by using 5-cross validation.

In the benchmark dataset, we compared the proposed method with the commonly known classifier i.e., Naive Bayes, Support Vector Machine (SVM) with radial basis function (RBF) kernel, multi-layered perceptron (MLP), bagging tree, and random forest. The MLP model has 3 layers and each layer has 10 neurons. The proposed method is also compared to deep learning methods i.e., convolutional neural networks (CNN) and fully connected networks (FCN).



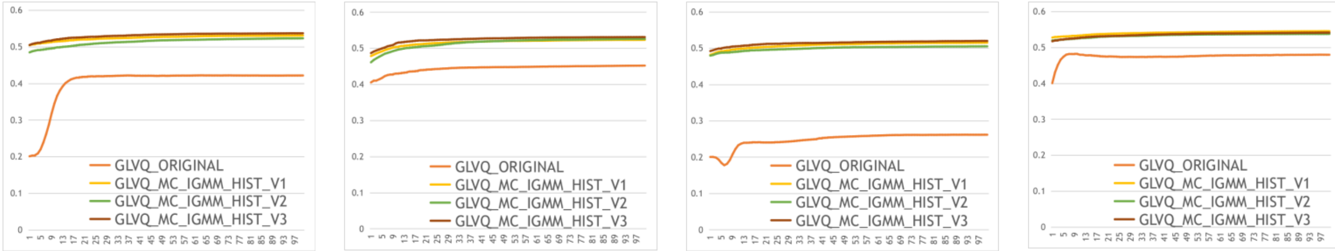
(a) (b) (c) (d)

Figure 6. Training Accuracy of Multi-codebook FNGLVQ using intelligent GMM in Synt_A datasets (a) instance 1 (b) instance 2 (c) instance 3 (d) instance 4



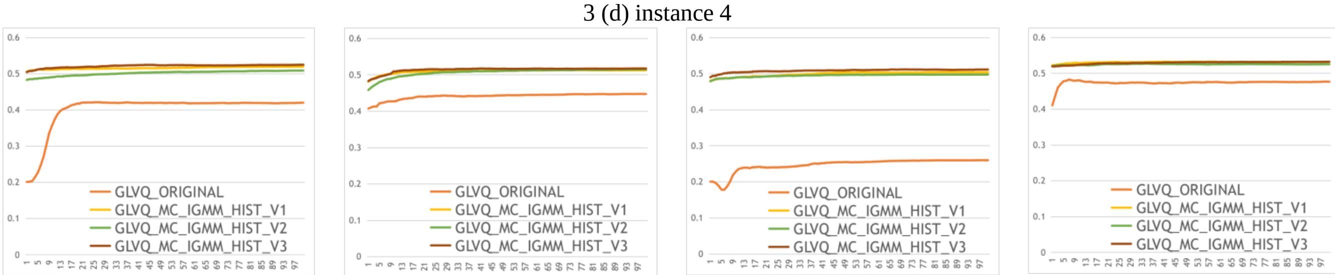
(a) (b) (c) (d)

Figure 7. Testing Accuracy of Multi-codebook FNGLVQ using intelligent GMMin Synt_A datasets (a) instance 1 (b) instance 2 (c) instance 3 (d) instance 4



(a) (b) (c) (d)

Figure 8. Training Accuracy of Multicodebook GLVQ using intelligent GMM in Synt_A datasets (a) instance 1 (b) instance 2 (c) instance 3 (d) instance 4



(a) (b) (c) (d)

Figure 9. Testing Accuracy of Multicodebook GLVQ using intelligent GMM in Synt_A datasets (a) instance 1 (b) instance 2 (c) instance 3 (d) instance 4

There are three variants of FCN that are compared in this study i.e., FCN2L (2 fully connected layers), FCN10L (ten fully connected layers), and FCN10L02 (ten fully connected layers) with dropout value=0.2. The other models except the last-mentioned model have dropout value=0.5 in their architecture. The proposed methods were also compared to the commonly known deep learning models i.e. AlexNet, VGG, DenseNet, ResNet, Inception, and MobileNet. For these models, instances of the input data need to be resized into (32,32,1) dimension, except the Inception that needs (75,75,1) minimal size for its input.

C. RESULT OF SCENARIO 1 : EXPERIMENT ON SYNT_A DATASETS

The first scenario is evaluating the proposed methods in the synthetic A dataset (Synt_A). The evaluation of the proposed method in the Synt_A dataset is proposed to confirm if the idea of the multi-codebook model is more suitable for multi-modal data classification than the original version.

Figures 6 and 7 show the accuracy of the three variants of multi-codebook FNGLVQ using intelligent GMM clustering in Synt_A datasets. The figures also show the performance of the original FNGLVQ. The figure shows that the three variations of the proposed method have achieved better accuracy compared to the original version. The figures also show that both training and testing have similar trends. In instance 1 and 3, The three variants of multi-codebook FNGLVQ achieved approximately around 55% while the original FNGLVQ only achieved no more than 40% both in terms of training accuracy and testing accuracy. In instance 2, the proposed methods achieved around 55% while the original version achieved no more than 48% accuracy. A similar trend is shown in the instance 4 dataset, where the prosed method achieved around 55% training and testing accuracy, while the proposed method achieved around 45% accuracy. Figure 6 and figure 7 also show that the proposed multi-codebook FNGLVQ is able to converge faster (less epoch) than the original version.

Along with multi-codebook FNGLVQ using intelligent GMM we also developed multi-codebook GLVQ using intelligent GMM. The multi-codebook GMM is also developed into three variants similar to the multi-codebook FNGLVQ. Figure 8 and figure 9 show that the proposed multi-codebook GLVQ achieved better performance than the original GLVQ. The trend is similar to the trend of multi-codebook FNGLVQ where the multi-codebook version gained better accuracy by a significant margin. The multi-codebook GLVQ achieved around 50%, 53%, 50%, and 51% accuracy in instance 1, instance 2, instance 3, and instance 4 respectively. While the original GLVQ achieved around 41%, 44%, 25%, and 42% accuracy in instance 1, instance 2, instance 3, and instance 4 respectively. Comparing those two base models, the multi-codebook FNGLVQ has slightly better performance than multi-codebook GLVQ in synthetic A datasets, while the original FNGLVQ and GLVQ have comparable accuracy. The detailed measurements of the performance of the classifiers in Synt_A datasets are shown in table 2 and table 3. Table 2 shows the recapitulation of accuracy measurements, while table 3 shows the recapitulation of kappa performance. Table 2 and Table 3 also show the comparison of the proposed method to the previous

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

TABLE II
ACCURACY (%) MEASUREMENT ON SYNTHETHIC A DATASET

Classifier\Dataset	Synt_A_1	Synt_A_2	Synt_A_3	Synt_A_4	Average
FNGLVQ_ORIGINAL	32.88	46.17	39.07	45.47	40.90
FNGLVQ_MC_KMEANS_C=2	54.08	52.60	53.98	54.42	53.77
FNGLVQ_MC_KMEANS_C=10	54.23	51.62	53.58	53.59	53.26
FNGLVQ_MC_IKMEANS_ANO	54.00	54.00	53.40	55.67	54.27
FNGLVQ_MC_IKMEANS_HIST	56.54	54.33	54.19	54.79	54.96
FNGLVQ_MC_GMM_C=2	55.65	54.36	55.25	55.87	55.28
FNGLVQ_MC_GMM_C=10	55.58	53.11	55.82	54.62	54.78
FNGLVQ_MC_IGMM_HIST_V1	55.56	53.11	54.99	54.79	54.61
FNGLVQ_MC_IGMM_HIST_V2	56.42	54.23	54.81	55.64	55.28
FNGLVQ_MC_IGMM_HIST_V3	55.40	53.40	54.65	55.50	54.74
GLVQ_ORIGINAL	41.97	44.70	25.93	47.64	40.06
GLVQ_MC_GMM_C=2	50.68	53.30	50.27	51.80	51.51
GLVQ_MC_GMM_C=10	56.04	52.92	54.94	54.45	54.59
GLVQ_MC_IGMM_HIST_V1	51.96	51.20	50.34	53.14	51.66
GLVQ_MC_IGMM_HIST_V2	50.88	51.47	49.69	52.47	51.13
GLVQ_MC_IGMM_HIST_V3	52.49	51.72	51.10	53.17	52.12

TABLE III
KAPPA MEASUREMENT ON SYNTHETIC A DATASET

Classifier\Dataset	Synt_A_1	Synt_A_2	Synt_A_3	Synt_A_4	Average
FNGLVQ_ORIGINAL	0.1610	0.3271	0.2384	0.3184	0.2612
FNGLVQ_MC_KMEANS_C=2	0.4260	0.4079	0.4248	0.4303	0.4222
FNGLVQ_MC_KMEANS_C=10	0.4279	0.3953	0.4198	0.4199	0.4157
FNGLVQ_MC_IKMEANS_ANO	0.4213	0.4236	0.4176	0.4406	0.4258
FNGLVQ_MC_IKMEANS_HIST	0.4568	0.4291	0.4274	0.4349	0.4370
FNGLVQ_MC_GMM_C=2	0.4456	0.4295	0.4406	0.4484	0.4410
FNGLVQ_MC_GMM_C=10	0.4448	0.4139	0.4478	0.4328	0.4348
FNGLVQ_MC_IGMM_HIST_V1	0.4445	0.4139	0.4374	0.4349	0.4327
FNGLVQ_MC_IGMM_HIST_V2	0.4553	0.4279	0.4351	0.4455	0.4409
FNGLVQ_MC_IGMM_HIST_V3	0.4425	0.4175	0.4331	0.4438	0.4342
GLVQ_ORIGINAL	0.2746	0.3088	0.0741	0.3455	0.2508
GLVQ_MC_GMM_C=2	0.3835	0.4163	0.3784	0.3975	0.3939
GLVQ_MC_GMM_C=10	0.4505	0.4115	0.4368	0.4306	0.4323
GLVQ_MC_IGMM_HIST_V1	0.3995	0.3900	0.3793	0.4143	0.3958
GLVQ_MC_IGMM_HIST_V2	0.3860	0.3934	0.3711	0.4059	0.3891
GLVQ_MC_IGMM_HIST_V3	0.4061	0.3965	0.3888	0.4146	0.4015

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

multi-codebook version that is generated by K-Means clustering, intelligent K-Means clustering, and Gaussian mixture model (GMM) clustering (parametric). The table shows that in the Synt_A dataset on average, the proposed multi-codebook FNGLVQ using intelligent GMM achieved the highest accuracy and Kappa. The highest performance is achieved by multi-codebook using intelligent GMM variant 2 with 55.28% and 0.0441 accuracy and kappa respectively. The value of accuracy and kappa is the same as with the multi-codebook FNGLVQ using GMM clustering with the number of clusters = 2.

The proposed multi-codebook FNGLVQ achieved better performance than the multi-codebook FNGLVQ using intelligent K-Means clustering that was proposed in the previous study. Looking at the table, we can also gain knowledge that the multi-codebook version using parametric GMM clustering gained better performance than using parametric K-Means clustering. We can confirm that the Gaussian mixture model clustering is more suitable than K-Means clustering in the dataset with a normal distribution. A different result is shown by multi-codebook GLVQ. The multi-codebook version of GLVQ achieved better performance than the original version. However, the proposed multi-codebook GLVQ using intelligent GMM clustering has a lower performance than multi-codebook GLVQ using parametric GMM clustering with the number of clusters = 2. But, the performance is still below the performance of the proposed multi-codebook FNGLVQ using intelligent GMM clustering. instances of the input data need to be resized into (32,32,1) dimension, except the Inception that needs (75,75,1) minimal size for its input. In general, we can confirm that the idea of multi-codebook architecture is more suitable for multi-modal data classification than the original model both for GLVQ and FNGLVQ base classifiers.

D. RESULT OF SCENARIO 2: EXPERIMENT ON SYNT_B DATASETS

The second scenario is evaluating the proposed methods in the synthetic B dataset (Synt_B). The evaluation of the proposed method in the Synt_B dataset is conducted to emphasize the first scenario if the idea of the multi-codebook model is more suitable for multi-modal data classification than the original version where the multi-modal data has a various number of classes and peaks. The Synt_B package has 8 instances. Each class in instances 1, 2,3, and 4 datasets has 2 peaks, while each class in instances 5,6,7, and 8 has 3 peaks. Instances 1 and 5 have the 2 class labels, instances 2 and 6 have 3 class labels, instances 3 and 7 have 4 class labels, while instances 4 and 8 have 4 class labels. All of the instances have 5 features and less inter-class overlapping than the Synt_A datasets.

Tables 4 and 5 show the recapitulation of accuracy and kappa measurements of the three variants of multi-codebook FNGLVQ using intelligent GMM clustering in Synt_B datasets. The tables also report the performance of multi-codebook GLVQ using intelligent GMM and previous multi-codebook versions.

Table 4 shows that the multi-codebook FNGLVQ using intelligent GMM achieved the highest accuracy in Synt_B datasets with 85.04% on average. Table 5 show shows that the multi-codebook FNGLVQ using intelligent GMM achieved the highest kappa in Synt_B datasets 0.7822 on average. Similar to the previous scenario, the highest performance was achieved by the multi-codebook FNGLVQ using intelligent GMM variant 2. The proposed multi-codebook version 2 also has better consistency compared to the multi-codebook FNGLVQ using parametric GMM. For example, the multi-codebook FNGLVQ using GMM with the number of clusters = 2 achieved good performance in instances 1,2,3, and 4, but bad performance in instances 5,6,7, and 8. Another example, the multi-codebook GMM with the number of clusters =10 achieves excellent performance in instance 5 but not a good performance in instance 1. Compared to the proposed multi-codebook variant 2, the other variants have comparable performance with 84.49% and 84.38% accuracy and 0.7745 and 0.7742 kappa for variant 1 and variant 3 respectively. Compared to the multi-codebook FNGLVQ using parametric GMM, the proposed method has higher performance with 0.23% and 0.006 accuracy and kappa respectively. Similar trends occurred for the multi-codebook GLVG using intelligent GMM clustering. Its variant 2 has the highest performance both from accuracy and kappa measurement with 77.59% and 0.6767 respectively. However, the performance is less than the multi-codebook GLVQ using parametric clustering with the number of clusters=10. It is the same trend as in the previous scenario. Compared to the original version, the multi-codebook FNGLVQ achieved higher accuracy with a 29% margin and kappa with a 0.34 margin. While the multi-codebook GLVQ using intelligent GMM clustering achieves better performance than its original version with 17% and 0.34 accuracy and kappa respectively. Compared to multi-codebook FNGLVQ using intelligent K-_Means clustering, the proposed method has better accuracy and kappa with 1% and 0.19 respectively. As for the multi-codebook AFNLGVQ using intelligent K-Means, its performance is lower than multi-codebook FNGLVQ using parametric K-Means. Overall, we can take some insights that in various multi-modal conditions the proposed multi-codebook FNGLVQ is still superior to its original version by a significant margin. Besides, the multi-codebook FNGLVQ using intelligent GMM clustering achieved better performance compared to multi-codebook FNGLVQ using intelligent K-Means clustering.

E. RESULT OF SCENARIO 3: EXPERIMENT ON BENCHMARK DATASETS

The third scenario is evaluating the proposed methods in the benchmark dataset. The evaluation of the proposed classifiers in the benchmark datasets is conducted to confirm if the idea of the multi-codebook model is suitable for real case problems that may have multi-modality in its feature compared to the original version.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

TABLE 4
ACCUARACY (%) MEASUREMENTS ON SYNTHETIC B DATASET

Classifier\Dataset	Synt_B_1	Synt_B_2	Synt_B_3	Synt_B_4	Synt_B_5	Synt_B_6	Synt_B_7	Synt_B_8	Average
FNGLVQ_ORIGINAL	70.10	52.35	53.46	46.17	62.62	54.30	53.54	48.17	55.09
FNGLVQ_MC_KMEANS_C=2	87.70	88.47	86.51	70.21	85.63	73.94	63.57	57.16	76.65
FNGLVQ_MC_KMEANS_C=10	85.62	86.58	86.46	75.59	91.71	88.67	85.77	77.71	84.76
FNGLVQ_MC_IKMEANS_ANO	85.88	87.30	86.46	73.02	87.21	83.59	78.76	71.93	81.77
FNGLVQ_MC_IKMEANS_HIST	86.80	87.85	85.32	71.63	91.50	89.29	84.48	74.43	83.91
FNGLVQ_MC_GMM_C=2	87.40	88.12	86.58	78.91	71.05	67.40	65.36	57.35	75.27
FNGLVQ_MC_GMM_C=10	85.75	86.33	86.06	76.60	91.75	88.54	85.92	77.56	84.81
FNGLVQ_MC_IGMM_HIST_V1	87.28	87.75	86.49	78.02	90.95	87.36	80.28	77.81	84.49
FNGLVQ_MC_IGMM_HIST_V2	87.50	88.02	86.66	77.63	90.72	89.12	82.44	78.23	85.04
FNGLVQ_MC_IGMM_HIST_V3	86.73	87.77	86.74	77.80	91.43	89.49	84.73	70.37	84.38
GLVQ_ORIGINAL	77.78	71.90	67.10	32.40	82.73	61.80	54.60	35.27	60.45
GLVQ_MC_GMM_C=2	81.90	78.18	74.29	65.41	77.78	69.79	57.68	45.23	68.78
GLVQ_MC_GMM_C=10	86.40	85.52	84.13	70.67	89.90	86.72	82.27	70.81	82.05
GLVQ_MC_IGMM_HIST_V1	83.85	81.72	78.83	65.13	87.28	77.29	74.17	64.91	76.65
GLVQ_MC_IGMM_HIST_V2	83.60	82.52	78.13	65.61	87.63	81.21	75.31	66.71	77.59
GLVQ_MC_IGMM_HIST_V3	83.55	81.32	76.49	63.77	87.12	81.34	75.35	59.31	76.03

TABLE 5
KAPPA MEASUREMENTS ON SYNTHETIC B DATASET

Classifier\Dataset	Synt_B_1	Synt_B_2	Synt_B_3	Synt_B_4	Synt_B_5	Synt_B_6	Synt_B_7	Synt_B_8	Average
FNGLVQ_ORIGINAL	0.4020	0.3028	0.3812	0.3266	0.2540	0.3158	0.3443	0.3471	0.3342
FNGLVQ_MC_KMEANS_C=2	0.7540	0.8270	0.8202	0.6276	0.7127	0.6092	0.5143	0.4646	0.6662
FNGLVQ_MC_KMEANS_C=10	0.7125	0.7988	0.8195	0.6949	0.8343	0.8302	0.8103	0.7214	0.7777
FNGLVQ_MC_IKMEANS_ANO	0.7315	0.8070	0.8157	0.6585	0.7500	0.7088	0.6949	0.6364	0.7254
FNGLVQ_MC_IKMEANS_HIST	0.7370	0.8150	0.8153	0.6249	0.8297	0.8408	0.7860	0.6594	0.7635
FNGLVQ_MC_GMM_C=2	0.7480	0.8218	0.8210	0.7364	0.4210	0.5110	0.5381	0.4669	0.6330
FNGLVQ_MC_GMM_C=10	0.7150	0.7950	0.8142	0.7075	0.8350	0.8282	0.8122	0.7195	0.7783
FNGLVQ_MC_IGMM_HIST_V1	0.7455	0.8163	0.8198	0.7253	0.8190	0.8103	0.7371	0.7226	0.7745
FNGLVQ_MC_IGMM_HIST_V2	0.7500	0.8203	0.8222	0.7204	0.8143	0.8368	0.7659	0.7279	0.7822
FNGLVQ_MC_IGMM_HIST_V3	0.7345	0.8165	0.8232	0.7225	0.8287	0.8423	0.7964	0.6296	0.7742
GLVQ_ORIGINAL	0.5555	0.5785	0.5613	0.1550	0.6547	0.4270	0.3947	0.1908	0.4397
GLVQ_MC_GMM_C=2	0.6380	0.6728	0.6572	0.5676	0.5557	0.5468	0.4358	0.3154	0.5487
GLVQ_MC_GMM_C=10	0.7280	0.7828	0.7883	0.6334	0.7980	0.8008	0.7636	0.6351	0.7412
GLVQ_MC_IGMM_HIST_V1	0.6770	0.7258	0.7177	0.5641	0.7457	0.6593	0.6556	0.5614	0.6633
GLVQ_MC_IGMM_HIST_V2	0.6720	0.7378	0.7083	0.5701	0.7527	0.7182	0.6708	0.5838	0.6767
GLVQ_MC_IGMM_HIST_V3	0.6710	0.7198	0.6865	0.5471	0.7423	0.7202	0.6713	0.4913	0.6562

Benchmark datasets were taken from the UCI machine learning repository and previous studies.

A benchmark dataset may have multi-modality on its feature but not all features have multi-modality, different from synthetic dataset where all features have multi-modal distribution. In this scenario, the proposed multi-codebook FNGLVQ was compared to the original FNGLVQ version, several multi-codebook versions as evaluated in the previous scenarios, and many well-known classifier models both from classical machine learning models: Naive Bayes, SVM, MLP, Bagging Tree, random forest and deep learning models: CNN, FCN, VGG, DenseNet, ResNet, Inception, AlexNet, and MobileNet. We hypothesized that the deep learning model is not usually superior to other models,

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

especially where the task (data) is not similar to the case where the deep learning models have been developed.

Tables 6 and 7 show the recapitulation of accuracy and kappa measurements of the three variants of multi-codebook FNGLVQ using intelligent GMM clustering in benchmark datasets. The tables also report the performance of multi-codebook GLVQ using intelligent GMM, previous multi-codebook versions, classical machine learning models, and deep learning models. Tables 6 and 7 show that the multi-codebook FNGLVQ using intelligent GMM achieved very good performance on average in benchmark datasets with 81.82% accuracy and 0.7165 kappa on average. The proposed method is only second best to multi-codebook GLVQ using intelligent GMM variant 1. However, the difference in performance is only slight i.e. 0.4 % accuracy and 0.0039 kappa. Different from the previous scenario, the highest performance of the proposed methods was achieved by the multi-codebook FNGLVQ using intelligent GMM variant 3. The other variants also performed well, as the difference of performance with variant 3 is less than 1%.

TABLE 6
ACCURACY (%) MEASUREMENTS ON BENCHMARK DATASET

Classifier\Dataset	Pinwheel	Glass	Ionosphere	Breast Cancer Coimbra	Wall Following	Segment	Ecoli	Odor	Average
FNGLVQ_ORIGINAL	92.24	59.25	88.60	64.67	78.42	80.86	85.40	75.45	78.11
FNGLVQ_MC_KMEANS_C=2	96.38	61.28	91.73	66.52	78.41	83.80	85.70	77.33	80.14
FNGLVQ_MC_KMEANS_C=10	77.02	53.17	88.90	65.36	82.12	81.52	86.86	75.17	76.27
FNGLVQ_MC_IKMEANS_ANO	94.41	55.72	93.43	63.96	73.15	85.58	87.20	77.58	78.88
FNGLVQ_MC_IKMEANS_HIST	91.08	59.85	91.15	65.40	80.92	86.70	86.00	78.25	79.92
FNGLVQ_MC_GMM_C=2	95.94	60.74	93.73	62.07	81.82	84.89	87.51	78.00	80.59
FNGLVQ_MC_GMM_C=10	81.60	52.29	88.61	62.86	82.28	72.25	86.33	75.92	75.27
FNGLVQ_MC_IGMM_HIST_V1	92.56	62.14	92.87	63.88	82.09	88.40	88.11	81.38	81.43
FNGLVQ_MC_IGMM_HIST_V2	92.68	62.19	92.02	66.34	79.14	88.01	87.51	79.21	80.89
FNGLVQ_MC_IGMM_HIST_V3	96.12	63.55	90.60	64.64	82.92	87.27	87.20	82.29	81.82
GLVQ_ORIGINAL	88.28	66.84	84.62	51.78	97.51	73.64	94.94	89.46	80.88
GLVQ_MC_GMM_C=2	98.32	66.39	87.18	48.26	95.78	74.07	93.74	86.71	81.31
GLVQ_MC_GMM_C=10	99.60	63.60	88.31	49.93	95.84	78.23	93.75	85.75	81.88
GLVQ_MC_IGMM_HIST_V1	99.26	61.63	89.73	54.54	97.07	77.06	92.84	85.46	82.20
GLVQ_MC_IGMM_HIST_V2	99.42	65.34	90.90	48.15	97.20	75.19	93.77	85.08	81.88
GLVQ_MC_IGMM_HIST_V3	99.26	69.15	90.60	44.89	96.44	72.81	94.67	85.67	81.69
Naive Bayes	88.84	46.10	89.43	62.61	90.63	79.78	68.06	72.74	74.77
SVM	99.72	64.39	92.00	55.65	93.57	59.29	85.97	86.79	79.67
Bagging Tree	76.18	58.66	89.43	68.70	93.99	55.61	85.08	35.85	70.44
Random Forest	88.92	61.48	90.00	66.09	93.99	73.15	87.76	48.69	76.26
MLP	98.96	33.81	90.00	48.70	84.44	38.65	82.99	73.49	68.88
CNN_2L	99.74	34.29	56.00	51.30	98.04	14.64	90.75	89.20	66.74
CNN_10L	99.66	33.29	64.00	51.30	97.56	97.58	90.75	86.87	77.63
FCN_2L	99.66	24.34	59.71	51.30	97.45	12.95	91.94	88.54	65.74
FCN_10L	31.66	24.91	56.57	48.70	47.99	14.46	47.76	12.17	35.53
FCN_10L02	35.68	29.58	55.43	48.70	40.42	13.21	43.88	7.84	34.34
AlexNet	94.42	12.71	61.43	93.91	39.87	13.47	77.91	97.79	61.44
VGG16	18.44	34.29	64.00	46.96	39.89	12.82	42.39	6.79	33.20
DenseNet121	80.42	31.44	59.71	93.04	41.65	14.64	64.18	96.12	60.15
ResNet50	82.89	26.79	59.14	60.87	39.76	14.16	40.30	72.54	49.56
InceptionV3	18.68	16.60	61.14	45.22	35.84	13.90	60.90	80.25	41.57
MobileNet	26.59	35.75	29.10	65.14	55.65	40.09	13.64	36.42	37.80

TABLE 7
KAPPA MEASUREMENTS ON BENCHMARK DATASET

Classifier\Dataset	Pinwheel	Glass	Ionosphere	Breast Cancer Coimbra	Wall Following	Segment	Ecoli	Odor	Average
FNGLVQ_ORIGINAL	0.9030	0.4240	0.7413	0.3125	0.6734	0.7768	0.7795	0.7323	0.6678
FNGLVQ_MC_KMEANS_C=2	0.9548	0.4645	0.8173	0.3466	0.6674	0.8111	0.7832	0.7527	0.6997
FNGLVQ_MC_KMEANS_C=10	0.7128	0.3491	0.7653	0.3168	0.7236	0.7843	0.8057	0.7291	0.6483
FNGLVQ_MC_IKMEANS_ANO	0.9513	0.4376	0.8319	0.2879	0.6033	0.8258	0.8280	0.7782	0.6930
FNGLVQ_MC_IKMEANS_HIST	0.9090	0.4846	0.7861	0.3109	0.6830	0.8556	0.7900	0.7668	0.6983
FNGLVQ_MC_GMM_C=2	0.9493	0.4435	0.8618	0.2502	0.7275	0.8237	0.8102	0.7600	0.7033
FNGLVQ_MC_GMM_C=10	0.7700	0.3580	0.7586	0.2500	0.7281	0.6763	0.7945	0.7373	0.6341
FNGLVQ_MC_IGMM_HIST_V1	0.9070	0.4662	0.8444	0.2760	0.7236	0.8647	0.8187	0.7968	0.7122
FNGLVQ_MC_IGMM_HIST_V2	0.9085	0.4698	0.8269	0.3351	0.6775	0.8601	0.8110	0.7732	0.7078
FNGLVQ_MC_IGMM_HIST_V3	0.9515	0.4867	0.7963	0.2959	0.7380	0.8515	0.8055	0.8068	0.7165
GLVQ_ORIGINAL	0.8535	0.5505	0.6434	0.0449	0.9626	0.6924	0.9244	0.8850	0.6946
GLVQ_MC_GMM_C=2	0.9790	0.5464	0.7040	-0.0107	0.9358	0.6975	0.9066	0.8550	0.7017
GLVQ_MC_GMM_C=10	0.9950	0.4929	0.7350	0.0307	0.9373	0.7460	0.9065	0.8445	0.7110
GLVQ_MC_IGMM_HIST_V1	0.9908	0.4789	0.7690	0.0985	0.9558	0.7323	0.8924	0.8414	0.7199
GLVQ_MC_IGMM_HIST_V2	0.9928	0.5305	0.7981	-0.0233	0.9577	0.7106	0.9065	0.8373	0.7138
GLVQ_MC_IGMM_HIST_V3	0.9908	0.5788	0.7893	-0.0668	0.9465	0.6828	0.9203	0.8436	0.7107
Naive Bayes	0.8604	0.3171	0.7553	0.2640	0.8596	0.7637	0.5441	0.7025	0.6333
SVM	0.9965	0.4958	0.8194	0.0000	0.9031	0.5278	0.7829	0.8558	0.6727
Bagging Tree	0.7021	0.3894	0.7683	0.3463	0.9063	0.4856	0.7675	0.3047	0.5838
Random Forest	0.8614	0.4290	0.7727	0.3099	0.9063	0.6879	0.8067	0.4433	0.6522
MLP	0.9870	0.0380	0.7701	0.0000	0.7621	0.2886	0.7382	0.7108	0.5369
CNN_2L	0.9968	0.0000	0.0484	0.0000	0.9703	0.0000	0.8573	0.8820	0.4694
CNN_10L	0.9958	0.0532	0.0000	0.0000	0.9633	0.9717	0.8577	0.8567	0.5873
FCN_2L	0.9958	0.0000	0.1066	0.0000	0.9617	0.0000	0.8741	0.8748	0.4766
FCN_10L	0.1476	0.0000	0.0000	0.0000	0.1958	0.0000	0.1506	0.0427	0.0671
FCN_10L02	0.1942	0.0000	0.0000	0.0000	0.0000	0.0000	0.2000	0.0000	0.0493
AlexNet	0.9302	0.0228	0.0467	0.8790	0.0983	-0.0075	0.6466	0.9759	0.4490
VGG16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DenseNet121	0.7565	0.0348	0.0695	0.8593	0.1008	0.0049	0.4603	0.9576	0.4055
ResNet50	0.7859	-0.0488	-0.0598	0.2507	0.0451	-0.0045	0.1030	0.7003	0.2215
InceptionV3	-0.0050	0.0000	0.1212	-0.0428	-0.0049	-0.0005	0.3880	0.7842	0.1550
MobileNet	0.0823	0.1962	0.0000	0.1082	0.0000	0.0038	-0.0065	-0.0198	0.0455

Compared to the proposed multi-codebook variant 3, the other variants have comparable performance with 81.43% and 80.89% accuracy and 0.7122 and 0.7078 kappa for variant 1 and variant 2 respectively. Compared to the multi-codebook FNGLVQ using parametric GMM, the proposed method has higher performance with 2% and 0.0132 accuracy and kappa respectively. A different trend also occurred for the multi-codebook GLVG using intelligent GMM clustering. Its variant 1 has the highest performance both from accuracy and kappa measurement with 82.20% and 0.7199 respectively. Besides, the performance is also higher than the multi-codebook GLVQ using parametric clustering both with the number of clusters=2 and the number of clusters=10. Compared to the original version, the multi-codebook FNGLVQ achieved higher accuracy with a 3.7% margin and kappa with a 0.0493 margin. While the multi-codebook GLVQ using intelligent GMM clustering achieves better performance than its original version with 1.4% and 0.027 accuracy and kappa respectively. Compared to multi-codebook FNGLVQ using

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

intelligent K-Means clustering, the proposed method has better accuracy and kappa with 1.9 % and 0.019 respectively. Different from previous scenarios, multi-codebook FNLGVQ using intelligent K-Means, performed better than multi-codebook FNGLVQ using parametric K-Means.

Some of the classical machine learning models performed well in the benchmark datasets as shown in tables 6 and 7. The model with the highest performance in the classical machine learning group is SVM with 79.67% accuracy and 0.6727 kappa, followed by random forest with 76.26% accuracy and 0.6522 kappa. Naive Bayes and bagging tree achieved adequate performance with 74.44% and 70.44% accuracy respectively, and 0.6333 and 0.6522 kappa respectively. While the MLP achieved the lowest performance among the classical machine learning model with 68.88% accuracy and 0.5368 kappa. The SVM model performed well in all benchmark datasets. The random forest model performed well in all datasets except the odor dataset. The Naive Bayes achieved bad performance in the glass dataset but achieved good performance in the other datasets. The bagging tree model achieved inadequate performance in the glass, segment, and odor dataset. While the model performed well in the other five datasets. The MLP model performed badly in the glass, breast cancer, and segment dataset but performed adequately in other datasets. The trend occurred in classical machine learning also occurred in the deep learning model where a model performed well in several datasets and performed badly in other datasets. The best performance among deep learning models is achieved by CNN10L with 77.63% accuracy and 0.5873 kappa. Despite its good performance overall, the model performed badly in the glass dataset and breast cancer dataset. The second-best performance among deep learning models is achieved by CNN_2L with 66.74 accuracy followed by FCN_2L with 65.74% accuracy.

TABLE 8
IMPROVEMENT OF THE MULTI-CODEBOOK FNGLVQ BASED ON ACCURACY METRICS

Dataset\Classifier	FNGLVQ_MC_KMEANS_C=2	FNGLVQ_MC_KMEANS_C=10	FNGLVQ_MC_IKMEANS_ANO	FNGLVQ_MC_IKMEANS_HIST	FNGLVQ_MC_GMM_C=2	FNGLVQ_MC_GMM_C=10	FNGLVQ_MC_IGMM_HIST_V1	FNGLVQ_MC_IGMM_HIST_V2	FNGLVQ_MC_IGMM_HIST_V3
Synt_A_1	21.20	21.35	21.12	23.66	22.77	22.70	22.68	23.54	22.52
Synt_A_2	6.43	5.45	7.83	8.16	8.19	6.94	6.94	8.06	7.23
Synt_A_3	14.91	14.51	14.33	15.12	16.18	16.75	15.92	15.74	15.58
Synt_A_4	8.95	8.12	10.20	9.32	10.40	9.15	9.32	10.17	10.03
Synt_B_1	17.60	15.52	15.78	16.70	17.30	15.65	17.18	17.40	16.63
Synt_B_2	36.12	34.23	34.95	35.50	35.77	33.98	35.40	35.67	35.42
Synt_B_3	33.05	33.00	33.00	31.86	33.12	32.60	33.03	33.20	33.28
Synt_B_4	24.04	29.42	26.85	25.46	32.74	30.43	31.85	31.46	31.63
Synt_B_5	23.01	29.09	24.59	28.88	8.43	29.13	28.33	28.10	28.81
Synt_B_6	19.64	34.37	29.29	34.99	13.10	34.24	33.06	34.82	35.19
Synt_B_7	10.03	32.23	25.22	30.94	11.82	32.38	26.74	28.90	31.19
Synt_B_8	8.99	29.54	23.76	26.26	9.18	29.39	29.64	30.06	22.20
Pinwheel	4.14	-15.22	2.17	-1.16	3.70	-10.64	0.32	0.44	3.88
Glass	2.03	-6.08	-3.53	0.60	1.49	-6.96	2.89	2.94	4.30
Ionosphere	3.13	0.30	4.83	2.55	5.13	0.01	4.27	3.42	2.00
Breast Cancer Coimbra	1.85	0.69	-0.71	0.73	-2.60	-1.81	-0.80	1.67	-0.04
Wall Following	-0.01	3.70	-5.27	2.50	3.40	3.86	3.67	0.72	4.50
Segment	2.94	0.66	4.72	5.84	4.03	-8.61	7.54	7.15	6.41
Ecoli	0.30	1.46	1.80	0.60	2.11	0.93	2.71	2.11	1.80
Odor	1.88	-0.28	2.13	2.80	2.55	0.47	5.93	3.76	6.84
Average Benchmark	2.03	-1.85	0.77	1.81	2.48	-2.84	3.32	2.78	3.71
Average Synthetic	18.66	23.90	22.24	23.90	18.25	24.45	24.17	24.76	24.14
Average All	12.01	13.60	13.65	15.07	11.94	13.53	15.83	15.97	15.97

TABLE 9
IMPROVEMENT OF THE MULTI-CODEBOOK FNGLVQ BASED ON KAPPA METRICS

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

Dataset\Classifier	FNGLVQ_MC_KMEANS_C=2	FNGLVQ_MC_KMEANS_C=10	FNGLVQ_MC_IKMEANS_ANO	FNGLVQ_MC_IKMEANS_HIST	FNGLVQ_MC_GMM_C=2	FNGLVQ_MC_GMM_C=10	FNGLVQ_MC_IGMM_HIST_V1	FNGLVQ_MC_IGMM_HIST_V2	FNGLVQ_MC_IGMM_HIST_V3
Synt_A_1	0.2650	0.2669	0.2603	0.2958	0.2846	0.2838	0.2835	0.2943	0.2815
Synt_A_2	0.0808	0.0681	0.0965	0.1020	0.1024	0.0867	0.0867	0.1008	0.0904
Synt_A_3	0.1864	0.1814	0.1793	0.1890	0.2022	0.2094	0.1990	0.1967	0.1947
Synt_A_4	0.1119	0.1015	0.1223	0.1165	0.1300	0.1144	0.1165	0.1271	0.1254
Synt_B_1	0.3520	0.3105	0.3295	0.3350	0.3460	0.3130	0.3435	0.3480	0.3325
Synt_B_2	0.5243	0.4960	0.5043	0.5123	0.5190	0.4923	0.5135	0.5175	0.5138
Synt_B_3	0.4390	0.4383	0.4345	0.4342	0.4398	0.4330	0.4387	0.4410	0.4420
Synt_B_4	0.3010	0.3683	0.3319	0.2983	0.4098	0.3809	0.3986	0.3937	0.3959
Synt_B_5	0.4587	0.5803	0.4960	0.5757	0.1670	0.5810	0.5650	0.5603	0.5747
Synt_B_6	0.2933	0.5143	0.3930	0.5250	0.1952	0.5123	0.4945	0.5210	0.5265
Synt_B_7	0.1700	0.4660	0.3506	0.4417	0.1938	0.4679	0.3928	0.4216	0.4521
Synt_B_8	0.1175	0.3743	0.2893	0.3123	0.1198	0.3724	0.3755	0.3808	0.2825
Pinwheel	0.0518	-0.1903	0.0483	0.0060	0.0463	-0.1330	0.0040	0.0055	0.0485
Glass	0.0405	-0.0749	0.0136	0.0606	0.0195	-0.0660	0.0422	0.0458	0.0626
Ionosphere	0.0760	0.0240	0.0906	0.0449	0.1205	0.0173	0.1032	0.0857	0.0550
Breast Cancer Coimbra	0.0341	0.0043	-0.0246	-0.0016	-0.0623	-0.0625	-0.0365	0.0226	-0.0167
Wall Following	-0.0060	0.0502	-0.0701	0.0096	0.0541	0.0547	0.0502	0.0041	0.0646
Segment	0.0343	0.0076	0.0490	0.0788	0.0470	-0.1005	0.0879	0.0833	0.0748
Ecoli	0.0037	0.0262	0.0485	0.0105	0.0307	0.0150	0.0392	0.0315	0.0260
Odor	0.0205	-0.0032	0.0459	0.0345	0.0277	0.0050	0.0645	0.0409	0.0745
Average Benchmark	0.0319	-0.0195	0.0251	0.0304	0.0354	-0.0338	0.0443	0.0399	0.0487
Average Synt_	0.2750	0.3472	0.3156	0.3448	0.2591	0.3539	0.3507	0.3586	0.3510
Average All	0.1777	0.2005	0.1994	0.2190	0.1697	0.1988	0.2281	0.2311	0.2301

However, measuring from kappa metrics, the performance of FCN_2L (0.4766 kappa) is better than CNN_2L performance (0.4694 kappa). Similar to CNN10L, these models also perform badly in the glass and breast cancer datasets. Besides, they also performed badly in the segment dataset. The FCN_10L and FCN_10L02 performed badly in all datasets which result in their very low performance on overall datasets.

The well-known deep learning architectures seem not suitable for this problem. It is shown by the tables that the best achievement is only 61.44% accuracy and 0.449 kappa. The models only performed well in a few tested benchmark datasets. AlexNet performed well only in pinwheel, odor, and breast cancer datasets, and performed badly in other datasets. Similar to ALexNet, the DenseNet also performed well only in those three datasets and performed badly in other datasets. VGG performed badly in all datasets shown by the kappa value=0.000 for all datasets, and so did the MobileNet. Inception performed well only in the odor dataset and performed badly in other datasets. ResNet performed well only in the pinwheel and odor dataset and performed badly in other datasets.

Overall, we can take some insights from this experiment. First, similar to the scenario in the synthetic dataset, in benchmark dataset that may contain multi-modality in its feature, the proposed multi-codebook FNGLVQ is still superior to its original version by a significant margin. Besides, the multi-codebook FNGLVQ using intelligent GMM clustering achieved better performance compared to multi-codebook FNGLVQ using intelligent K-Means clustering. In the benchmark datasets, the multi-codebook GLVQ using intelligent GMM clustering performed better than the original GLVQ version and multi-codebook version using parametric GMM clustering. Some classical machine learning models performed well, and some classical machine learning models performed adequately, except MLP which performed badly overall. The most suitable deep learning model is CNN_10L (10 layered convolutional neural networks). Many well-known deep learning models were not suitable for this problem since they were originally proposed for image classification with many classes and many instances.

F. *IMPROVEMENT OF MULTI-CODEBOOK NEURAL NETWORK*

After conducting the three experiment scenarios, we have analyzed the improvements of the proposed classifiers compared to the base classifiers for all datasets. The improvement values from original FNGLVQ to multi-codebook FNGLVQ is shown in table 8 and 9. Among the three variants of the proposed multi-codebook FNGLVQ using intelligent GMM clustering, variant 2 achieved the best improvement. The model achieved 24.76%, 2.78%, and 15.97% accuracy improvements in synthetic datasets, benchmark datasets, and overall datasets respectively. The model achieved 0.3586, 0.0399, and 0.2311 kappa improvements in the synthetic dataset, benchmark datasets, and overall datasets respectively. The proposed multi-codebook FNGLVQ variant 3 achieved a similar overall accuracy improvement of 15.97%. It achieved 24.14% and 3.71% accuracy improvement in the synthetic and benchmark datasets respectively. Variant 3 achieved better improvements in the benchmark dataset compared to variant 2. However, from kappa measurements, variant 3 achieved slightly lesser performance than variant 2 i.e. 0.3510, 0.0487, and 0.2301 in the synthetic datasets, benchmark datasets, and overall datasets respectively. The proposed method variant 1 achieved slightly lesser performance improvement than variants 2 and 3. It achieved 24.17%, 15.83%, and 15.83% accuracy improvements in the synthetic datasets, benchmark datasets, and overall datasets respectively. While from kappa measurements, variant 1 achieved 0.3507, 0.0443, and 0.02281 improvements for those three groups of datasets respectively. The proposed models had higher improvements for the synthetic, benchmark, and overall datasets compared to multi-codebook FNGLVQ using parametric GMM clustering. The multi-codebook using parametric clustering achieved 11.94% and 13.53% accuracy improvements in overall datasets for the number of cluster=2 and the number of cluster=10 respectively. The overall improvements are less than the improvements of the proposed methods by 2.4% accuracy margin. Besides, the number of cluster=10 achieved negative improvements on benchmark datasets. It means that the method is worse than the original FNGLVQ version in the benchmark dataset. The multi-codebook FNGLVQ using intelligent K-Means clustering based on histogram achieved comparable performance with the proposed method with a 15.07% accuracy improvement.

TABLE 10
IMPROVEMENT OF THE MULTI-CODEBOOK GLVQ BASED ON ACCURACY METRICS

Dataset\Classifier	GLVQ_MC_GMM_C=2	GLVQ_MC_GMM_C=10	GLVQ_MC_IGMM_HIST_V1	GLVQ_MC_IGMM_HIST_V2	GLVQ_MC_IGMM_HIST_V3
Synt_A_1	8.71	14.07	9.99	8.91	10.52
Synt_A_2	8.60	8.22	6.50	6.77	7.02
Synt_A_3	24.34	29.01	24.41	23.76	25.17
Synt_A_4	4.16	6.81	5.50	4.83	5.53
Synt_B_1	4.13	8.63	6.07	5.82	5.77
Synt_B_2	6.28	13.62	9.82	10.62	9.42
Synt_B_3	7.19	17.03	11.73	11.03	9.39
Synt_B_4	33.01	38.27	32.73	33.21	31.37
Synt_B_5	-4.95	7.17	4.55	4.90	4.38
Synt_B_6	7.99	24.92	15.49	19.41	19.54
Synt_B_7	3.08	27.67	19.57	20.71	20.75
Synt_B_8	9.97	35.54	29.65	31.44	24.04
Pinwheel	10.04	11.32	10.98	11.14	10.98
Glass	-0.45	-3.24	-5.22	-1.50	2.31
Ionosphere	2.56	3.69	5.11	6.28	5.98
Breast Cancer Coimbra	-3.52	-1.85	2.77	-3.63	-6.88
Wall Following	-1.72	-1.67	-0.44	-0.31	-1.06
Segment	0.43	4.59	3.42	1.56	-0.82
Ecoli	-1.20	-1.19	-2.10	-1.18	-0.28
Odor	-2.75	-3.71	-4.00	-4.38	-3.79
Average Benchmark	0.42	0.99	1.32	1.00	0.80
Average Synt_	9.38	19.25	14.67	15.12	14.41
Average All	5.79	11.94	9.33	9.47	8.97

TABLE 11
IMPROVEMENT OF THE MULTI-CODEBOOK GLVQ BASED ON KAPPA METRICS

Dataset\Classifier	GLVQ_MC_GMM_C=2	GLVQ_MC_GMM_C=10	GLVQ_MC_IGMM_HIST_V1	GLVQ_MC_IGMM_HIST_V2	GLVQ_MC_IGMM_HIST_V3
Synt_A_1	0.1089	0.1759	0.1249	0.1114	0.1315
Synt_A_2	0.1075	0.1028	0.0813	0.0846	0.0878
Synt_A_3	0.3043	0.3626	0.3051	0.2970	0.3146
Synt_A_4	0.0520	0.0851	0.0688	0.0604	0.0691
Synt_B_1	0.0825	0.1725	0.1215	0.1165	0.1155
Synt_B_2	0.0942	0.2043	0.1473	0.1593	0.1413
Synt_B_3	0.0958	0.2270	0.1563	0.1470	0.1252
Synt_B_4	0.4126	0.4784	0.4091	0.4151	0.3921
Synt_B_5	-0.0990	0.1433	0.0910	0.0980	0.0877
Synt_B_6	0.1198	0.3738	0.2323	0.2912	0.2932
Synt_B_7	0.0411	0.3689	0.2609	0.2761	0.2767
Synt_B_8	0.1246	0.4443	0.3706	0.3930	0.3005
Pinwheel	0.1255	0.1415	0.1373	0.1393	0.1373
Glass	-0.0041	-0.0576	-0.0716	-0.0200	0.0283
Ionosphere	0.0606	0.0916	0.1256	0.1547	0.1459
Breast Cancer Coimbra	-0.0557	-0.0142	0.0536	-0.0682	-0.1117
Wall Following	-0.0268	-0.0253	-0.0068	-0.0049	-0.0161
Segment	0.0051	0.0535	0.0399	0.0182	-0.0096
Ecoli	-0.0178	-0.0180	-0.0321	-0.0180	-0.0041
Odor	-0.0300	-0.0405	-0.0436	-0.0477	-0.0414
Average Benchmark	0.0071	0.0164	0.0253	0.0192	0.0161
Average Synt_	0.1204	0.2616	0.1974	0.2041	0.1946
Average All	0.0751	0.1635	0.1286	0.1301	0.1232

However, the model improvement in the benchmark datasets is very low i.e 1.81% accuracy improvement. Besides, if we compare the proposed multi-codebook FNGLVQ using intelligent GMM clustering to multi-codebook FNGLVQ using intelligent K-Means clustering, we can find that from 20 test datasets the proposed method achieved better performance in 12 test datasets and lower performance in 8 datasets. Therefore, both from margin improvements and head-to-head, the proposed method achieves better performance than the multi-codebook FNGLVQ using intelligent clustering. The other multi-codebook version using parametric K-Means also achieved lower performance improvements than the proposed method with a difference of more than 2.3% accuracy and 0.028 kappa.

After conducting the three experiment scenarios, we have analyzed the improvements of the proposed classifiers compared to the base classifiers for all datasets. The improvement values from original FNGLVQ to multi-codebook FNGLVQ is shown in table 8 and 9. Among the three variants of the proposed multi-codebook FNGLVQ using intelligent GMM clustering, variant 2 achieved the best improvement. The model achieved 24.76%, 2.78%, and 15.97% accuracy improvements in synthetic datasets, benchmark datasets, and overall datasets respectively. The model achieved 0.3586, 0.0399, and 0.2311 kappa improvements in the synthetic dataset, benchmark datasets, and overall datasets respectively. The proposed multi-codebook FNGLVQ variant 3 achieved a similar overall accuracy improvement of 15.97%. It achieved 24.14% and 3.71% accuracy improvement in the synthetic and benchmark datasets respectively. Variant 3 achieved better improvements in the benchmark dataset compared to variant 2. However, from kappa measurements, variant 3 achieved slightly lesser performance than variant 2 i.e. 0.3510, 0.0487, and 0.2301 in the synthetic datasets, benchmark datasets, and overall datasets respectively. The proposed method variant 1 achieved slightly lesser performance improvement than variants 2 and 3. It achieved 24.17%, 15.83%, and 15.83% accuracy improvements in the synthetic datasets, benchmark datasets, and overall datasets respectively. While from kappa measurements, variant 1 achieved 0.3507, 0.0443, and 0.02281 improvements for those three groups of datasets respectively. The proposed models had higher improvements for the synthetic, benchmark, and overall datasets compared to multi-codebook FNGLVQ using parametric GMM clustering.

The multi-codebook using parametric clustering achieved 11.94% and 13.53% accuracy improvements in overall datasets for the number of cluster=2 and the number of cluster=10 respectively. The overall improvements are less than the improvements of the proposed methods by

TABLE 12
SUMMARY OF SIGNIFICANCE TESTING OF MULTI-CODEBOOK NEURAL NETWORK BY USING ONE-WAY ANOVA

		Based on Accuracy Measurement			Based on Kappa Measurement		
Base Model	**Proposed Model**	**Fc**	**p-value**	**Significance**	**Fc**	**p-value**	**Significance**
FNGLVQ	FNGLVQ_MC_IGMM_HIST_V1	10.01961	0.003047	Yes	12.17209	0.001244	Yes
FNGLVQ	FNGLVQ_MC_IGMM_HIST_V2	10.40336	0.002588	Yes	13.01192	0.000888	Yes
FNGLVQ	FNGLVQ_MC_IGMM_HIST_V3	10.0957	0.00295	Yes	12.38248	0.001142	Yes
FNGLVQ	FNGLVQ_MC_IKMEANS_HIST	9.19133	0.004363	Yes	11.65812	0.001533	Yes
FNGLVQ	FNGLVQ_MC_IKMEANS_ANO	7.42375	0.009675	Yes	9.47961	0.003847	Yes
GLVQ	GLVQ_MC_IGMM_HIST_V1	2.34971	0.13359	No	3.901	0.055555	No
GLVQ	GLVQ_MC_IGMM_HIST_V2	2.35618	0.133072	No	2.46887	0.124412	Yes
GLVQ	GLVQ_MC_IGMM_HIST_V3	2.09901	0.155597	No	2.1826	0.147819	No

a 2.4% accuracy margin. Besides, the number of clusters=10 achieved negative improvements on benchmark datasets. It means that the method is worse than the original FNGLVQ version in the benchmark dataset. The multi-codebook FNGLVQ using intelligent K-Means clustering based on histogram achieved comparable performance with the proposed method with a 15.07% accuracy improvement. However, the model improvement in the benchmark datasets is very low i.e 1.81% accuracy improvement. The other multi-codebook version using parametric K-Means achieved lower performance improvements than the proposed method with a difference of more than 2.3% accuracy and 0.028 kappa.

In this analysis, we have analyzed the improvements of multi-codebook GLVQ using intelligent GMM clustering compared to the original GLVQ. The improvements data is displayed in tables 10 and 11 The three variants of the developed multi-codebook GLVQ achieved performance improvements around 9% accuracy and 0.2 kappa. However, the improvements are lower than the improvements of the multi-codebook GLVQ using parametric GMM clustering with the number of clusters=10. However, looking at improvements in the benchmark dataset, all multi-codebook achieved very low improvements compared to the original GLVQ. It means that the multi-codebook GLVQ is as good as the original GLVQ in the benchmark dataset. In other words, for GLVQ classifiers, applying the multi-codebook approach does not significantly improve the model performance in the benchmark datasets. On the contrary, the multi-codebook GLVQ achieved far better performance compared to the original GLVQ in the synthetic dataset i.e., more than 9% accuracy improvements and more than 0.007 kappa improvements. Recalling that the synthetic dataset has multi-modality in all features, while the benchmark dataset may have multi-modality in some features, we can take the insight that the multi-codebook GLVQ is more suitable in the data where all features have multi-modality than in the data that only has multi-modality in some / few features only.

G. SIGNIFICANCE TESTING FOR PROPOSED MULTI-CODEBOOK MODELS

In this study, we have analyzed the improvements of multi-codebook GLVQ using intelligent GMM clustering compared to the original GLVQ. The improvements data is displayed in tables 10 and 11 The three variants of the developed multi-codebook GLVQ achieved performance improvements around 9% accuracy and 0.2 kappa.

However, the improvements are lower than the improvements of the multi-codebook GLVQ using parametric GMM clustering with the number of clusters=10. Besides, looking at improvements in the benchmark dataset, all multi-codebook versions of GLVQ achieved very low improvements compared to the original GLVQ. It means that the multi-codebook GLVQ is as good as the original GLVQ in the benchmark dataset. In other words, for GLVQ classifiers, applying the multi-codebook approach does not significantly improve the model performance in the benchmark datasets. On the contrary, the multi-codebook GLVQ achieved far better performance compared to the original GLVQ in the synthetic dataset i.e., more than 9% accuracy improvements and more than 0.007 kappa improvements. Recalling that the synthetic dataset has multi-modality in all features, while the benchmark dataset may have multi-modality in some features, we can take the insight that the multi-codebook GLVQ is more suitable in the data where all features have multi-modality than in the data that only has multi-modality in some / few features only. In this sub-section, we evaluate the significance of the improvement of the proposed multi-codebook neural networks compared to its original performance. To determine whether the proposed method improved significantly, we utilize one-way analysis of variance (ANOVA) testing. ANOVA test is a method to compare whether two sets of samples are taken from the two populations that have the same mean value. The method compares two means from two populations by using F distribution [55]. The idea of the ANOVA test for significance testing is to analyze the performance of a base model and the proposed model. If the ANOVA test result concludes that the two sets of performance represent two populations that have the same mean, then we conclude that the two sets of performance are not significantly different. In other words, the performance of the proposed method is not significantly different than the performance of the base method. Otherwise, if the ANOVA test concludes that the two sets of performance represent two populations that have a different mean, then those two sets are significantly different. In other words, the performance of the proposed method is significantly different than the performance of the base method.

After conducting the experiments, we compute the significance testing of multi-codebook FNGLVQ and GLVQ by using a significant level = 0.05. The ANOVA test was conducted by evaluating the performance of the classifier models in 20 datasets. The ANOVA test for the multi-codebook models is summarized in table 12. The multi-codebook models are labeled as achieving significant improvement compared to the base model if the p-value is less than 0.05. Table 12 shows that the 3 variants of the proposed multi-codebook FNGLVQ using intelligent GMM clustering have significant improvements compared to the base (original) FNGLVQ. However, Table 12 shows that the 3 variants of multi-codebook GLVQ do not have significant improvements compared to GLVQ models. The multi-codebook FNGLVQ using intelligent K-Means clustering is also evaluated for significance testing. As shown in Table 12, both two models of multi-codebook FNGLVQ using intelligent K-Means clustering have significant improvements compared to the original FNGLVQ. Looking into the p-value and Fc, the proposed methods have higher significance than those two models as they have less p-value and higher Fc.

H. COMPUTATIONAL COMPLEXITY ANALYSIS

In this sub-section, we compared the complexity of the proposed multi-codebook neural network model and the base (single codebook) neural network model. Given a dataset consisting of n records, where each record has m features.4, the codebook update is conducted using equation 3 and 7-20. Since the number of operations is constant for a reference vector, then the complexity of the codebook update is notated as O(1). As explained in section 4, let's say that the models will be trained for t epochs. Let's note that the complexity is notated as C(). The complexity of the base model is written as follow:

$$\begin{aligned} C(\text{base model}) &= t \,.\, n \,.\, m \,.\, C(\text{codebook update}) \\ &= t \,.\, n \,.\, m \,.\, O(1) \\ &= O(mnt) \end{aligned}$$

In a case that number of features and epochs is defined as constant, then the complexity of the base model is notated as O(n). In the proposed multi-codebook model, there is a clustering process before the training process. Before the main clustering process, the method analyzes the histogram of the data to find the number of peaks which is becoming the number of mixtures (clusters) The complexity of GMM clustering can be approximated as $O(m^2 c^{3.5})$ according to the derivation in the previous study, where c is the number of the cluster [56]. While histogram analysis has O(n) complexity according to algorithm1. The complexity of the training process is the same as the training complexity of the base model, O(mnt). The complexity of the base model is written as follows:

$$\begin{aligned} C(\text{multicodebook}) &= C(\text{Clustering}) + C(\text{training}) \\ &= (O(n) + O(m^2 c^{3.5})) + O(mnt) \\ &= O(n) + O(m^2 c^{3.5}) + O(mnt) \end{aligned}$$

In our cases, $c^{3.5} <$ n and m < t, given the number of features and instances as shown in table 1, and the number of peaks as shown by the histogram plot in the appendix. Therefore the complexity of multi codebook is as follow:

$$\begin{aligned} C\,(\text{multicodebook}) &= O(n) + O(m^2 c^{3.5}) + O(mnt) \\ &< O(mnt) + O(m.\,m.\,c^{3.5}) + O(mnt) \\ &< O(mnt) + O(m.\,t.\,n) + O(mnt) \\ &< O(mnt) + O(mnt) + O(mnt) \\ &< 3.O(mnt) \\ &= O(mnt) \end{aligned}$$

This analysis shows that the multi-codebook model has the same complexity as the base model even though looking from the magnitude, the complexity of the multi-codebook model is higher than the base model. It implies that the training process of the multi-codebook model is longer than the base model. However, in the testing phase, the multi-codebook and the base model have the same mechanism, so their complexity is the same: n . O(1) = O(n).

I. DISCUSSION, RECOMMENDATION AND LIMITATION

The proposed multi-codebook FNGLVQ using intelligent GMM clustering has performed well in the 3 experiment scenarios. The 3 variants of the proposed multi-codebook FNGLVQ also showed consistency both in synthetic datasets and benchmark datasets. The best performance is achieved by variant 3 with 24.14%, 15.97%, and 3.71% improvements in synthetic datasets, benchmark datasets, and overall datasets respectively. The most consistent model among the proposed models is variant 2 wherein all datasets achieve better performance than the original version. The proposed model also achieved better performance compared with multi-codebook FNGLVQ using intelligent K-Means clustering that was proposed in the previous study. Compared head-to-head to multi-codebook based on intelligent K-Means clustering based on histogram information, the proposed method has better performance in 12 instances datasets from 20 total datasets, while in the 8 other datasets, the proposed method achieved lower performance even though with a very slight margin. In Synthetic datasets, where all features have multi-modality, the best performance is achieved by variant 2. Therefore, variant 2 is recommended to classify the dataset

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3213674

with multi-modality in all features. In the benchmark dataset where the data may or may not have multi-modality in its feature, variant 3 achieved better performance. Therefore, we recommend applying multi-codebook FNGLVQ for classification in the data where only a few features or no features that have multi-modal distribution

The current research is limited to classify multi-modal data in 1D. The currently proposed methods have been not evaluated yet in the case of 2D or higher dimensional data such as image, video, or point clouds. The other limitation of the proposed method is at the moment, the multi-codebook architecture was applied in single hidden layer neural networks that utilized the winner takes all approach. The proposed multi-codebook architecture and algorithm is not tested yet in complex architecture e.g. deep neural networks and convolutional neural networks.

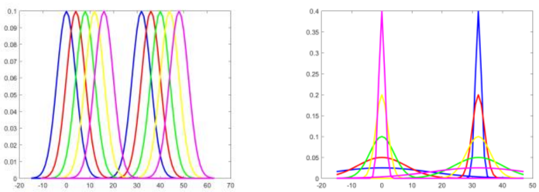
VII. CONCLUSION AND FUTURE WORKS

In this study, we have developed intelligent Gaussian mixture model clustering and multi-codebook FNGLVQ using intelligent GMM clustering. The proposed intelligent Gaussian mixture model clustering analyzes the gradient of the input data histogram to determine the number of clusters. We have developed three variants of multi-codebook FNGLVQ models. that performed well in the synthetic and benchmark datasets. From the theoretical analysis, the proposed method is expected to fit the data better and achieve better performance than the base method, while its computational complexity is the same as the base models'. From the experimental analysis, the proposed model improved the original FNGLVQ by 24.14%, 15.97%, and 3.71% accuracy and 0.3510, 0.0487, and 0.2031 kappa for synthetic datasets, benchmark datasets, and overall datasets respectively. By using the ANOVA test, we have proved that all three variants of the proposed multi-codebook FNGLVQ were proved to have significant improvements compared to the original version. The proposed method has several limitations. First, the proposed multi-codebook model was evaluated in 1-dimensional data. Second, the proposed method was applied in simple neural networks that have a single hidden layer. The drawback of the proposed method is that It requires a longer training time than the base method even though It has the same computational complexity. The other drawback is It has more parameters than the base model as It has multiple codebooks or reference vectors for each class.

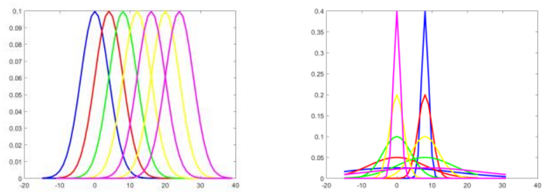
In the future, there are several ideas worth trying to investigate and enhance the proposed methods. One of the ideas is applying the multi-codebook approach for classification in higher-dimensional data such as 2D and 3D data. The second idea is combining clustering and other approaches such as incremental learning to enhance the classifiers' performance. The other idea is developing a multi-codebook architecture for deep learning.

APPENDIX

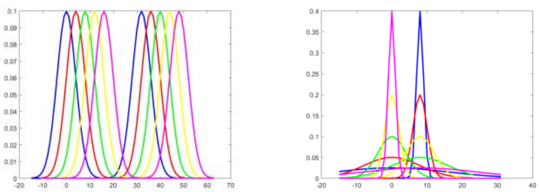
1. The distribution of SyntA dataset



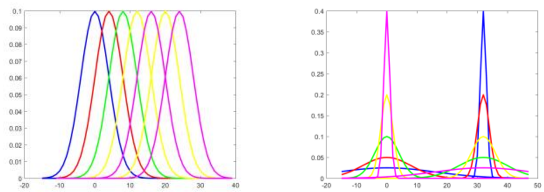
(instance 1)



(instance 2)

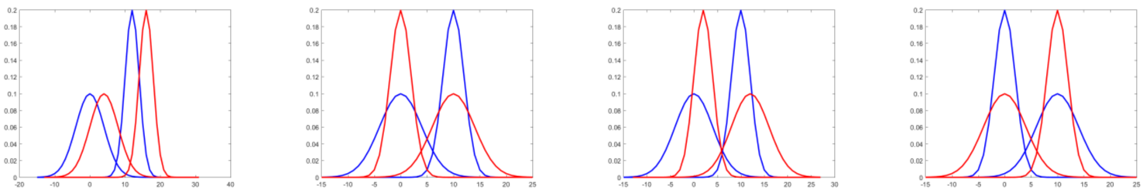


(instance 3)

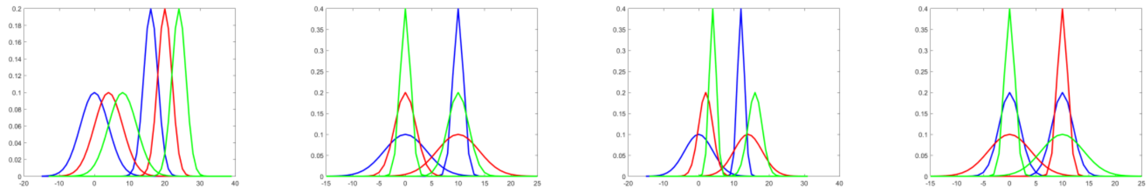


(instance 4)

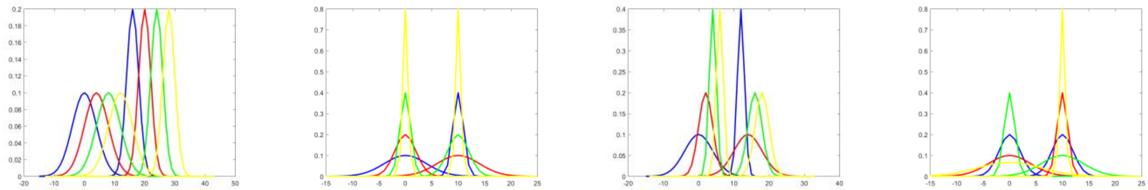
2. The distribution of SyntB dataset



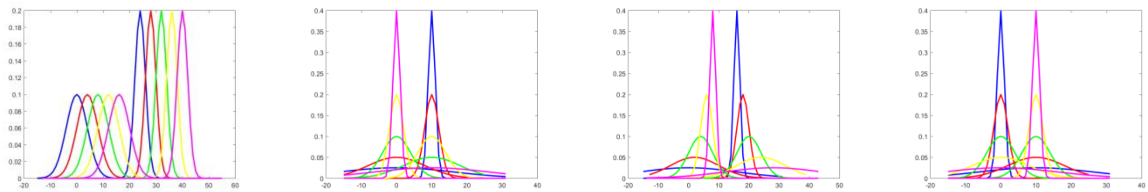
(2 class 2 peak)



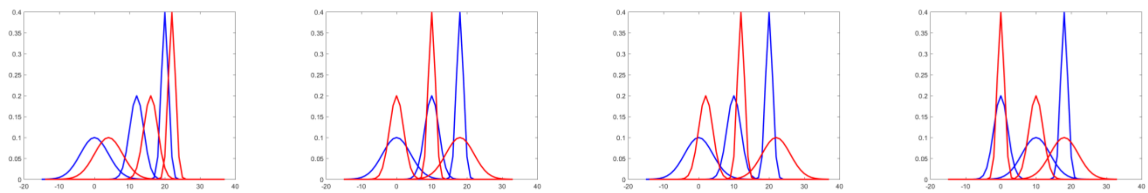
(3 class 2 peak)



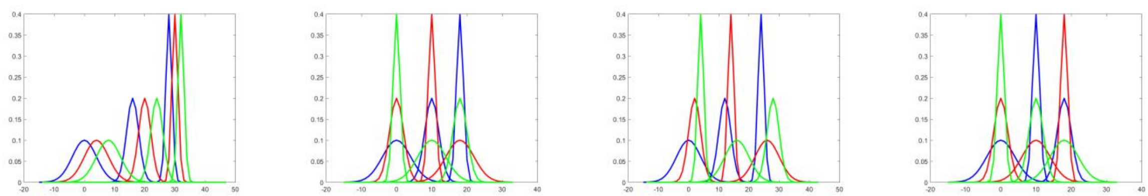
(4 class 2 peak)



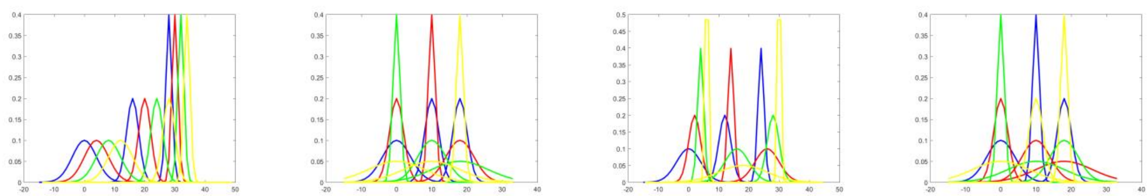
(5 class 2 peak)



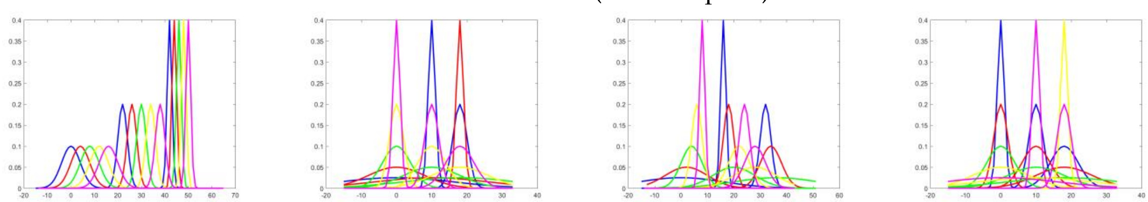
(2 class 3 peak)



(3 class 3 peak)

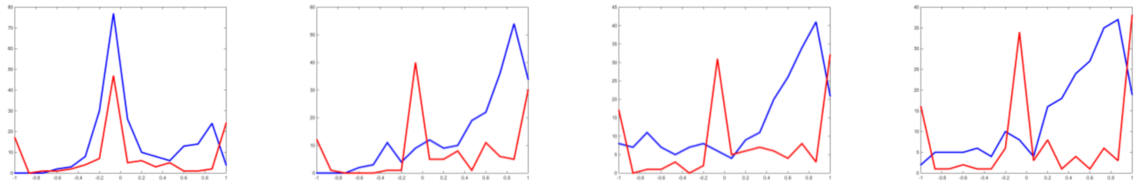


(4 class 3 peak)

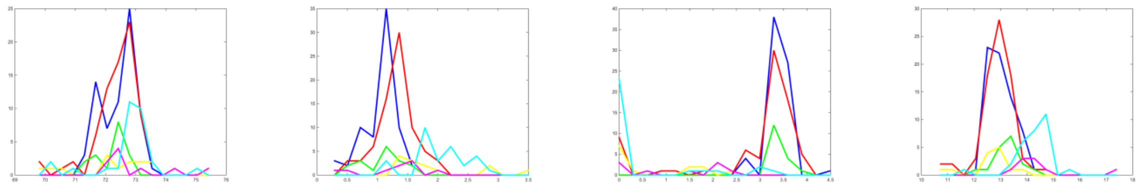


(5 class 3 peak)

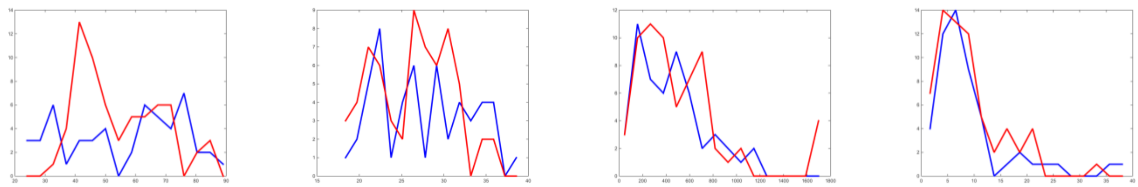
3. The distribution of benchmark dataset



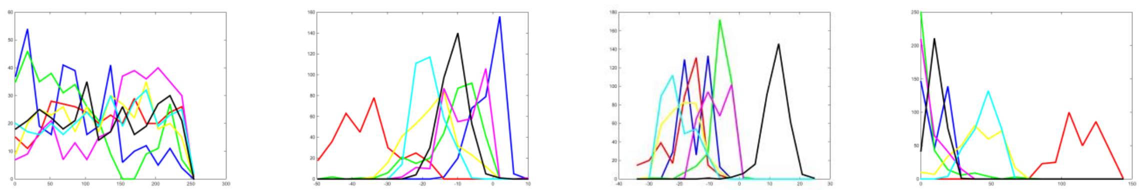
(Ionosphere)



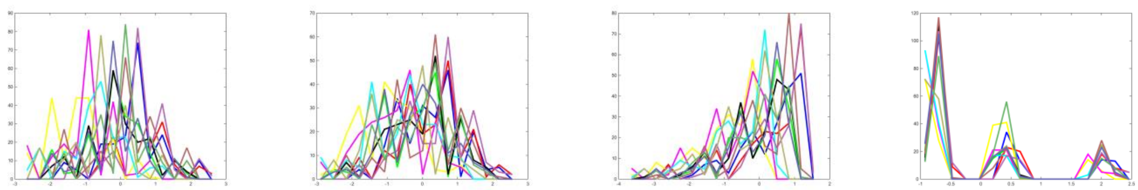
(Glass)



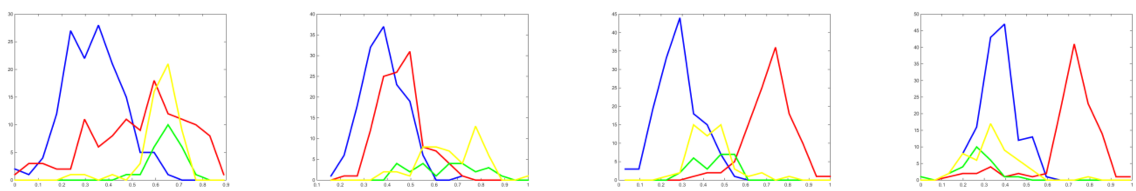
(Breast Cancer)



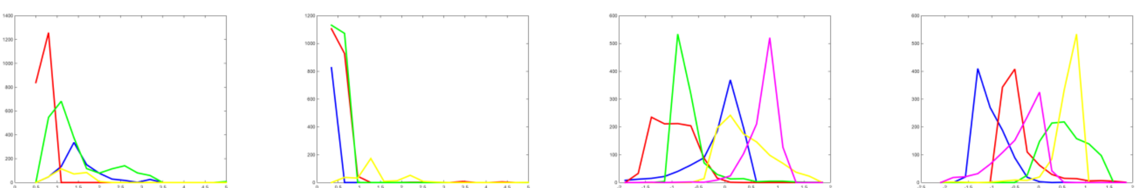
(Segment)



(Odor)



(Ecoli)



(Wall following) (Pinwheel)

ACKNOWLEDGMENT

This paper is supported by Universitas Indonesia PUTI Q1 Research Grant 2020-2021 Grant Number NKB-3885/UN2.RST/HKP.05.00/2020.

REFERENCES

[1] M. A. Ma'Sum *et al.*, "Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance," in *2013 international conference on advanced computer science and information systems (ICACSIS)*, 2013, pp. 161–166.
[2] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Rob. Auton. Syst.*, vol. 56, no. 11, pp. 915–926, 2008.
[3] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos, "Active policy learning for robot planning and exploration under uncertainty.," in *Robotics: Science and systems*, 2007, vol. 3, pp. 321–328.
[4] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, 2018.
[5] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero, "Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1548–1568, 2016.
[6] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic, "A survey of multimodal sentiment analysis," *Image Vis. Comput.*, vol. 65, pp. 3–14, 2017.
[7] A. Kumar, J. Kim, W. Cai, M. Fulham, and D. Feng, "Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data," *J. Digit. Imaging*, vol. 26, no. 6, pp. 1025–1039, 2013.
[8] P. Oskouie, S. Alipour, and A.-M. Eftekhari-Moghadam, "Multimodal feature extraction and fusion for semantic mining of soccer video: a survey," *Artif. Intell. Rev.*, vol. 42, no. 2, pp. 173–210, 2014.
[9] B. R. Abidi, N. R. Aragam, Y. Yao, and M. A. Abidi, "Survey and analysis of multimodal sensor planning and integration for wide area surveillance," *ACM Comput. Surv.*, vol. 41, no. 1, pp. 1–36, 2009.
[10] D. Kiela, E. Grave, A. Joulin, and T. Mikolov, "Efficient large-scale multi-modal classification," 2018.
[11] L.-M. Vortmann, M. Schult, M. Benedek, S. Walcher, and F. Putze, "Real-time multimodal classification of internal and external attention," in *Adjunct of the 2019 International Conference on Multimodal Interaction*, 2019, pp. 1–7.
[12] S. Poria, E. Cambria, A. Hussain, and G.-B. Huang, "Towards an intelligent framework for multimodal affective data analysis," *Neural Networks*, vol. 63, pp. 104–116, 2015.
[13] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimed. Syst.*, vol. 16, no. 6, pp. 345–379, 2010.
[14] M. A. Ma'Sum, H. R. Sanabila, W. Jatmiko, and others, "Multi codebook LVQ-based artificial neural network using clustering approach," in *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2015, pp. 263–268.
[15] M. A. Ma'sum, "Intelligent Clustering and Dynamic Incremental Learning to Generate Multi-Codebook Fuzzy Neural Network for Multi-Modal Data Classification," *Symmetry (Basel).*, vol. 12, no. 4, p. 679, 2020.
[16] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.
[17] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, 2017.
[18] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. R. Stat. Soc. Ser. c (applied Stat.*, vol. 28, no. 1, pp. 100–108, 1979.
[19] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, pp. 803–821, 1993.
[20] M. M.-T. Chiang and B. Mirkin, "Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads," *J. Classif.*, vol. 27, no. 1, pp. 3–40, 2010.
[21] M. A. Ma'sum, D. M. S. Arsa, I. Hermawan, W. Jatmiko, and A. Nurhadiyatna, "Multicodebook Neural Network Using Intelligent K-Means Clustering Based on Histogram Information for Multimodal Data Classification," in *2018 International Workshop on Big Data and Information Security (IWBIS)*, 2018, pp. 129–135.
[22] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst. Man. Cybern.*, vol. 21, no. 3, pp. 660–674, 1991.
[23] I. Rish and others, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, vol. 3, no. 22, pp. 41–46.
[24] K.-M. Chung, W.-C. Kao, C.-L. Sun, L.-L. Wang, and C.-J. Lin, "Radius margin bounds for support vector machines with the RBF kernel," *Neural Comput.*, vol. 15, no. 11, pp. 2643–2681, 2003.
[25] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 683–697, 1992.
[26] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.
[29] I. M. A. Setiawan, E. M. Imah, and W. Jatmiko, "Arrhytmia classification using fuzzy-neuro generalized learning vector quantization," in *2011 International Conference on Advanced Computer Science and Information Systems*, 2011, pp. 385–390.
[30] T. Kohonen, "Learning vector quantization," in *Self-organizing maps*, Springer, 1995, pp. 175–189.
[31] A. Sato and K. Yamada, "A formulation of learning vector quantization using a new misclassification measure," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, 1998, vol. 1, pp. 322–325.
[32] D. Zhang *et al.*, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *Neuroimage*, vol. 55, no. 3, pp. 856–867, 2011.
[33] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *Adv. Neural Inf. Process. Syst.*, pp. 409–415, 2001.
[34] J. F. Garc\'\ia Molina, L. Zheng, M. Sertdemir, D. J. Dinter, S. Schönberg, and M. Rädle, "Incremental learning with SVM for multimodal classification of prostatic adenocarcinoma," *PLoS One*, vol. 9, no. 4, p. e93600, 2014.
[35] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.
[36] M. A. Ma'sum, D. M. S. Arsa, N. Habibie, and W. Jatmiko, "Enhance generalized learning vector quantization using unsupervised extreme learning machine and intelligent k-means clustering," in *2017 International Workshop on Big Data and Information Security (IWBIS)*, 2017, pp. 77–83.
[37] E. Roszkowska and T. Wachowicz, "Application of fuzzy TOPSIS to scoring the negotiation offers in ill-structured negotiation problems," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 920–932, 2015.
[38] B. Kusumoputro, H. Budiarto, and W. Jatmiko, "Fuzzy-neuro LVQ and its comparison with fuzzy algorithm LVQ in artificial odor discrimination system," *ISA Trans.*, vol. 41, no. 4, pp. 395–407, 2002.
[39] W. Jatmiko, R. Rochmatullah, B. Kusumoputro, K. Sekiyama, and T. Fukuda, "Fuzzy learning vector quantization based on particle swarm optimization for artificial odor dicrimination system," *WSEAS Trans. Syst.*, vol. 8, no. 12, pp. 1239–1252, 2009.
[40] E. M. Imah, W. Jatmiko, and T. Basaruddin, "Adaptive Multilayer Generalized Learning Vector Quantization (AMGLVQ) as new algorithm with integrating feature extraction and classification for Arrhythmia heartbeats classification," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2012, pp. 150–155.
[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.
[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[44] B. Zoph, V. Vasudevan, J. Shlens, and Q. V Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
[45] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv Prepr. arXiv1704.04861*, 2017.
[46] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proceedings of The Web Conference 2020*, 2020, pp. 1400–1410.
[47] K.-L. Lim, X. Jiang, and C. Yi, "Deep clustering with variational autoencoder," *IEEE Signal Process. Lett.*, vol. 27, pp. 231–235, 2020.
[48] X. Yang, C. Deng, K. Wei, J. Yan, and W. Liu, "Adversarial learning for robust deep clustering," *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.
[49] Y. Ni, P. Müller, M. Diesendruck, S. Williamson, Y. Zhu, and Y. Ji, "Scalable Bayesian nonparametric clustering and classification," *J. Comput. Graph. Stat.*, vol. 29, no. 1, pp. 53–65, 2020.
[50] M. d'Errico, E. Facco, A. Laio, and A. Rodriguez, "Automatic topography of high-dimensional data sets by non-parametric Density Peak clustering," *Inf. Sci. (Ny).*, vol. 560, pp. 476–492, 2021.
[51] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: Incomplete multi-view clustering via contrastive prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11174–11183.
[52] Z. Huang, P. Hu, J. T. Zhou, J. Lv, and X. Peng, "Partially view-aligned clustering," *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.
[53] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *International conference on machine learning*, 2019, pp. 5092–5101.
[54] T. Glarner, P. Hanebrink, J. Ebbers, and R. Haeb-Umbach, "Full Bayesian Hidden Markov Model Variational Autoencoder for Acoustic Unit Discovery.," in *Interspeech*, 2018, pp. 2688–2692.
[55] D. C. Howell, *Statistical methods for psychology*. Cengage Learning, 2012.
[56] I. Kerenidis, A. Luongo, and A. Prakash, "Quantum expectation-maximization for Gaussian mixture models," in *International Conference on Machine Learning*, 2020, pp. 5187–5197.

M. Anwar Ma'sum (M'17) born in Nganjuk, East Java, Indonesia in 1990. He received Bachelor's and Master's Degree in Computer Science from Universitas Indonesia in 2013 and 2014 respectively.

He is currently working as a lecturer and researcher in the Faculty of Computer Science Universitas Indonesia. From 2010 to 2014 he worked as a research assistant in the laboratory of Computer Networks, Architecture and High-Performance Computing (CNAHPC), Faculty Of Computer Science Universitas Indonesia. His research interests are in Intelligent System and Machine Learning. He is currently working on a research project to develop learning methods for multi-modal data analysis. Prior to this study, he worked in neural networks, ECG data analysis, USG data analysis, UAV for surveillance and other intelligent system applications.

He became EEE member in 2017. He has actively contributed to conferences supported by IEEE held in Indonesia i.e. ICACSIS (2011-2018), and IWBIS (2016-2018) as the organizing committee. He also actively participated in IEEE Indonesia section events i.e. workshop, gathering, and meeting.

Noverina Alfiany was born in Kuala Simpang, East Aceh, Indonesia in 1986.She received a Bachelor's Degree in 2010 and a Master's Degree in 2013, both in Mathematics from Andalas University. From 2015 until 2018, she pursues her Doctoral course at Okayama University, Japan.

Since February 2021, she is currently working as a lecturer at Andalas University. From 2019 to the beginning of 2021, she works as a lecturer and researcher in the Faculty of Computer Science Universitas Indonesia. Her research interests are in Numerical Analysis, Tsunami Simulation, and Robotics Mechanical System. She is currently interested in the topics of Mathematics for Machine Learning. Her last work was a numerical simulation of the tsunami wave; and (with the team) ECG data analysis, USG data analysis, and Mathematics application on UAV and robotics system..

WISNU JATMIKO (Senior Member, IEEE) received the B.S. degree in electrical engineering and the M.Sc. degree in computer science from Universitas Indonesia, in 1997 and 2000, respectively, and the Dr.Eng. degree from Nagoya University, Japan, in 2007. He is currently a Full Professor with the Faculty of Computer Science, Universitas Indonesia. His research interests are autonomous robot, optimization, and real-time traffic monitoring systems.

.