CAMES, **29**(1–2): 105–123, 2022, doi: 10.24423/cames.409

This article belongs to the *Special Issue on Advanced Optimization Methods for Uncertainties in Intelligent Industrial Systems* edited by Dr. I.J. Jacob, Dr. B.H. bin Ahmad and Dr. Z.F. Khan

An Intelligent Neural Network Algorithm for Uncertainty Handling in Sensor Failure Scenario of Food Quality Assurance Model

S.N. DEEPA[1)*], N. Yogambal JAYALAKSHMI[2)]

[1)] *Department of Electrical Engineering*
National Institute of Technology Arunachal Pradesh
Jote, Arunachal Pradesh, India
*Corresponding Author e-mail: sndeepa@nitap.ac.in, deepapsg@gmail.com

[2)] *Department of Electrical and Electronics Engineering*
Dr. Mahalingham College of Engineering and Technology
Pollachi, Coimbatore, Tamil Nadu, India
e-mail: yogambaljayalakshmi@gmail.com

The quality of food is usually tested by sensing the product odor using e-nose technique. However, in a real-time testing environment, some of the employed sensors may fail to operate, which imposes great uncertainty on the food quality assurance model. To handle the uncertainty, a support vector machine (SVM) classifier algorithm is developed to deal with the failure sensor effect using a data imputation strategy. The proposed model is evaluated experimentally by means of benchmark datasets, and validated in a real-time environment by programming an Arduino-UNO controller in the internet of things (IoT) environment.

Keywords: e-nose, data imputation, quality assurance, multiclass SVM, k-nearest neighbor, IoT, Arduino UNO.

1. Introduction

In these modern days, food processing units are increasing, so food quality assurance is becoming the essential part of this busy world. The quality of food determines why it is consumed and the impact of consuming it. Food and food processing units include fruits, vegetables, milk, ghee, chicken, fish, beef, beverages such as fruit juice, beer, wine, etc. Classifying these foods based on their quality became mandatory to keep spoiled food from fresh one. The proposed model focuses on quality assurance of the product based on the odor produced by it based on the quality with the help of an electronic nose. The e-nose is the array of gas identifying sensors used to find the different aromas of the product under study. In various stages of the food fermentation process various gases are produced, and the nature of gas produced varies for different food products. By analyzing the nature of gas produced by the substrate, its quality can be predicted. For instance, a certain aroma for a specific product indicates the exact quality of that food, some indicate the ripening stage, and some gas is produced at a completely spoiled stage. By considering all these factors carefully, the substrate is classified. Dairy products such as milk can produce a combination of hydrogen and carbon (IV) oxide at the stage of fermentation.

When it comes to fruits and vegetables, the ethylene gas will initiate the ripening process, and identifying this stage will help to reduce spoilage. Organic material spoiled by water emits methane gas that may cause pollution to the environment. When it comes to meat quality. the characteristic indicators include texture, odor, flavor, water content level, etc. The odor of fish includes trimethylamine, ammonia, hydrogen sulfide (H_2S), etc. Processed or packaged meat affected by H_2O_2 (hydrogen peroxide) or H_2S (hydrogen sulfide) shows as fungal-like greening surface on the meat. Beverages such as wine, beer and processed fruit juice smell like carbon dioxide (CO_2) or ethanol by their nature. On the other hand, nitrogen (N_2), carbon dioxide (CO_2) and oxygen (O_2) gases are used for preservation and packing processes to provide a fresh taste and aroma [18]. All these gases are sensed and identified by the electronic nose equipped with various sensors such as MQ2, MQ3, MQ4, MQ5, MQ6, MQ8, MQ9, MQ135 and MQ136.

2. Survey of existing models in the literature

Sanaeifar *et al.* [1] proposed a computational model to study the ripening process of bananas with the help of an e-nose. Different algorithms employed to find the various ripening stages are principal component analysis (PCA), linear discriminant analysis (LDA), soft independent modelling of class analogy (SIMCA) and support vector machines (SVM). The SVM provided high accuracy results for finding bananas' maturity. In addition to ripening, the odor produced during storage life was calculated with the help of e-nose and an artificial neural network. The e-nose was equipped with metal oxide semi-conductor (MOS) for the investigation of various types of aroma in the study of Sanaeifar *et al.* [2]. Different approaches such as back-propagation multilayer perceptron (BP-MLP) neural network for pattern recognition and SVM for classifying different shelf-life stages were used. The enhanced model of this system predicted the quality of bananas with some more features such as total soluble solids (TSS), titratable acidity (TA), pH and stability. The model in Sanaeifar *et al.* [3] used partial least squares (PLS), multiple linear regression (MLR) and support vector regression (SVR) techniques to predict the quality.

Table 1. Survey of e-nose development with expert system.

Authors and reference	Material used	Functionality	Methods used	Year
Ordukaya and Karlik [6]	Olive oil	Quality control	NB, NN, LDA, DT, ANN, SVM	2017
Wijaya *et al.* [7]	Beef	Quality monitor	Machine learning methods	2018
Thorson *et al.* [8]	Chamber environment	Identifying pollutant mixers and sources	LR, RF, SVM, NN	2019
Han *et al.* [9]	MOX dataset	Identifying mixed gas	CNN	2019
Rodriguez Gamboa *et al.* [10]	Wine	Identifying beverage quality	Machine learning methods	2019
Kalpana and Baghyam [11]	Fruits	Classification of fruits and measuring freshness	KNN	2019
Ghasemi-Varnamkhasti *et al.* [12]	Strawberry	Identifying freshness in polymer packages	SVM, LDA, PCA	2019
Voss *et al.* [13]	Beer	Identifying alcohol content	ELM	2019
Hasan *et al.* [14]	Pineapple	Sweetness classification using aroma	PCA, KNN	2020
Rahman *et al.* [15]	Industrial area	Identifying volatile organic gases	PCA, MA	2020
Nouri *et al.* [16]	Pomegranate	Quality detection by identifying a fungal infection	BPNN, LDA, SVM	2020
Aghilinategh *et al.* [17]	Berries	Ripeness detection	ANN, LDA, PCA	2020

Abbreviations: NB – naive Bayesian, NN – nearest neighbor, LDA – linear discriminate analysis, DT – decision tree, ANN – artificial neural networks, SVM – support vector machine, LR – logistic regression, RF – random forests, NN – neural networks, CNN – convolutional neural networks, KNN – k-nearest neighbor, PCA – principal component analysis, ELM – extreme learning machines, MA – multivariate analysis, BPNN – back propagation neural network.

The aroma classification of four different fruits was proposed by Adak and Yumusak [4]. Their model used an artificial bee colony (ABC) algorithm and back propagation to train the classification model. Different aroma datasets for this model were collected using MOSES or e-nose and the best results were obtained with the ABC algorithm. Besides bananas and fruits, different types of saffron were identified with the help of e-nose coupled with the expert model proposed by Kiani *et al.* [5]. The aroma of eleven types of saffron from different regions were investigated by e-nose and the classification model was trained using supervised learning methods such as partial least squares (PLS) and multilayer perceptron (MLP). Qualities such as excellent, good, bad, etc. were studied by unsupervised methods known as principal component analysis (PCA) and hierarchical clustering analysis (HCA). MLP provided better result with the RMSE value of 0.0141.

Researchers in expert system propose several e-nose-based models for various foods' quality monitoring and ripeness predicting. The evolution of expert systems with e-nose is shown in Table 1 from 2017 to 2020. All the researchers from [6] to [17] in Table 1 used e-nose to obtain the various values of gas produced from foods. The extreme learning machine employed for classification was discussed in [18], and the performance of deep learning was analyzed in [19].

3. Proposed methodology

Artificial intelligence olfaction-based food quality monitoring model is developed in this paper. The conceptual framework is presented in Fig. 1. The sensor array collects the data from food substrate in real-time, and the collected data

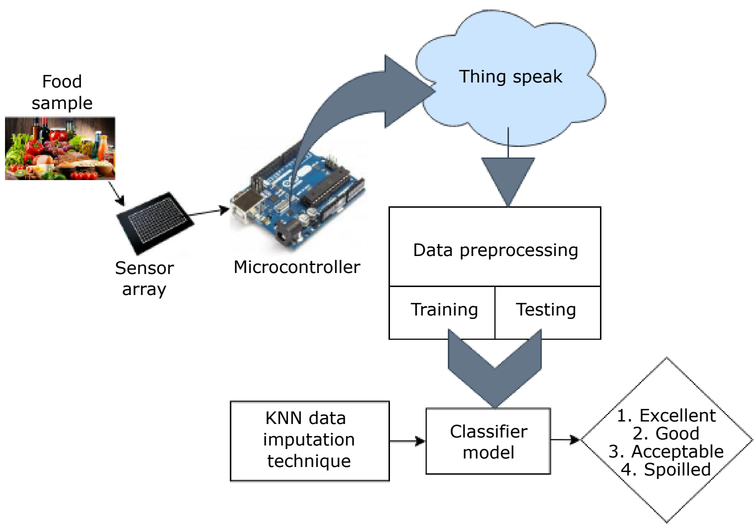

Fig. 1. Block diagrammatic representation of the proposed model.

is stored in the cloud by means of a programmed microcontroller. However, the operation of sensors can be affected by environmental factors, which may result in sensor drift. An ensemble machine learning model for sensor failure tolerable model was proposed by Kaya *et al.* [20].

In this study, an ensemble-based classifier model is proposed. The model is made to train for each sensor loss by employing five-fold cross-validation. In addition, in our proposed methodology a KNN-based data imputation technique is incorporated with the multiclass SVM classifier model, and the model is trained by ten-fold cross-validation considering the loss of data acquired from the sensor array. The performance of the classifier model for food quality assurance is initially investigated with benchmark datasets considering the sensor failure, and validated with hardware design in the real-time scenario.

The proposed quality assurance model is designed in such a way that it can perform efficiently in sensor failure situations. The data acquired from the sensor array is in unreadable format or empty during the sensor miss operation and failure scenarios. In the case of sensor failure or miss operation, the data field of the particular sensor will be empty or undefined, so the KNN will take care of those fields by substituting them with average feature values. The training is conducted by employing ten-fold cross-validation, the entire training data is segregated into ten folds, while nine-fold data is employed for training the remaining 1-fold employed for testing. The training is conducted with no sensor loss, while the testing fold has 10% of sensor loss, i.e., one sensor is considered as failure, another sensor is assumed to be a failure for the next fold, the procedure continues for all ten folds, and at the end of the cross-validation process all sensor failure situations will be fed into the model. Now, the model is completely trained for all sensor failure situations.

4. Experimental modeling of the proposed methodology

The experimental modeling of the proposed model has three main modules, the data acquisition module, classification and communication module. The description of each module is presented as follows.

4.1. Data acquisition

The collected food substrate is taken in an airtight closed chamber, the generated gas in the closed chamber goes into the sensing chamber where the sensor array is installed. In the proposed study. nine gas sensors are employed along with temperature and humidity measurement, so a total of 11 field measurements is done and stored in the cloud, the employed sensors and their corresponding sensing gases are given in Table 2, the collected data is transmitted into the created

Table 2. Sensor and the corresponding sensing gas.

Sensor	Sensing gas
MQ-2	Methane, Butane, LPG, Smoke
MQ-3	Alcohol, Ethanol, Smoke
MQ-4	Methane, CNG gas
MQ-5	Natural gas, LPG
MQ-6	LPG, Butane
MQ-8	Hydrogen gas
MQ-9	Carbon monoxide, Flammable gasses
MQ135	Air quality
MQ136	Hydrogen sulfide gas

new channel in ThingSpeak cloud by Arduino UNO controller board through the ESP8266 wifi module. The ThinkSpeak is an open source cloud storage framework, and the data can be stored and retrieved through the programming of ThingHTTP by the Arduino controller.

4.2. Communication module

The communication is established by means of the ESP8266 wifi module, the collected data is stored in an open-source ThingSpeak cloud framework, and the ThingSpeak application is interfaced with the server through the WriteAPI key and Arduino UNO programming. The given data substrate is analyzed by the classifier model and the quality of food is expressed by four classes such as 'excellent', 'good', 'acceptable', and 'spoiled'. The decision made in the remote server is transmitted to the analyzer through the wifi module. Further, to send message notification to the user, another application is employed named React, and the quality of food is mailed to the user through the IFTTT. The IFTTT is an interface application that triggers message based on the if-then rule.

4.3. Classifier module

The developed food quality assurance model based on a multiclass SVM classifier algorithm with KNN-based data imputation strategy is programmed in MATLAB environment. The data stored in the cloud is exported to a CSV file, and the entire dataset is segregated into training and testing datasets. The training is done with ten-fold cross-validation considering the sensor failure situation for each with the data imputation technique.

Multiclass SVM classifier. The proposed model flags the food sample into four different classes of qualities as 'excellent', 'good', 'acceptable' and 'spoiled', so a multiclass SVM classifier algorithm is considered in this study. The SVM classifier algorithm was proposed initially to handle the binary classification problems, Cortes and Vapnik [21]; to separate the entire samples into two classes, a hyperplane is constructed in a high dimensional space over the available training samples. The multiclass SVM classifier combines multiple binary classes by one vs. one, one vs. all or direct acrylic graph construction. The basic illustration of binary classifier algorithm and multiclass SVM classifier is presented in Fig. 2.

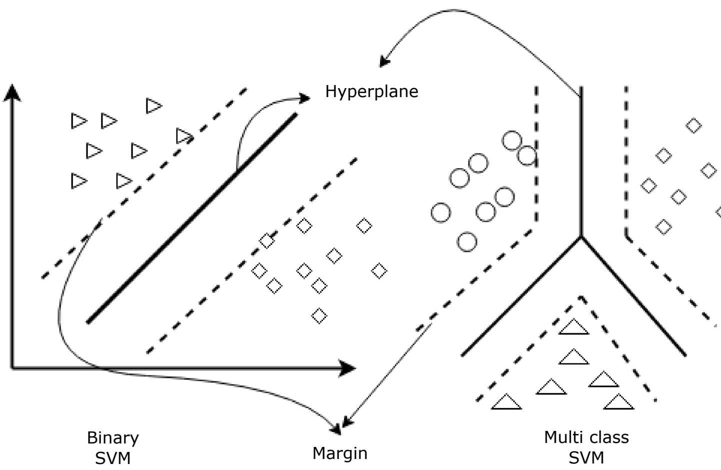

FIG. 2. Illustration of binary and multiclass SVM.

For the binary separable problem, for the given input sample $\{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i)\}_{i=1}^{N}$, where $x_i (x_i \in R^t)$ represents the input set of i training samples, $x_i = \left[x_i^1, x_i^2, ..., x_i^t\right]^T$ and $y_i \in (-1, +1)$ represents the output class, the hyperplane is constructed by the equation $w^T \cdot x + b = 0$.

The parallel boundaries of the hyperplanes are given by:

$$w^T \cdot x + b = +1 \qquad \text{and} \qquad w^T \cdot x + b = -1. \tag{1}$$

The hyperplane is constructed such that $\frac{1}{2} \|w\|^2$ satisfies the constraints presented as follows:

$$y_i(w^t \cdot x_i + b) \geq 1 \qquad \forall i = 1, 2, ..., N. \tag{2}$$

On introducing Lagrange multipliers:

$$L(w, b, \alpha) = \frac{1}{2} \left\|w^2\right\| - \sum_{i=1}^{N} \alpha_i [y_i(w^t \cdot x_i + b) - 1]. \tag{3}$$

The expression (3) should be minimized with respect to w and b, and subject to the following constraints $\alpha_i \geq 0$, $\forall i = 1, 2, ..., N$,

$$\left.\begin{aligned} w &= \sum_{i=1}^{N} y_i \alpha_i x_i \\ \sum_{i=1}^{N} y_i \alpha_i &= 0 \end{aligned}\right\}. \tag{4}$$

Now, the dual optimization problem is given by

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_j^i x_j - \sum_{i=1}^{N} \alpha_i. \tag{5}$$

Subject to the constraints

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad \alpha_i \geq 0 \quad \forall i = 1, 2, ..., N. \tag{6}$$

For optimal hyperplane construction, a slack variable ξ_i is introduced in Eq. (2),

$$y_i(w^t \cdot x_i + b) \geq (1 - \xi_i) \qquad \text{and} \qquad \xi_i \geq 0 \quad \forall i = 1, 2, ..., N. \tag{7}$$

The optimal separating plane is given by

$$\min_{(w,\xi)} L(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \tag{8}$$

and subject to

$$\left.\begin{array}{ll} y_i(w^t \cdot x_t + b) \geq 1 - \xi_i & \text{for} \quad 1 \leq i \leq N \\ \xi_i \geq 0 & \text{for} \quad 1 \leq i \leq N \end{array}\right\}. \tag{9}$$

Next, Lagrange multipliers are introduced:

$$L(w, b, \alpha, \xi, \beta) = \frac{1}{2} \left\|w^2\right\| - \sum_{i=1}^{N} \alpha_i [y_i(w^t \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{m} \beta_i \xi_i. \tag{10}$$

The dual optimization problem is attained by

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_j^i x_j - \sum_{i=1}^{N} \alpha_i \tag{11}$$

with constraints,

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C \quad \forall i = 1, 2, ..., N, \tag{12}$$

and solution of $w = \sum_{i=1}^{N} \alpha_i y_i x_i$, the decision function of $\mathrm{sign}(w^t \cdot x + b)$.

So far, the soft margin equation construction was discussed. The following section presents the hard margin construction for the transformation $\phi: R^t \to R^n$, then Eq. (11) is written as:

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i) \phi(x_j) - \sum_{i=1}^{N} \alpha_i. \tag{13}$$

The optimization function is $\mathrm{sign}(w^t \phi(x) + b)$, where $w = \sum_{i=1}^{N} y_i \alpha_i \phi(x_i)$.

The function $K(x, x') = \phi(x)' \phi(x')$ represents the kernel function, Eq. (13) is represented as

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i. \tag{14}$$

Subject to the constraints

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C \quad \forall i = 1, 2, ..., N.$$

$K(x, x')$ represents the kernel function; in the proposed study, a radial basis function is presented and is given by

$$K(x, x') = \exp(-\delta \left\| a - b \right\|^2). \tag{15}$$

To perform multiclass classification, in the proposed approach the one-vs-one method is employed to predict the quality of the food sample under study. For n-events, the one-vs-one strategy constructs $N = n(n-1)/2$ number of binary classifiers, then they are combined as multiclass by the one against two classes i, j as:

$$\min_{w^{ij}, b^{ij}, \zeta^{ij}} \frac{1}{2} (w^{ij})^T (w^{ij}) + \psi \sum_t \zeta_t^{ij} (w^{ij})^T. \tag{16}$$

Subject to

$$(w^{ij})^T \phi(x_t) + b^{ij} \ge 1 - \zeta_t^{ij} \qquad \text{if} \qquad y_t = i,$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \geq -1 + \zeta_t^{ij} \qquad \text{if} \qquad y_t = j,$$
$$\zeta_t^{ij} \geq 0,$$

where the slack variable is ζ_t^{ij}, ψ is the tradeoff parameter between the margin size and the error, and the majority voting scheme is employed here to choose the class of similar index. The pseudo-code of the proposed algorithm is presented as follows.

Pseudo-code of the proposed algorithm

Input: Training dataset $\{(x_i, y_i),\ \ i = 1, 2, 3, ..., n\}$
Let $X = \{(x_i^{\text{com}},\ y_i^{\text{com}}),\quad i = 1, 2, 3, ..., n\}$ are the data with no sensor failure scenario
$Y = \left\{(x_i^{\text{in\_com}}, y_i^{\text{in\_com}}),\quad i = 1, 2, 3, ..., n\right\}$ are the data with missing data field due to sensor failure scenario
Output : $W \in$ class: 'excellent', 'good', 'acceptable', 'spoiled'
parameters $\leftarrow C = 10^{-3}$ and $\lambda = 1000$
Stopping condition $\leftarrow$ iter_Max||Min_err_rate
Initialize: $w \leftarrow 0$ to 1, a $\leftarrow 0$, $\eta \leftarrow 0.5$ (threshold)
For $i = 1 : n$
For $j = 1 : m$
Calculate distance between y_i and x_j
Cum_dist = sum (distance of y_i to X)
$W_i = |X|\,/2 *$ Cum_dist
For $m = 1$: in_com
$d = 0$
for $n = 1$: com
if (class of x_n^{com} and $x_m^{\text{in\_com}}$ is same)
$d = W_m *$ dist_$x_n^{\text{com}}$to dist_$x_n^{\text{in_com}}$
if $(d < \eta)$
include x_n^{com} as ref point
replace $x_m^{\text{in\_com}}$ with average of ref points
While (iter_Max $= 500$||Min_err_rate $= 10^{-5}$)
Do {
For input data $i = 1$ to n
$s_i \leftarrow w^T x_i$// Prediction score
$a_{i-1} \leftarrow a_i$// load previous cache value
Update C and λ
$w \leftarrow w + x_i(a_i - a_{i-1})^T$}
End

4.4. KNN-based data imputation methodology

The missing values in the dataset are replaced by weighted KNN-based imputation strategy, see Cheng and Huang [22]. The first step of data pre-processing is normalization, which converts the dataset into a range of [0–1], and a min-max normalization technique is employed here to normalize the data samples. The nearest neighbor is estimated by identifying the distance between the complete and incomplete missing samples, based on the distance value weight vector is assigned as $W = [w_1, w_2, ..., w_n]$,

$$w_i = \frac{|X|}{2 \sum\limits_{x \in X} \|y_i - x_i\|^{2/(q-1)}}, \tag{17}$$

where w_i is the incomplete weight value of i-th incomplete data, $|X|$ is the training data without missing values, y_i is the i-th instance of incomplete data, x_i is the i-th complete data, and $q = 2$ to define the Euclidean distance between complete and incomplete data.

Now the weighted distance is estimated as

$$D_{ij} = [d_{i1}, d_{i2}, ..., d_{in}], \qquad d_{ij} = w_i(y_i - x_j)^2,$$

if $d_{ij} < 2$ then the data belong to the outliner.

Replace the missing data with the average of reference values $M(i, j, X) = X/n$, where X represents the neighbor vector and n is the number of neighbor set, (i, j) represents the i-th instance of the j-th attribute.

5. Discussion of obtained results

The proposed food quality analyzer is initially validated by benchmark datasets, and the data available in the public domain is adopted to validate the performance of the proposed model. Two food quality datasets are employed in the study, the wine quality check is made with the dataset of Rodriguez Gamboa *et al.* [10]. The dataset consist of eight field data flagged into three class of quality such as AQ: average-quality, HQ: high-quality, and LQ: low-quality. The meat quality check is made with the dataset of Wijaya *et al.* [7], and the dataset consists of four classes of data representing 'excellent', 'good', 'acceptable', and 'spoiled' quality of food. The complete dataset is normalized and segregated into 70% of data for training and 30% for testing. While testing, 10% of the data is made empty so that the model is made ready for sensor failure situations with the help of the proposed KNN imputation strategy. The following performance metrics are employed to study the performance of the proposed model in the study.

Accuracy. Accuracy is the quantity that depicts the quality of the right prediction made by the classifier:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{18}$$

where TP – true positive, TN – true negative, FP – false positive, FN – false negative.

Precision. Precision is a measure that demonstrates the degree of abnormal activity predicted as abnormal by the classifier model:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{19}$$

Recall/true positive rate (sensitivity). True positive rate (TPR) illustrates the extent of positive information emphases that are effectively considered as positive by the classifier, considering each and every single positive datum of the data sample:

$$\text{Recall or TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{20}$$

True negative rate (specificity). True negative rate (TNR) illustrates the extent of false information emphases that are effectively considered as negative by the classifier, considering each and every single negative datum of the data sample:

$$\text{Specificity or TNR} = \frac{\text{TN}}{\text{FP} + \text{TN}}. \tag{21}$$

F1 score. The harmonic mean of precision and recall is presented by the F1 score:

$$\text{F1 score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}. \tag{22}$$

The obtained results for wine quality assurance are presented in Fig. 3, where the obtained performance metric results for data without loss and data with loss are depicted, and the significance of the proposed KNN data imputation strategy is illustrated by the obtained metric values. The obtained results are tabulated in Table 3. The proposed model reports better classification performance for missing data scenario. The accuracy of the dataset without missing value is 0.9766, the missing dataset without KNN imputation strategy reported is 0.7832, whereas the proposed model reports 0.9426 for sensor failure scenarios. The proposed food quality assurance model reports 18% improved accuracy rate for sensor failure situation and 16.44% improved precision rate, 15.58% improved sensitivity, 21.68% of selectivity, and 15.96% of F1 score. Similarly, the performance of the model for meat quality check is demonstrated in Fig. 4, the performance of the proposed model for sensor failure scenario is improved about

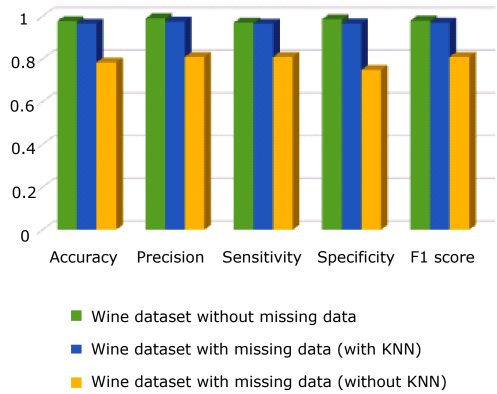

FIG. 3. Performance of the proposed model for the wine dataset.

TABLE 3. Classification performance for wine and meat dataset.

Dataset	Accuracy	Precision	Sensitivity	Specificity	F1 score
Wine dataset					
No sensor failure scenario	0.9766	0.9896	0.9699	0.9859	0.9796
Sensor failure scenario without KNN-based data imputation	0.7832	0.8096	0.8095	0.7484	0.8096
Sensor failure scenario with KNN-based data imputation	0.9648	0.9740	0.9645	0.9652	0.9692
Meat dataset					
No sensor failure scenario	0.9784	0.9902	0.9724	0.9868	0.9813
Sensor failure scenario without KNN-based data imputation	0.7415	0.7879	0.7650	0.7082	0.7763
Sensor failure scenario with KNN-based data imputation	0.9717	0.9896	0.9617	0.9857	0.9754

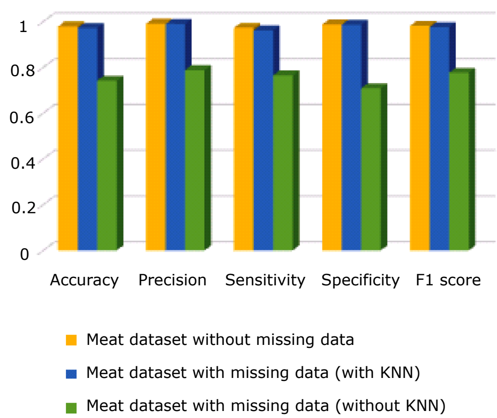

FIG. 4. Performance of the proposed model for the meat dataset.

23% of accuracy, 20% of precision, 19.67% of sensitivity, 27.7% of specificity, and 19.9% of F1 score.

The proposed model is experimentally validated with meat substrate, and the collected sensor data in the sensing chamber is fed into the cloud using Arduino controller. The data is stored in the cloud. Figures 5 and 6 show the field graph representing the data uploaded into the ThinkSpeak cloud. The proposed model employed nine gas sensors for sensing the aroma produced by the meat substrate under study. In addition, a temperature and humidity sensor is also employed to check the quality of food, so a total of eleven field data is uploaded into the ThinkSpeak cloud to validate the performance of the model for sensor failure situation by randomly disconnecting any of two sensors. The corresponding field chart is represented in Fig. 6. The data from the cloud is

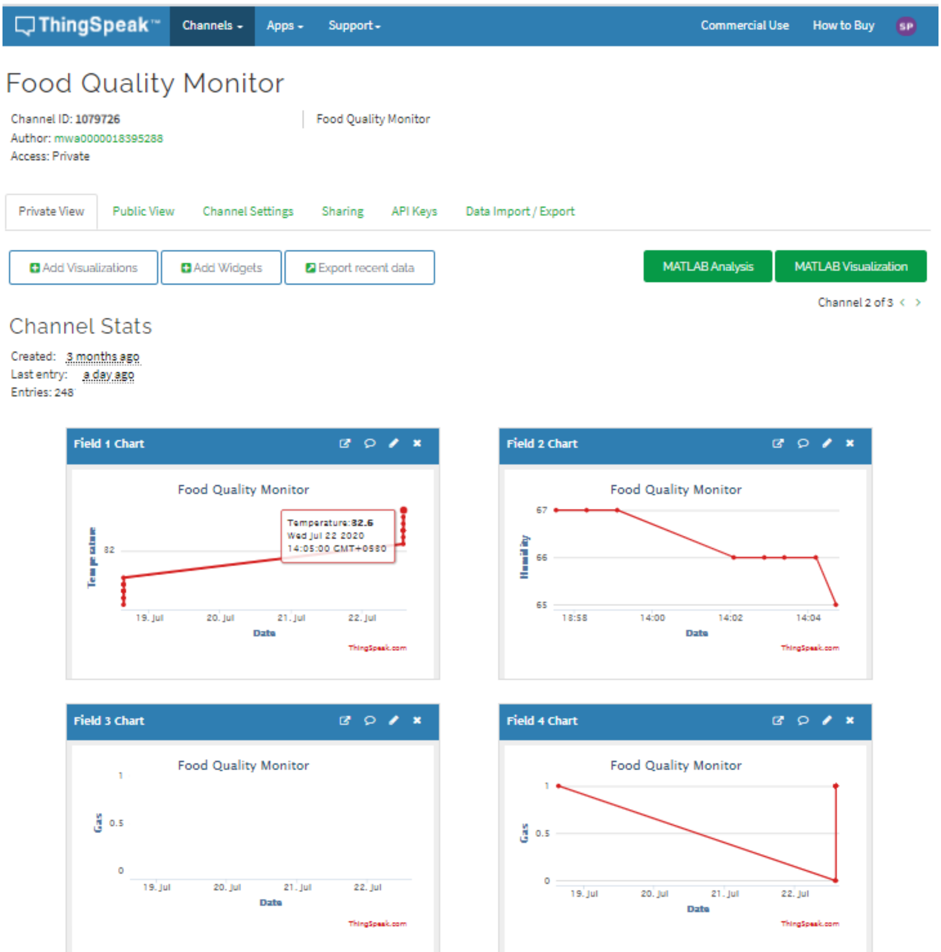

Fig. 5. Field chart of the ThinkSpeak cloud.

FIG. 6. Field chart in the ThinkSpeak cloud representing the missing field.

exported into the Matlab environment to classify the quality of the sample under study, and the classification performance of the model for both the complete dataset and the missing dataset is presented in Fig. 7, the reported quality is sent as mail for the user analyzing the sample through the IFTTT application. The message notification is represented in Fig. 8.

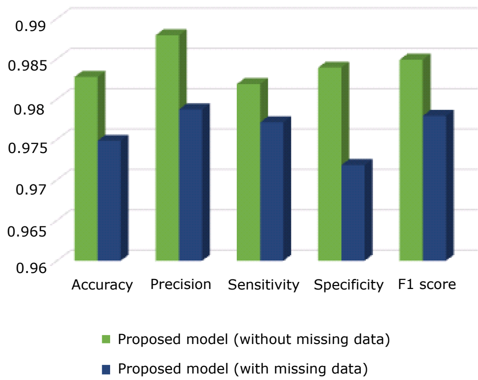

FIG. 7. Model performance for the meat substrate under study.

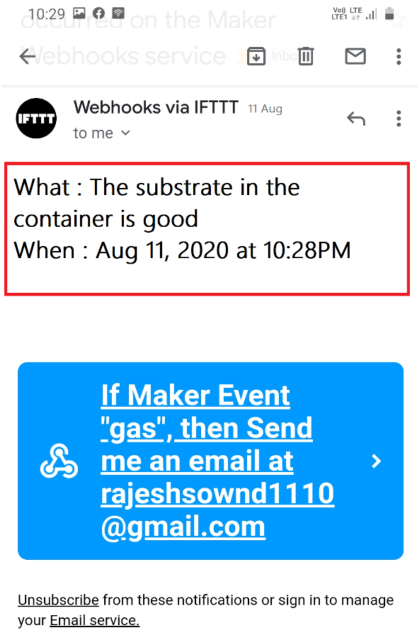

FIG. 8. Mail communication about quality of food.

In order to demonstrate the significance of the proposed study, the model is compared to other models in existing works of literature and the corresponding performance is reported in Table 4. The proposed KNN-based data imputation strategy with multiclass SVM model outperformed other models in the literature with better classification results. From the table, it is inferred that the proposed data imputation strategy sustained the performance of the model during the sensor failure scenario, whereas the other models existing works of literature failed to perform at that sensor failure condition.

Table 4. Comparative analysis with existing works of literature.

Model under study	Accuracy	Precision	Sensitivity	Specificity	F1 score
Model performance without sensor failure – meat substrate					
Proposed model	0.9827	0.9879	0.9819	0.9839	0.9849
Sanaeifar *et al.* [2]	0.9559	0.9662	0.9568	0.9548	0.9615
Voss *et al.* [13]	0.9606	0.9655	0.9654	0.9544	0.9654
Kaya *et al.* [20]	0.9736	0.9817	0.9722	0.9755	0.9769
Wijaya *et al.* [7]	0.9523	0.9522	0.9524	0.9523	0.9642
Han *et al.* [9]	0.9549	0.9523	0.9316	0.9432	0.9614
Kalpana and Baghyam [11]	0.9634	0.9712	0.9502	0.9685	0.9703
Model performance with sensor failure scenario – meat substrate					
Proposed model	0.9748	0.9787	0.9771	0.9718	0.9779
Sanaeifar *et al.* [2]	0.7282	0.7921	0.7460	0.7008	0.7684
Voss *et al.* [13]	0.7832	0.8875	0.7678	0.8128	0.8233
Kaya *et al.* [20]	0.8476	0.9162	0.8328	0.8724	0.8725
Wijaya *et al.* [7]	0.7824	0.7475	0.7633	0.7254	0.7434
Han *et al.* [9]	0.8623	0.8628	0.8648	0.8544	0.8653
Kalpana and Baghyam [11]	0.8453	0.7682	0.7468	0.7692	0.7972

6. Conclusion

This article presented a novel approach to ensure quality of food using an e-nose strategy in the sensor failure scenario. The uncertainty of the missing data during training and testing of the neural network was handled by the proposed KNN-based data imputation strategy. The food samples from wine and meat were classified into four classes; 'excellent', 'good', 'acceptable', and 'spoiled' using a multiclass SVM classifier algorithm. The proposed model outperformed other models in the literature for both sensor performances with and without failure scenario with improved classification performance.

Conflict of interest

The authors claim no conflict of interest in publishing this work.

References

1. A. Sanaeifar, S.S. Mohtasebi, M. Ghasemi-Varnamkhasti, H. Ahmadi, J. Lozano, Development and application of a new low cost electronic nose for the ripeness monitoring of banana using computational techniques (PCA, LDA, SIMCA and SVM), *Czech Journal of Food Sciences*, **32**(6): 538–548, 2014, doi: 10.17221/113/2014-CJFS.

2. A. Sanaeifar, S.S. Mohtasebi, M. Ghasemi-Varnamkhasti, M. Siadat, Application of an electronic nose system coupled with artificial neural network for classification of banana samples during shelf-life process, [in:] *2014 International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 753–757, IEEE, 2014, November, doi: 10.1109/CoDIT.2014.6996991.

3. A. Sanaeifar, S.S. Mohtasebi, M. Ghasemi-Varnamkhasti, H. Ahmadi, Application of MOS based electronic nose for the prediction of banana quality properties, *Measurement*, **82**: 105–114, 2016, doi: 10.1016/j.measurement.2015.12.041.

4. M.F. Adak, N. Yumusak, Classification of e-nose aroma data of four fruit types by ABC-based neural network, *Sensors*, **16**(3): 304, 2016, doi: 10.3390/s16030304.

5. S. Kiani, S. Minaei, M. Ghasemi-Varnamkhasti, A portable electronic nose as an expert system for aroma-based classification of saffron, *Chemometrics and Intelligent Laboratory Systems*, **156**(15): 148–156, 2016.

6. E. Ordukaya, B. Karlik, Quality control of olive oils using machine learning and electronic nose, *Journal of Food Quality*, **2017**: Article ID 9272404, doi: 10.1155/2017/9272404.

7. D.R. Wijaya, R. Sarno, E. Zulaika, Electronic nose dataset for beef quality monitoring in uncontrolled ambient conditions, *Data in Brief*, **21**: 2414–2420, 2018, doi: 10.1016/j.dib.2018.11.091.

8. J. Thorson, A. Collier-Oxandale, M. Hannigan, Using a low-cost sensor array and machine learning techniques to detect complex pollutant mixtures and identify likely sources, *Sensors*, **19**(17): 3723, 2019, doi: 10.3390/s19173723.

9. L. Han, C. Yu, K. Xiao, X. Zhao, A new method of mixed gas identification based on a convolutional neural network for time series classification, *Sensors*, **19**(9): 1960, 2019, doi: 10.3390/s19091960.

10. J.C. Rodriguez Gamboa, E.S. Albarracin E., A.J. da Silva, T.A.E. Ferreira, Electronic nose dataset for detection of wine spoilage thresholds, *Data in Brief*, **25**: 104202, 2019, doi: 10.1016/j.dib.2019.104202.

11. S. Kalpana, A.L. Baghyam, Electronic-nose system for classification of fruits and freshness measurement using K-NN algorithm, *International Journal of Innovative Technology and Exploring Engineering*, **8**(6S4): 641–644, 2019, doi: 10.35940/ijitee.F1132.0486S419.

12. M. Ghasemi-Varnamkhasti, A. Mohammad-Razdari, S.H. Yoosefian, Z. Izadi, G. Rabiei, Selection of an optimized metal oxide semiconductor sensor (MOS) array for freshness characterization of strawberry in polymer packages using response surface method (RSM), *Postharvest Biology and Technology*, **151**: 53–60, 2019, doi: 10.1016/j.postharvbio.2019.01.016.

13. H.G.J. Voss, J.J.A. Mendes Júnior, M.E. Farinelli, S.L. Stevan, A prototype to detect the alcohol content of beers based on an electronic nose, *Sensors*, **19**(11): 2646, 2019, doi: 10.3390/s19112646.

14. M.A. Hasan, R. Sarno, S.I. Sabilla, Optimizing machine learning parameters for classifying the sweetness of pineapple aroma using electronic nose, *International Journal of Intelligent Engineering and Systems*, **13**(5): 122–132, 2020, doi: 10.22266/ijies2020.1031.12.

15. S. Rahman, A.S. Alwadie, M. Irfan, R. Nawaz, M. Raza, E. Javed, M. Awais, Wireless e-nose sensors to detect volatile organic gases through multivariate analysis, *Micromachines*, **11**(6): 597, 2020, doi: 10.3390/mi11060597.

16. B. Nouri, S.S. Mohtasebi, S. Rafiee, Quality detection of pomegranate fruit infected with fungal disease, *International Journal of Food Properties*, **23**(1): 9–21, 2020, doi: 10.1080/10942912.2019.1705851.

17. N. Aghilinategh, M.J. Dalvand, A. Anvar, Detection of ripeness grades of berries using an electronic nose, *Food Science & Nutrition*, **8**(9): 4919–4928, 2020, doi: 10.1002/fsn3.1788.

18. J.S. Manoharan, Study of variants of extreme learning machine (ELM) brands and its performance measure on classification algorithm, *Journal of Soft Computing Paradigm (JSCP)*, **3**(2): 83–95, 2021, doi: 10.36548/jscp.2021.2.003.

19. A.P. Pandian, Performance Evaluation and comparison using deep learning techniques in sentiment analysis, *Journal of Soft Computing Paradigm*, **3**(2): 123–134, 2021, doi: 10.36548/jscp.2021.2.006.

20. A. Kaya, A.S. Keçeli, C. Catal, B. Tekinerdogan, Sensor failure tolerable machine learning-based food quality prediction model, *Sensors*, **20**(11): 3173, 2020, doi: 10.3390/s20113173.

21. C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, **20**(3): 273–297, 1995, doi: 10.1007/BF00994018.

22. C. Cheng, H. Huang, A distance-threshold kNN method for imputing medical data missing values, *Journal of Advances in Computer Networks*, **7**(1): 13–17, 2019, doi: 10.18178/JACN.2019.7.1.265.

Received November 21, 2021; revised version January 16, 2022;
accepted January 22, 2022.