Journal of Advanced Research in Natural and Applied Sciences
dergipark.org.tr/tr/pub/jarnas
Open Access

ISSN: 2757-5195 | 11 (3) (2025) 276-285 | doi.org/10.28979/jarnas.1705283

GP-Optimized E-Nose: A Low-Cost Solution for Apple Pesticide Detection

Ahmed Yılmaz[1], Cemaleddin Şimşek[2]

Article Info
Received: 23 May 2025
Accepted: 02 Sep 2025
Published: 30 Sep 2025
Research Article

Abstract— This study presents a cost-effective, user-friendly electronic nose (e-nose) system for detecting Malathion pesticide residues on apples. Genetic Programming (GP) was utilized for system optimization. Adapting a previous methodology with different fruits and pesticides, we analyzed 100 apple samples (sprayed and control) using an 11-sensor e-nose. GP was employed to classify data, derive sensor output equations, and select optimal sensor combinations, by aiming to reduce hardware costs and computational load. The optimized system achieved a significant level of classification success. It demonstrated approximately 92% validation accuracy with only three sensors and achieved 100% validation accuracy when six sensors were used. These findings offer a viable, accurate, and economical alternative for pesticide detection. Future work is planned to expand the database and develop a home-use device.

Keywords — *Electronic nose (e-nose), pesticide detection, genetic programming (GP), Malathion, apple, classification*

1. Introduction

Agriculture is of vital importance for the continuity of food production worldwide [1]. The pesticides used to combat plant diseases and pests play a significant role in the sustainability of this sector. While these pesticides are used to increase product efficiency and food safety, they also have significant unintended side effects [2]. Problems such as the excessive use of chemical pesticides [3], the risks they pose to the environment and human health [4], and the chemical residues they leave in agricultural products and soil [5] are among the negative consequences of spraying. Therefore, the development and quantification of effective and sustainable spraying methods are of great importance [6]. In this study, we adapted the material and method from the study by [7] to achieve similar results with a lower-cost and user-friendly approach. For this purpose, experiments were repeated using apple fruit and Malathion pesticide instead of the cherry fruit and Diazinon pesticide examined in [7], and optimization was performed on the developed e-nose system. Malathion and Diazinon are organophosphate insecticides widely used to combat harmful insects [8]. Malathion is effective against several insects that can damage apple crops, including aphids, apple maggot, codling moth, fruit flies, etc., and its effectiveness against certain apple pests is well-documented [9]. It's a broad-spectrum insecticide, meaning it can target multiple types of insects with a single application. Compared to some other insecticides, Malathion has a relatively short residual life [4], which can be beneficial closer to harvest time, provided pre-harvest

[1]yilmazahmet@kmu.edu.tr; [2]csimsek@kmu.edu.tr (Corresponding Author)
[1]Computer Engineering, Faculty of Engineering, Karamanoğlu Mehmetbey University, Karaman, Türkiye
[2]Electrical and Electronics Engineering, Faculty of Engineering, Karamanoğlu Mehmetbey University, Karaman, Türkiye

intervals are strictly followed [10]. During application, applying the pesticide at the correct dose and timing is essential for both its effectiveness and residue levels [11]. Although the half-life of Malathion in the environment is relatively short, strict adherence to manufacturer instructions, the use of personal protective equipment, and attention to the pre-harvest interval are necessary to minimize its potential effects on human health and the environment [12].

The primary objective of the optimization in this study is to reduce the cost of the electronic nose system for home use by minimizing the number of sensors involved, while maintaining accurate detection of improper pesticide application. This optimization focuses on identifying the most informative subset of sensors among the 11 included in the system. Instead of relying on complex or opaque dimensionality reduction techniques, we employ Genetic Programming (GP) [15, 16] to evolve interpretable mathematical expressions that perform both classification and implicit feature selection. For this purpose, a total of 50 apple samples were collected from 10 apple trees in four different maturity periods, to which the Sanctum brand [13, 14] Malathion pesticide, recommended by the Ministry of Agriculture and Forestry of the Republic of Türkiye, was applied by mixing it with 40 ml/100 L of water. The same number of untreated apple samples was also collected as a control group, resulting in a total of 100 samples. The collected samples were transported to the experimental environment in a thermally insulated box within 24 hours for analysis. In the experimental environment, an e-nose system with the same sensor specifications as those used in the study by Tozlu et al. (2024) [7] was employed (MQ-2, MQ-3, MQ-4, MQ-5, MQ-6, MQ-7, MQ-8, MQ-9, MQ-131, MQ-135, and MQ-137 gas sensors). A total of 100 samples were analyzed with the e-nose system using 60-second measurement cycles.

The e-nose data obtained were classified using GP, both to generate equations between sensor values and the output class and to determine the most suitable sensor combination. This study aims to design a home-type pesticide detection device that reduces cost and processing complexity. This study differs from previous research in two key ways: (1) it integrates sensor selection directly into the learning process via GP, and (2) it emphasizes the development of a low-cost, interpretable, and practical solution suitable for household pesticide detection. This constitutes a novel contribution, demonstrating that GP can be effectively used to minimize sensor usage while maintaining high classification accuracy, thereby reducing both hardware costs and computational complexity. Unlike black-box classifiers or traditional feature selection methods, our approach produces compact mathematical models that are directly applicable to low-resource devices. To establish performance benchmarks, we additionally evaluated two conventional machine learning approaches, Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN), under identical experimental conditions (4-fold cross-validation using the same dataset splits). These baseline models, which utilize the complete set of 11 sensors, unlike our optimized GP solutions, provide essential reference points for assessing the relative performance of our method. The comparative results presented in Section 3 demonstrate how the GP-derived models achieve competitive detection accuracy while requiring significantly fewer sensor inputs.

2. Metodology

2.1. Data Collection and Description

Since both Malathion and Diazinon insecticides are organophosphate-based, we found it appropriate to replicate the content provided in the Materials and Methods section in [7]. Therefore, we repeated the electronic nose design, product procurement, and data collection methods, even though for a different fruit, while maintaining their integrity. The necessary precision was exercised in the apple collection, electronic nose design, and measurement acquisition to ensure repeatability of the experiment.

2.2. Sample Preparation

Pesticide-sprayed apples and non-pesticide-sprayed apples were collected from our fruit garden located in the central district of Karaman. The fruit harvesting stage was carried out manually without damaging the fruits to replicate the real-world effect. The apples in the control group, which were intended to be pesticide-free, were collected before the spraying process began. In contrast, the second group of apples was collected 24 hours after the relevant pesticide was prepared at a concentration of 40 ml/100 L of water and sprayed [17]. Application rates are typically given in kilograms of active ingredient per hectare (kg ai/ha) [18]. However, some labels also specify rates per liter of water for smaller applications [13], which is the approach we used in our own small fruit garden.

2.3. Electronic Nose System Design

In this study, experiments were conducted using a system similar to the one developed in [7], which investigated pesticides and cherry fruit of similar origin. For the electronic nose setup, 11 different gas sensors that are widely available on the market were assembled (MQ-2, MQ-3, MQ-4, MQ-5, MQ-6, MQ-7, MQ-8, MQ-9, MQ-131, MQ-135, MQ-137). This group of sensors was chosen not for their specificity to a single chemical compound, but rather to leverage their varying sensitivities to create a collective ”fingerprint” that allows for robust classification. The system was placed inside an odorless, sealed plastic container for measurement. The data connection between the hardware inside the container and the outside world was provided using a Deneyap evaluation module (EVM), developed by the T3 Foundation, which has wireless communication capabilities. Due to the high energy consumption of the gas sensors, 2S LiON 18650 battery packs provided voltage to the system, while the sensor voltages were constantly monitored to minimize calibration drift. All data were transferred to a PC via Wi-Fi, which enabled protection of the system from external volatile chemicals during measurement.

2.4. Genetic Programming

GP is an evolutionary computation technique that is a special form of genetic algorithms (GAs) [15, 16, 19]. It can autonomously produce programs, such as algorithms or computer programs, for a specific class of problems. GP is inspired by Darwin’s theory of natural selection and the principle of the survival of the fittest. It randomly initializes the programs in the population and evolves the population iteratively, considering the program’s fitness to the solution. Fitness is a metric that shows how efficiently the programs solve the problem. The programs are then selected using a selection mechanism, such as a tournament or a roulette wheel, that considers the fitness of the programs. The selection mechanisms typically ensure that more efficient programs have a higher chance of being selected for the reproduction of their components, which is analogous to the principle of survival of the fittest. GP has been widely used in many fields for various tasks, including evolving mathematical functions and designing electronic circuits.

The working principle of the GP is similar to that of the GA, but its representation and usage differ in aims from those of the GA. There are many representation forms of GP, such as tree-structured GP [15, 16], Linear GP [20], and Cartesian GP [21]. In this study, we employed the tree-based GP, a well-known one that was first introduced. Therefore, GP will refer to the tree-based GP for the rest of the paper. Figure 1 shows an example of a tree-based GP. The root and intermediate nodes of the tree are selected from the function set, while the outermost leaves of the tree are selected from the terminal set. The function set consists of arithmetic, logic, and problem-specific operators, while the terminal set includes input variables and some constants. The operators in the function set and constants in the terminal set are determined by the user, taking into account their suitability for the problem.

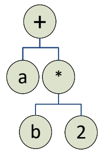

Figure 1. An example tree-based GP representation (Expression of the tree: a+2b)

The steps of the GP system used in this study are shown in Algorithm 1 and Figure 2. Initially, the programs in the population are randomly created in a tree structure using the elements of function and terminal. After making the initial population, the programs are evaluated using a fitness function that measures their efficiency in solving the problem. Then, the genetic operators, including selection, crossover, and mutation, are iteratively applied to the population to evolve the programs within it. The selection operator selects parents, which are fitter programs for reproduction. The crossover operator creates offspring by exchanging a random sub-tree of the parent trees to approximate the population to the solution. The mutation operator replaces a subtree in the parents or offspring to provide diversity in the population. The crossover operator is usually applied to the parents with a relatively high probability, while the mutation operator is frequently used to the relevant programs with a relatively low probability. This sequence of processes iterates over the generations, and thus, the population evolves toward the solution.

Algorithm 1: Tree-Based GP

```
input  :
    • Function set F
    • Terminal set T
    • Max Depth
    • Fitness function fitness()
    • Population size N
    • Maximum number of generations G
    • Probabilities for crossover, mutation, and reproduction
output : p* The best solution found
Initialize population P with N random programs (trees) generated using F, T, and Max Depth;
Set generation counter g ← 0;
for each program   p ϵ P : do
|   Evaluate fitness(p);
end
while g < G do
|   for i = 1 to N/2 do
|   |   p1, p2: Select two parents from P based on a selection method
|   |   if crossover probability is met then
|   |   |   Crossover: Exchange subtrees between p1 and p2 ;
|   |   end
|   |   if mutation probability is met then
|   |   |   Mutation: Replace a randomly selected subtree of corrosponding p with
|   |   |    a newly generated subtree;
|   |   end
|   |   Evaluate fitness(p1) and Evaluate fitness(p2);
|   end
|   Increment g ← g + 1;
end
Return: p∗ (the best solution found);
```

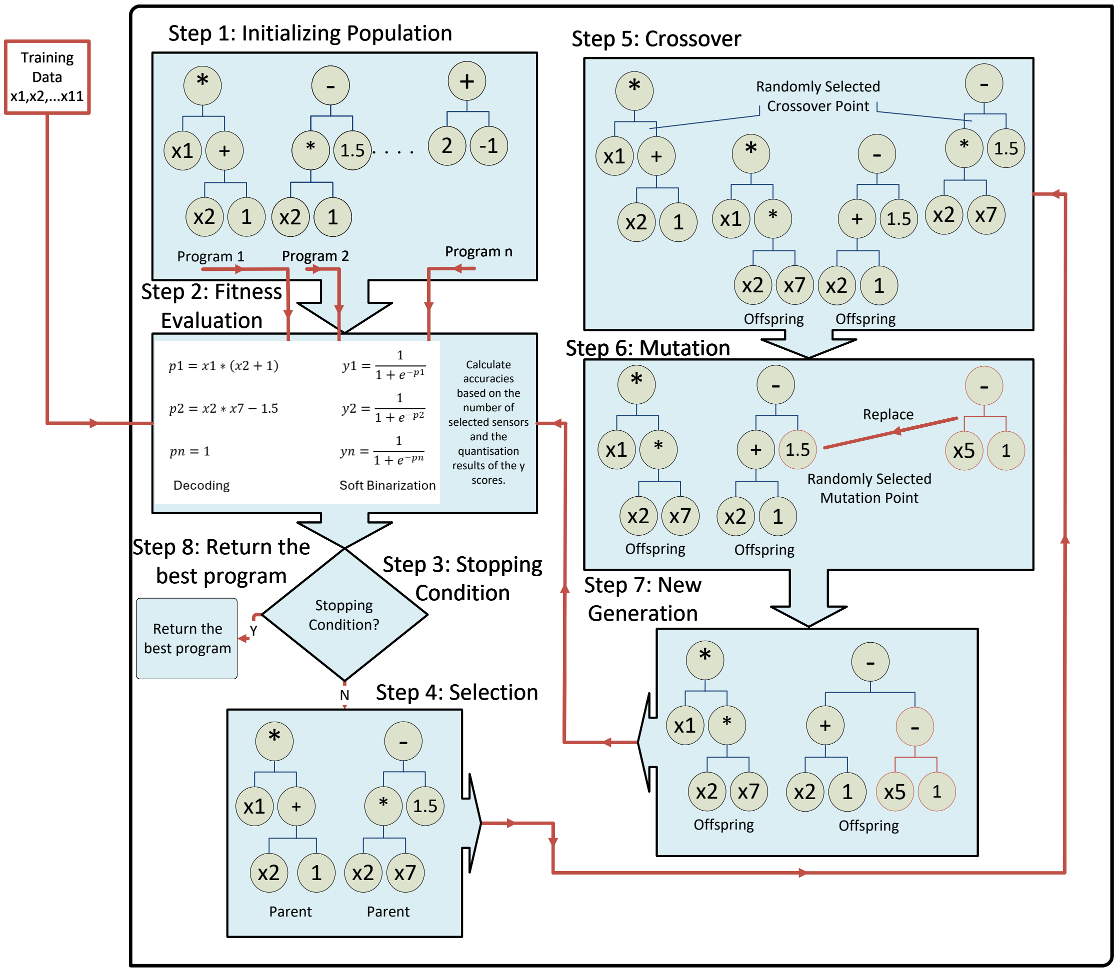

Figure 2. The general framework of the GP system

2.5. Hyperparameters of the GP system

As previously mentioned, a GP-based classification method was applied to determine the presence of pesticide residues in apples in this study. Data was collected via 11 different sensors, each measuring various gases for a specific period. Before being provided as input to the GP system, these 11 average values (x_{mean} in (2.1)) underwent min-max normalization. Thus, the terminal set of the GP consisted of 11 inputs collected via those sensors and constants randomly selected from the range $[-2, 2]$. The function set is limited to addition, subtraction, multiplication, and safe division operations.

$$x_{mean} = \frac{1}{s}\sum_{t=1}^{s} x(t) \tag{2.1}$$

We performed 4-fold cross-validation, dividing the dataset into two subsets for each fold: 75% for training and 25% for testing. The fitness values of the GP individuals were computed based on a combination of classification accuracy on the training set and the number of sensors used. While the number of sensors was included in the fitness calculation for all individuals, it was weighted with a small coefficient to minimize its overall impact on the calculation. As a result, it primarily influenced selection only when individuals achieved identical accuracy rates. The fitness function is provided in (2.2).

$$fitness = \alpha * accuracy - \beta * \frac{N_{sensors}}{N_{Total}} \tag{2.2}$$

where accuracy denotes the classification accuracy on the training set, $N_{sensors}$ represents the number of sensors employed by the individual, N_{Total} is the total number of available sensors, α is the weight assigned to the accuracy component (set to 1), and β is a small coefficient (set to 0.001) used to penalize sensor usage and thus minimize its impact on the overall fitness.

The initial population consisted of 1000 individuals, and the parent selection was made with the tournament selection with a size of 3. The crossover probability was set at 90%, and the mutation probability at 10% throughout the evolutionary process. The outputs of the evolved programs were fed as input to the logistic sigmoid function to model nonlinear decision boundaries, as shown in (2.3). The evolutionary process continued for a total of 100 generations.

$$logistic\ sigmoid\ function, f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

3. Experimental Results

3.1. Evolving Classification Function by GP

Since the GP is a stochastic method, the GP system was run 10 times in each fold, resulting in a total of 40 runs. In each run, accuracy, precision, recall, and F1-score metrics obtained from the validation set were recorded. The summarized results are presented in Table 1. The comprehensive evaluation of our genetic programming approach across four validation folds reveals several vital patterns in model performance. As shown in Table 1, Fold 1 demonstrates near-perfect recall, achieving 1.00 in 9 out of 10 runs with a single instance at 0.92, while maintaining consistent accuracy (0.92 ± 0.038) and F1-scores (0.92 ± 0.032). This remarkable consistency suggests robust generalization capabilities within this data partition. Fold 4, while showing slightly higher variability (accuracy SD = 0.101), produces our top-performing model capable of perfect classification. Fold 2 exhibits balanced precision (0.795) and recall (0.85), though with somewhat lower overall accuracy (0.80). Notably, Fold 3 maintains strong precision (median 1.00) but displays greater recall volatility (0.67–0.92 range), indicating potential sensitivity to specific pesticide signatures. These variations across folds highlight how different data partitions can influence the evolved solutions, with each fold producing models with distinct strengths in either stability, peak performance, or balanced metric optimization.

From the numerous solutions evolved by our GP framework, we selected two particularly representative models that best demonstrate the method's ability to balance accuracy and sensor efficiency. The first model, derived from Fold 4 Run 6, represents our peak-performance solution, achieving perfect classification (100% accuracy and F1-score) through the quadratic polynomial formula:

$$g_{\text{opt}}(x) = 2x_1 + x_2 - x_3 - x_6 - x_8 - (x_4 - 1)^2 \tag{3.1}$$

where x_i denotes the measurement from the i-th sensor indicated in Table 2.

This 6-sensor model demonstrates several critical features that contribute to its exceptional performance. The equation combines linear terms with distinct weights, where sensors x_1 and x_2 serve as the primary positive indicators while x_3, x_6, and x_8 contribute negatively to the decision function. The quadratic term $(x_4 - 1)^2$ creates a nonlinear decision boundary. The model's perfect classification performance suggests these specific sensor interactions capture the essential features for reliable pesticide detection.

For applications where sensor minimization is paramount, the Fold 1 Run 10 model provides an excellent compromise, delivering 92% accuracy and F1-score while using only three sensors through

Table 1. Performance Metrics Across Validation Folds

Metric	Std Dev	Median	Min	Max
Fold 1				
Accuracy	0.0377	0.92	0.84	0.96
Precision	0.0574	0.86	0.75	0.92
Recall	0.0253	1.00	0.92	1.00
F1-Score	0.0320	0.92	0.86	0.96
Fold 2				
Accuracy	0.0548	0.80	0.72	0.88
Precision	0.0657	0.795	0.69	0.92
Recall	0.0781	0.85	0.72	1.00
F1-Score	0.0524	0.80	0.72	0.90
Fold 3				
Accuracy	0.0389	0.88	0.84	0.96
Precision	0.0525	1.00	0.85	1.00
Recall	0.0865	0.83	0.67	0.92
F1-Score	0.0492	0.875	0.80	0.96
Fold 4				
Accuracy	0.1006	0.90	0.68	1.00
Precision	0.1119	0.865	0.67	1.00
Recall	0.0760	0.92	0.77	1.00
F1-Score	0.0890	0.905	0.71	1.00

the rational function:

$$g_{\text{eff}}(x) = 2x_1 - \frac{0.5}{0.5 + x_2 - x_6} \tag{3.2}$$

This efficient solution maintains strong performance through a computationally simple form requiring just five arithmetic operations. The equation structure reveals several essential characteristics: the dominant $2x_1$ term provides the primary detection signal, while the rational component introduces nonlinear behavior that refines the classification based on the interaction between sensors x_2 and x_6. Remarkably, this model achieves its performance using 72% fewer sensors than the full array, making it particularly suitable for resource-constrained implementations.

Table 2. Available and Selected Sensors with Accuracy Performance

Sensor Name	MQ 2	MQ 3	MQ 4	MQ 5	MQ 6	MQ 7	MQ 8	MQ 9	MQ 131	MQ 135	MQ 137
Symbol	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
$g_{\text{opt}}(x)$	1	1	1	1	0	1	0	1	0	0	0
$g_{\text{eff}}(x)$	1	1	0	0	0	1	0	0	0	0	0

*Chosen Sensor Amount Among Eleven Sensors

3.2. Comparative Performance Analysis

For comparative performance evaluation, we implemented two established machine learning algorithms - SVM and k-NN - as baseline models. These conventional approaches were evaluated using identical experimental protocols, including the same 4-fold cross-validation scheme with equivalent training/test splits, as well as similar performance metrics (accuracy, precision, recall, and F1-score) applied to our GP solution. The complete results of these baseline evaluations are systematically presented in Table 3, enabling direct comparison with the GP-derived models. Both baseline methods were configured with standard hyperparameters (SVM with an RBF kernel, C = 1.0; k-NN with k = 5) and, significantly, utilized the full complement of 11 available sensors, unlike our sensor-optimized GP solutions. This evaluation framework ensures fair comparison while highlighting the unique advantages of the GP approach in subsequent analysis.

Table 3. Baseline Classifier Performance Across Folds

Model	Fold	Accuracy	Precision	Recall	F1-Score
SVM	Fold 1	0.96	0.92	1.00	0.96
	Fold 2	0.76	0.73	0.85	0.79
	Fold 3	0.92	1.00	0.83	0.91
	Fold 4	0.88	0.86	0.92	0.89
k-NN	Fold 1	0.92	0.86	1.00	0.92
	Fold 2	0.68	0.65	0.85	0.73
	Fold 3	0.88	0.85	0.92	0.88
	Fold 4	0.96	1.00	0.92	0.96

The evaluation of baseline classifiers provides crucial insights when compared with our GP approach. Although traditional methods show strong peak performance — SVM achieves an accuracy of 0.96 and an F1-score of 0.96 in Fold 1, and k-NN reaches an accuracy of 0.96 and an F1-score of 0.96 in Fold 4 — both exhibit concerning volatility. SVM's accuracy drops to 0.76 in Fold 2, while k-NN falls to 0.68 in the same fold, revealing significant sensitivity to data variations. However, these traditional methods exhibit critical limitations that our GP solution overcomes. Most significantly, they require complete sensor arrays (all 11 sensors) for operation. In contrast, our GP-evolved models achieve comparable performance using only 3-4 strategically selected sensors, offering substantial advantages in hardware implementation and power efficiency.

Our GP solutions demonstrate three key advantages:

- **Sensor Efficiency**: The 3-sensor GP model, g_{eff}, (0.92 accuracy) demonstrates that comparable performance to baseline averages (k-NN: 0.86, SVM: 0.88) can be achieved with 73% fewer sensors. While not exceeding the baselines' peak performance, this represents a favorable trade-off for resource-constrained applications.
- **Peak Performance**: Our 6-sensor GP model, g_{opt}, reaches perfect 1.00 accuracy, outperforming the best baseline results but with 45% fewer sensors.
- **Interpretability**: The mathematical forms explicitly show sensor relationships, unlike black-box decision mechanisms of SVM and k-NN.

This combination of sensor optimization, performance superiority, and model interpretability positions GP as particularly suitable for pesticide screening devices, where cost, efficiency, and reliability are key considerations.

4. Conclusion

This study aimed to build upon an existing pesticide detection system from the literature, in terms of its methodology and data collection system, by applying it to a different fruit and using a pesticide of similar origin, with a significant focus on modifying and optimizing the Genetic Programming (GP) based evaluation algorithm. The goal was to develop a GP-based classification formula that optimizes the number of sensors, thereby improving upon the existing method in terms of both hardware cost and computational load. This was achieved by strategically selecting a group of low-cost sensors, leveraging their collective and varied sensitivities to create a unique chemical ”fingerprint” for robust classification, rather than relying on a single specific chemical marker. The results obtained support the achievement of this aim. Future studies can focus on testing various pesticide and fruit combinations to expand the database and transform the device into an economical format suitable for home users, as well as to develop the most robust classification system. To this end, researchers can plan to examine the most consumed fruits in our country using similar equipment to that in the current publication.

Author Contributions

All authors contributed equally to this work.

Conflicts of Interest

All the authors declare no conflict of interest.

Ethical Review and Approval

No approval from the Board of Ethics is required.

References

[1] M. Wesoły, W. Przewodowski, P. Ciosek-Skibińska, *Electronic noses and electronic tongues for agricultural purposes*, TrAC Trends in Analytical Chemistry 164 (2023) 117082.

[2] D. Chen, B. Wang, T. Zhang, Z. Chang, *Towards accuracy recognition and content estimation of typical pesticides in groundwater via electronic nose*, Sensors and Actuators A: Physical 362 (2023) 114642.

[3] J. Qiao, Y. Lv, Y. Feng, C. Liu, Y. Zhang, J. Li, X. Weng, *Research on soil pesticide residue detection using an electronic nose based on hybrid models*, Agronomy 14 (4) (2024) 766.

[4] C. M. Durán Acevedo, D. D. Cárdenas Niño, J. K. Carrillo Gómez, *Sensory perception systems and machine learning methods for pesticide detection in fruits*, Applied Sciences 14(17) (2024) 8074.

[5] T. Seesaard, N. Goel, M.Kumar, C. Wongchoosuk, *Advances in gas sensors and electronic nose technologies for agricultural cycle applications*, Computers and Electronics in Agriculture 193 (2022) 106673.

[6] B. Wang, D. Chen, X. Weng, Z. Chang, *Development an electronic nose to recognize pesticides in groundwater*, Talanta 269 (2024) 125506.

[7] B. H. Tozlu, *Electronic detection of pesticide residues on cherry fruits*, International Journal of Computational and Experimental Science and Engineering 10(3) (2024) 364–370.

[8] M. M. Bordbar, T. A. Nguyen, A. Q. Tran, H. Bagheri, *Optoelectronic nose based on an origami paper sensor for selective detection of pesticide aerosols*, Scientific Reports 10 (2020) 17412.

[9] A. Amkor, N. El Barbri, *A measurement prototype based on gas sensors for detection of pesticide residues in edible mint*, Journal of Food Measurement and Characterization 15 (2021) 170–180.

[10] M. D. H. Prodhan, M. W. Akon, S. N. Alam, *Determination of pre-harvest interval for quinalphos, Malathion, Diazinon and cypermethrin in major vegetables*, Journal of Environmental and Analytical Toxicology 8(1) (2018).

[11] F. Malhat, M. Abdel-Megeed, E. S. Saber, S. A. S. Shokr, A. N. Saber, *Monitoring and risk assessment of pesticide residues in bananas: Insights from Egypt*, Journal of Food Composition and Analysis 143 (2025) 107610.

[12] Y. Liu, S. Liu, Y. Zhang, D. Qin, Z. Zheng, G. Zhu, X. Li, *The degradation behaviour, residue distribution, and dietary risk assessment of malathion on vegetables and fruits in China by GC-FPD*, Food Control 107 (2020) 106754.

[13] M. H. Kodandaram, K. Banerjee, A. B. Rai, B. Singh, *Pre-and post-harvest pesticide contamination management for production of quality vegetables*, In: Advances in Postharvest Technologies of Vegetable Crops (2018) 443–483.

[14] T.C. Tarım ve Orman Bakanlığı, Accessed 23 May 2025, https://bku.tarimorman.gov.tr/.

[15] J. R. Koza, *On the programming of computers by means of natural selection*, MIT Press, 1993.

[16] J. R. Koza, *Genetic programming as a means for programming computers by natural selection*, Statistics and Computing 4(2) (1994) 87–112.

[17] M. S. Ahmed, A. Begum, D. Sarker, *Determination of pre-harvest interval for dimethoate and quinalphos in selected vegetables*, Asian-Australasian Journal of Bioscience and Biotechnology 5(1) (2020) 42–47.

[18] F. Malhat, K. M. Kasiotis, S. Shalaby, *Magnitude of cyantraniliprole residues in tomato following open field application: pre-harvest interval determination and risk assessment*, Environmental Monitoring and Assessment 190 (2018) 1–10.

[19] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, MIT Press, 1992.

[20] M. O'Neill, Riccardo Poli, William B. Langdon, Nicholas F. McPhee, A field guide to genetic programming, 2008, Accessed 23 May 2025 http://www.gp-field-guide.org.uk.

[21] J. Miller, A. Turner, *Cartesian genetic programming*, in: A. I. E. Alcazar (Ed.), Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, (2015) pp. 179–198.