IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3816~3826
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3816-3826

Artificial intelligence-powered smart roads: leveraging orange3 for traffic signs recognition

Areen Arabiat[1], Muneera Altayeb[1], Sanaa Salama[2]
[1]Department of Communications and Computer Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Amman, Jordan
[2]Department of Biomedical Engineering, Faculty of Engineering, Arab American University, Jenin, Palestine

Article Info

Article history:

Received Sep 25, 2024
Revised Jun 21, 2025
Accepted Jul 10, 2025

Keywords:

AdaBoost
Advance driver assistance systems
Performance metrices
Random forest
Traffic signs

ABSTRACT

Traffic sign recognition systems are an important concern of advance driver assistance systems (ADAS) and intelligent autonomous vehicles. Recently, many studies have emerged that aim to employ artificial intelligence (AI) and machine learning (ML) to detect and classify traffic signs to improve a system that can be embedded in vehicles to increase efficiency and safety. This work's primary goal is to address traffic sign identification and recognition utilizing a 2,339-image open-source dataset from Kaggle. Our detection model for extracting and classifying traffic sign suggestions is built using Orange3 data mining tools, based on four classifiers random forest (RF), k-nearest neighbors (KNN), decision tree (DT), and adaptive boosting (AdaBoost). Signs are classified into eight categories: don't go signs, go signs, horn signs, roundabout signs, danger signs, crossing signs, speed limit sign, and unallowed signs. The results of examining and evaluating the proposed model based on the performance evaluation metrics showed that RF outperformed with an accuracy rate of 99.8%, followed by AdaBoost with a classification accuracy of 99.2%, and the classification accuracy of DT and KNN was 98.3% and 94.9%, respectively.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Muneera Altayeb
Department of Communications and Computer Engineering, Faculty of Engineering
Al-Ahliyya Amman University
Amman, Jordan
Email: m.altayeb@ammanu.edu.jo

1. INTRODUCTION

Nowadays, one of the important features of modern cars is their ability to sense the environment and assist drivers in dangerous situations. Traffic signs detection and classification play a very important role in this issue. However, due to the realistic factors affecting image quality, detecting traffic signs efficiently and accurately is still a challenging issue. In addition, most of the algorithms could detect a small number of categories [1], [2]. Many approaches have been discussed in the literature for traffic sign detection based on color and/or shape. Several algorithms have been presented for traffic sign detection and classification based on a set of features boundary distortion, color information, and traffic sign shapes [3]–[6].

Lai *et al*. [7] introduced a model for traffic sign detection that was based on a convolutional neural network (CNN) and support vector machine (SVM) model. The suggested model had an accuracy rate of 98.6%. According to Tabernik and Skocaj [8], a CNN strategy was used. The mistake rate was less than 3% when the suggested method was used to identify 200 different types of traffic signs.

In a study by Chung *et al*. [9], a convolutional pooling neural network is used in place of max pooling to increase recognition accuracy in challenging conditions. The convolutional pooling structure preserves classification performance in the presence of outside noise, which explains why. Taking into account various scenarios, the proposed model is evaluated in accordance with the German traffic sign recognition standard. Conventional CNNs and state-of-the-art CNNs performed 66.981% and 83.198%, respectively, in traffic sign recognition. A deep neural network-based method for classifying traffic signs was presented and evaluated in [10] using the Belgium traffic sign dataset (BTSD). The effectiveness of the suggested method was assessed with various optimizers and activations. SoftMax activation and the adaptive moment estimation optimizer were shown to function well.

On the other hand, Sharma *et al*. [11] introduced a novel framework for real-time traffic sign recognition and classification using dash cams from vehicles, with a refined you only look once version 8 (YOLOv8) model, it achieves a 99.348% mAP50 for detection. The system achieves an F1-score of 98.61% using a two-stage method combining random forest (RF) classification with visual geometry group 16 (VGG16)-based picture embedding. For speed limit sign numerical values, Microsoft's TrOCR is used, and the model is trained on a variety of datasets. A model for real-time traffic sign recognition was proposed by Yang *et al*. [12] utilizing a color histogram of oriented gradients (HOG) and a rapid detection model based on a color probability model. The presented model attained an accuracy of 98.24%. According to Wu *et al*. [13], a real-time traffic sign recognition method was put forward. The region proposal module (RPM) and the classification module (CM) make up the suggested model. The objects are located using the RPM, and the CM is in charge of classifying the objects that are found.

According to Youssouf [14], a CNN model was built to classify 43 different traffic signs. For traffic sign recognition Faster region-based convolutional neural network (R–CNN) and YOLOv4 networks were used. The German traffic sign recognition and detection benchmark datasets were used. The proposed CNN for classification reached an accuracy of 99.20% with only 0.8 M parameters. A new methodology that combined the Haar cascade technique with deep CNN model was proposed in [15] where proposed model achieved an accuracy rate of 98.56%.

The study offers a novel approach for identifying traffic signs that utilize the use of innovative artificial intelligence (AI) capabilities. It presents a technique that builds a strong model for precisely identifying and categorizing traffic signs by utilizing 1,000 unique attributes that are taken from images and the Orange3 data mining platform. The accuracy and effectiveness of traffic sign recognition systems are improved by this work, which is essential for enhancing road safety and traffic control in urban settings. The suggested model works better than current techniques, showing increased accuracy in both detection and classification. Background details on smart road technology, traffic sign identification, and the application of AI in this field are given in the introduction. It also offers a roadmap, outlining the process, machine learning (ML) classifier training, and how these factors affect the model's performance. The model seeks to produce the best results for traffic sign identification and categorization by using this structured method. The accuracy and dependability of the final ML model is provided by the combination of careful feature extraction, a variety of classification strategies, and meticulous validation. To improve the safety and effectiveness of intelligent transportation systems (ITS), this study ultimately aims to provide insightful information about how to improve traffic sign recognition systems. As a result, the paper is an invaluable resource for everyone interested in AI and transportation safety because it emphasizes its importance and contribution to ITS.

2. METHOD

Numerous studies in the literature use image processing and deep learning (DL) techniques to detect and analyze traffic signs. The aim of the current study is to use a new ML model to detect different traffic sign images, as seen in Figure 1. Initially, the image dataset from Kaggle [16] was considered, which included 2,339 images of different traffic signs. It was used to import the Orange3-based classification model. Through the image embedder, 1,000 features were extracted which were fed to the suggested model which consists of four main techniques: adaptive boosting (AdaBoost), decision trees (DT), k-nearest neighbors (KNN), and RF. However, 10-fold cross-validation prevents overfitting. Additionally, the data set was split into two groups. 30% of the dataset is tested, and 70% is training.

To prevent overfitting, a 10-fold cross-validation procedure is employed during model evaluation. Performance metrics including F1-score, accuracy, sensitivity, and precision are used to assess the categorization process. Orange3's user-friendly interface facilitates the development and visualization of the model and allows for the analysis of decision limitations and feature importance. The user-friendly interface enhances the capacity to develop a comprehensive ML model by making it easy to visualize and analyze every stage of the procedure. Figure 1 depicts the classification model using Orange3.

2.1. Dataset

The open Kaggle dataset used in this research consists of 2,339 traffic sign images separated into eight categories: don't go sign, go sign, horn sign, roundabout sign, danger sign, crossing sign, speed limit sign, and unallowed sign. We can see the distribution of these eight categories using the line plot method as shown in Figure 2. This method is one of the techniques used to visualize data profiles, which can optionally show ranges in addition to showing all items on their parallel axes [17]. When displaying numerical data, a line plot shows the data as a collection of points connected by straight lines. Selections of lines, range, mean, group by, error bars, category attributes, and none are all possible. When studying the graph, the user has the option to pan, zoom, and zoom to fit. The human selection of data instances communicates changes automatically, just as a line selection will [18].

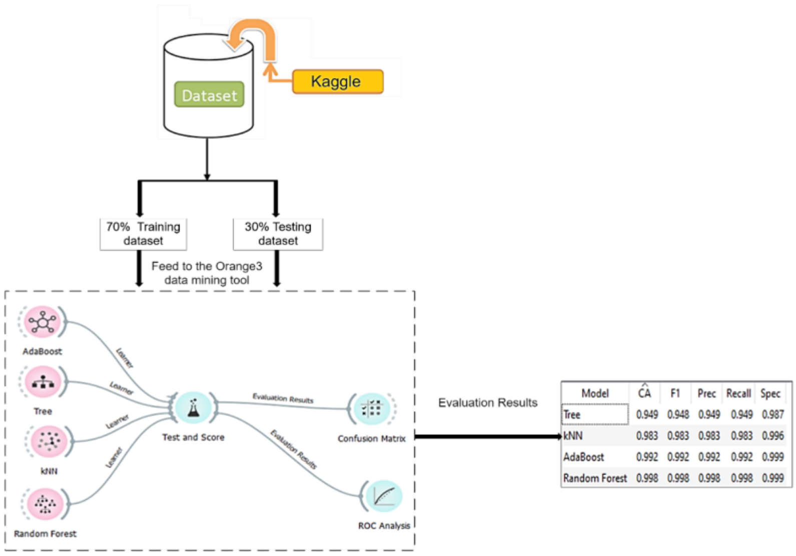


Figure 1. Classification model using Orange3

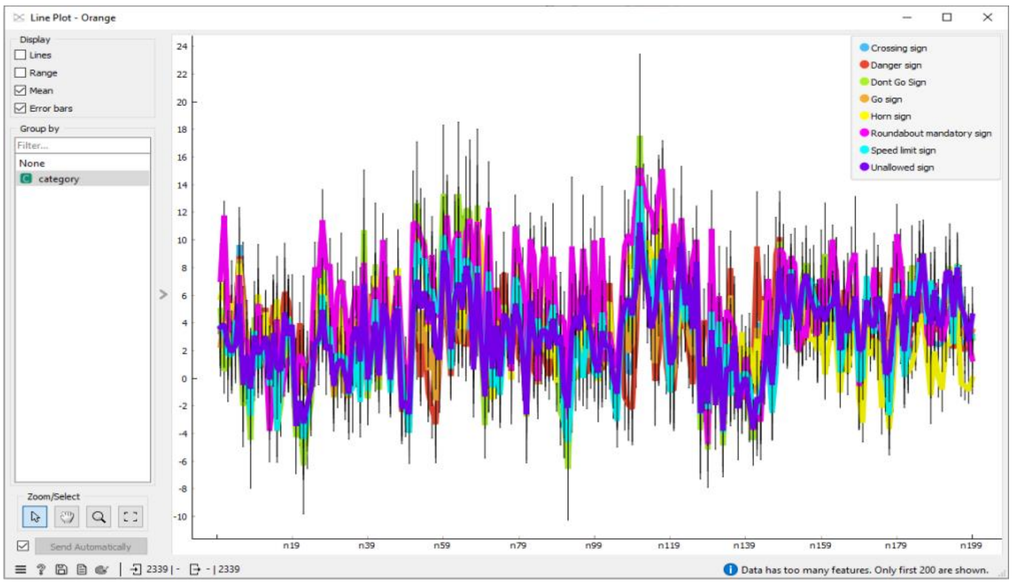


Figure 2. Line plot of eight categories

3. CLASSIFICATION MODEL

Analyzing complicated data as well as detecting patterns requires the use of data mining and ML. In industries like marketing, healthcare, and finance, these technologies are being utilized more and more to process data effectively and make well-informed decisions. Through using the Orange3 data mining application, a study created a classification model and used supervised classification techniques such as AdaBoost, DTs, KNN, and RF for enhancing the performance of weaker classifiers, offering transparency, and managing data with significant dimensions without overfitting. The next sections go over each classification method utilized in this study.

3.1. Adaptive boosting

It is an ML method which Yoav Freund and Robert Schapire suggest. This method employs several classifiers, each of which is trained by dividing the sample according to its output from the preceding classifier. In specific classification tasks, this strategy improves the AdaBoost algorithm's flexibility and resistance to overfitting. Using a voting model in which all of the weak classifiers are joined together, the AdaBoost weak classifier may be made better [19]. In order to improve the model's performance and integrate it with other learning algorithms, AdaBoost is an iterative model that modifies the weight of training samples by creating weak predictors and merging them into a strong predictor [20]. In (1) demonstrates that only the strongest tree will be added to the total each time a new tree model is introduced, eliminating the general tree in the process. The total model performance will progressively increase in this manner as a result of the accumulation of repeated computations.

$$Fn(x) = F_{m-1}(x) + argminh_h \sum_{i=1}^{n} L(y_{i,}\ F_{m-1}(x_i) + h(x_i) \quad (1)$$

Where $h(x_i)$ is the newly inserted tree, $Fn(x)$ is the overall model, $F_{m-1}(x)$ is the overall attained in the previous round, and y_i is the prediction result of the ith tree [21]. The AdaBoost model architecture is shown in Figure 3.

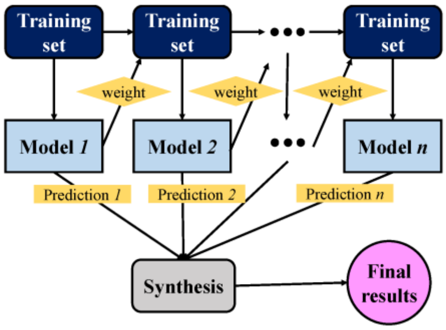


Figure 3. AdaBoost model architecture [21]

3.2. Random forest

As a robust classification and regression technique, the RF algorithm was first presented by Ho in 1995 and then further developed by Breiman in 2001. Utilizing "bagging" or bootstrap aggregating, it creates an ensemble of trees by training each one on two-thirds of the training set and assessing its efficacy. By minimizing model variance, the approach guarantees correctness and stability [22]. This technique builds an ensemble of classifiers by utilizing DT. However, RF is a precision model that assesses each T's distinct error rate for classification by using DT to choose a random sample of data [23]. The trained bootstrap samples of each DT are predicted to include 37% duplicate occurrences, and each DT is trained on a random sample with replacement from the original data. The other randomization technique is attribute sampling. By majority vote, the ultimate forecast is decided. The default configuration of RF exhibits impressive performance and is parameter-free. Creating leaf nodes, requiring a minimum sample size, and adjusting for tree depth are further tuning factors [24]. In order to achieve high accuracy and generalization performance through averaged outcomes using voting or mean-based approaches, the RF method is a non-parametric supervised learning methodology that combines numerous weak classifiers [25]. The number of features used to identify comparable accounts is denoted by N in the equal-representation RF (2), where F_i is the value received from the system and y_i is the original value used for feature i [26].

$$Random\ forest = \frac{1}{N} \sum_{i=1}^{N} (F_i - y_i)^2 \quad (2)$$

3.3. K-nearest neighbors

KNN is a technique that classifies data elements by utilizing multiple nearest neighbors. Because it uses runtime training samples, which constitutes a lazy learning methodology, it is referred to as a memory-based categorization method. Estimating the closest neighbors and determining the class connected to those neighbors are the two primary processes in the KNN process. For efficient classification, this method mostly depends on the training cases [27]. Based on a set's KNN in the training set, KNN predicts the state of the set.

Although it doesn't need any training, it has trouble choosing K values and calculating distances and neighbor matches [28]. Calculating the common Euclidean distance between each cluster center and the point using (3) yields the sum of errors (SSE) [29].

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2 (m_i, x) \tag{3}$$

Here, k is the number of clusters, mi is the cluster's center (mean), distance is the Euclidean distance, and x is a data point that belongs to cluster C_i [30].

3.4. Decision tree

DTs are predictive models that use a hierarchical tree structure. By branching the traits, they predict the desired values. When dealing with finite-set target parameters, classification trees are chosen, whereas regression trees are employed for continuous values [31]. Using a top-down, recursive process, DT build classification rules from irregular data, making them an essential tool for classification and prediction. At leaf nodes, they derive conclusions based on a comparison of attribute values in inner nodes. When the training case is described in attributes and conclusions, DTs which are crucial to data analysis can be utilized successfully even in the absence of a deep understanding of this topic [32], [33]. However, recursively splitting the training data set's feature space at branch nodes and assigning prediction labels at leaf nodes results in a binary branching structure, or DTs model, with internal and leaf nodes as shown in Figure 4. The ability to describe the formulation in multiple formats represents a challenge when building optimal DTs [34].

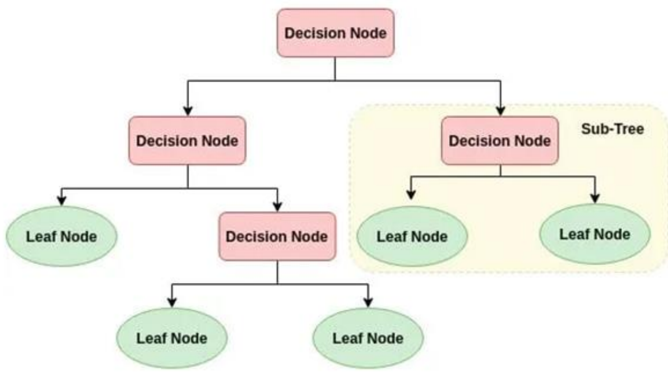


Figure 4. Decision tree architecture [35]

4. PERFORMANCE EVALUATION

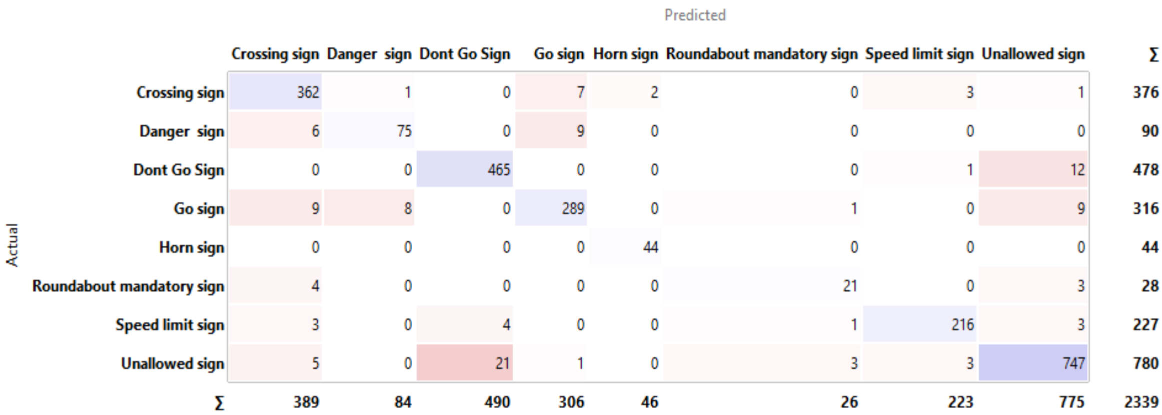
Assessing the predicted efficacy of the suggested model is crucial after testing and validating the main model assumptions. Consequently, evaluation metrics were employed to assess the proposed model's suitability. The confusion matrix is a helpful tool for assessing how well prediction and classification systems are performing. In accuracy rate computations, it establishes the volumes of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) [36]. Accuracy, precision, sensitivity, and specificity were utilized to evaluate the efficiency of the proposed model, as shown in Table 1 [37], and the F-measure was employed in this investigation.

Table 1. Performance matrices equation [37]

Performance metric	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F-measure	$\frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}$
Specificity	$\frac{TN}{TN+FP}$

4.1. Confusion matrix

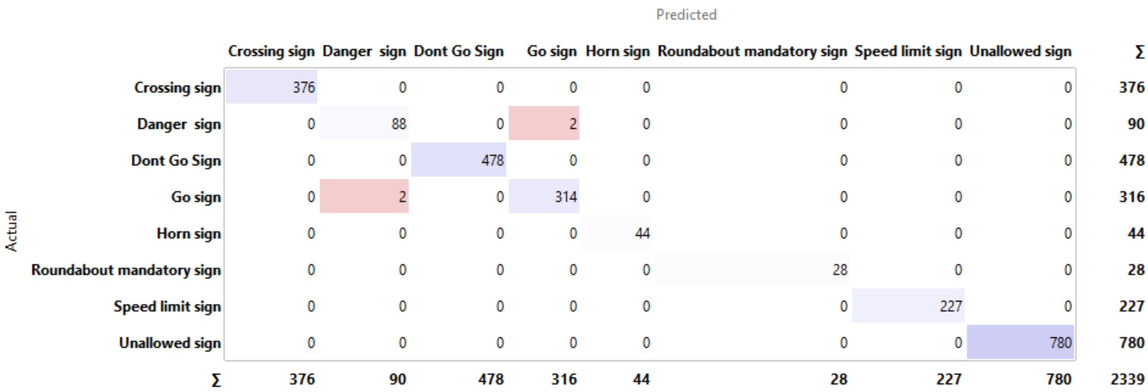
AdaBoost, DTs, KNN, and RF were among the algorithms utilized in this study to classify traffic indicators. A confusion matrix is used to evaluate each algorithm's performance as shown in Figures 5 to 8; which Figure 5 displays the DT confusion matrix; Figure 6 displays the RF confusion matrix; Figure 7 displays the AdaBoost confusion matrix; and Figure 8 displays the KNN confusion matrix. The observation from these results is that it exceeds RF with a classification accuracy of 99.8%. For AdaBoost, the classifier accuracy achieved 99.2%, while the classification accuracy of DTs and KNN was 98.3% and 94.9%, respectively.



Predicted

Actual	Crossing sign	Danger sign	Dont Go Sign	Go sign	Horn sign	Roundabout mandatory sign	Speed limit sign	Unallowed sign	Σ
Crossing sign	362	1	0	7	2	0	3	1	376
Danger sign	6	75	0	9	0	0	0	0	90
Dont Go Sign	0	0	465	0	0	0	1	12	478
Go sign	9	8	0	289	0	1	0	9	316
Horn sign	0	0	0	0	44	0	0	0	44
Roundabout mandatory sign	4	0	0	0	0	21	0	3	28
Speed limit sign	3	0	4	0	0	1	216	3	227
Unallowed sign	5	0	21	1	0	3	3	747	780
Σ	389	84	490	306	46	26	223	775	2339

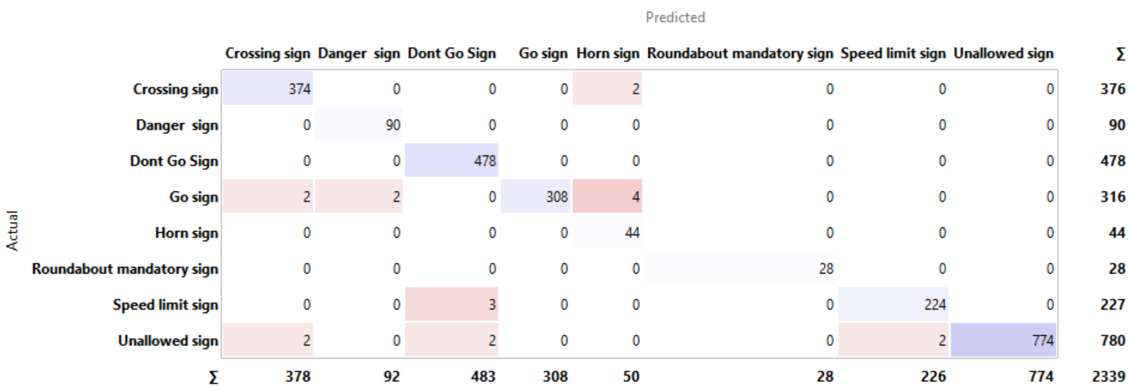
Figure 5. Confusion matrices of the evaluated classifiers of DT



Predicted

Actual	Crossing sign	Danger sign	Dont Go Sign	Go sign	Horn sign	Roundabout mandatory sign	Speed limit sign	Unallowed sign	Σ
Crossing sign	376	0	0	0	0	0	0	0	376
Danger sign	0	88	0	2	0	0	0	0	90
Dont Go Sign	0	0	478	0	0	0	0	0	478
Go sign	0	2	0	314	0	0	0	0	316
Horn sign	0	0	0	0	44	0	0	0	44
Roundabout mandatory sign	0	0	0	0	0	28	0	0	28
Speed limit sign	0	0	0	0	0	0	227	0	227
Unallowed sign	0	0	0	0	0	0	0	780	780
Σ	376	90	478	316	44	28	227	780	2339

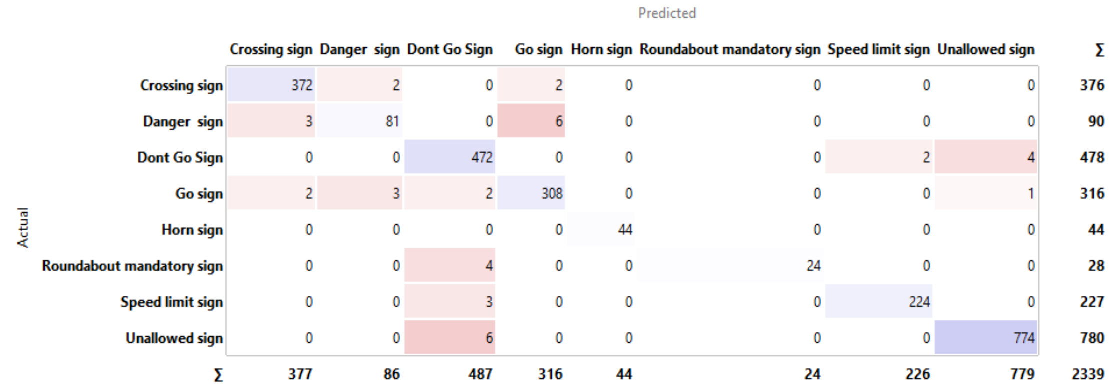
Figure 6. Confusion matrices of the evaluated classifiers of RF



Predicted

Actual	Crossing sign	Danger sign	Dont Go Sign	Go sign	Horn sign	Roundabout mandatory sign	Speed limit sign	Unallowed sign	Σ
Crossing sign	374	0	0	0	2	0	0	0	376
Danger sign	0	90	0	0	0	0	0	0	90
Dont Go Sign	0	0	478	0	0	0	0	0	478
Go sign	2	2	0	308	4	0	0	0	316
Horn sign	0	0	0	0	44	0	0	0	44
Roundabout mandatory sign	0	0	0	0	0	28	0	0	28
Speed limit sign	0	0	3	0	0	0	224	0	227
Unallowed sign	2	0	2	0	0	0	2	774	780
Σ	378	92	483	308	50	28	226	774	2339

Figure 7. Confusion matrices of the evaluated classifiers of AdaBoost



Actual \ Predicted	Crossing sign	Danger sign	Dont Go Sign	Go sign	Horn sign	Roundabout mandatory sign	Speed limit sign	Unallowed sign	Σ
Crossing sign	372	2	0	2	0	0	0	0	376
Danger sign	3	81	0	6	0	0	0	0	90
Dont Go Sign	0	0	472	0	0	0	2	4	478
Go sign	2	3	2	308	0	0	0	1	316
Horn sign	0	0	0	0	44	0	0	0	44
Roundabout mandatory sign	0	0	4	0	0	24	0	0	28
Speed limit sign	0	0	3	0	0	0	224	0	227
Unallowed sign	0	0	6	0	0	0	0	774	780
Σ	377	86	487	316	44	24	226	779	2339

Figure 8. Confusion matrices of the evaluated classifiers of KNN

However, both the RF and AdaBoost approaches perform significantly better, as evidenced by the classification results in Table 2, where accuracy values reach 99.80% and 99.20%, respectively. Additionally, the KNN approach performs admirably, scoring 98.30%. The findings demonstrated that, with 99.80% precision and recall, the RF approach was the most effective. Also, RF and AdaBoost, reach the largest specificity value of 99.90%, while DTs achieve the lowest specificity and F1-score value of 98.70% and 94.80% respectively. However, Table 3 provides the results of comparing the classifier's performance assessment of the proposed model with previous studies.

Table 2. Classification results

Model	AUC (%)	Accuracy (%)	F1-score (%)	Precision (%)	Recall (%)	Specificity (%)
DT	98.90	94.90	94.80	94.90	94.90	98.70
KNN	100.00	98.30	98.30	98.30	98.30	99.60
AdaBoost	99.50	99.20	99.20	99.20	99.20	99.90
RF	100.00	99.80	99.80	99.80	99.80	99.90

Table 3. Comparison of the classifier's performance assessment from earlier research and suggested model

Study/Year	Methods/model	Performance metrics	Percentage (%)
[7]/2023	CNN	Accuracy	98.6
[9]/2022	CNN	Accuracy	83.198
[11]/2024	CNN, YOLO v8	Accuracy	98.61
[14]/2024	CNN, YOLOv4	Accuracy	99.2
[15]/2024	CNN	Accuracy	98.56
Proposed	AdaBoost, DT, KNN, and RF	Accuracy	99.80

4.2. Receiver operating characteristic analysis

The receiver operating characteristic (ROC) curve is a graph in which a false positive rate (FPR) or 1-specificity is plotted on the x-axis and a true positive rate (TPR) or sensitivity is displayed on the y-axis. Classifiers with ROC curves toward the upper left corner are higher performers. Classifiers or forecast accuracy can be compared using the area under the ROC curve [38]–[40]. In addition to its significance when assessing binary predictors, it draws attention to its characteristics, interpretation, and utility when comparing other tests or predictor variables [41]–[44].

Figure 9 shows the ROC curve is also used in this study to evaluate the performance of the traffic sign prediction model. Figure 9(a) displays the ROC analysis of the AdaBoost classifier, Figure 9(b) displays the ROC analysis of the RF classifier, Figure 9(c) displays the ROC analysis of the DT classifier, and Figure 9(d) displays the ROC analysis of the KNN classifier. Figure 9 shows that, based on the ROC curve data, the AUC values for DT (98.90%), KNN (100.00%), AdaBoost (99.50%), and RF (100.00%) show outstanding performance across the four classification models. When it comes to effective class separation, KNN and RF both have perfect AUC values. All models, especially the ensemble methods that improve accuracy like RF and AdaBoost, are robust, as seen by the high AUC values. It is crucial to validate the models' generalization abilities using additional validation processes such as cross-validation, error analysis, and classification threshold tuning in order to guarantee practical use.

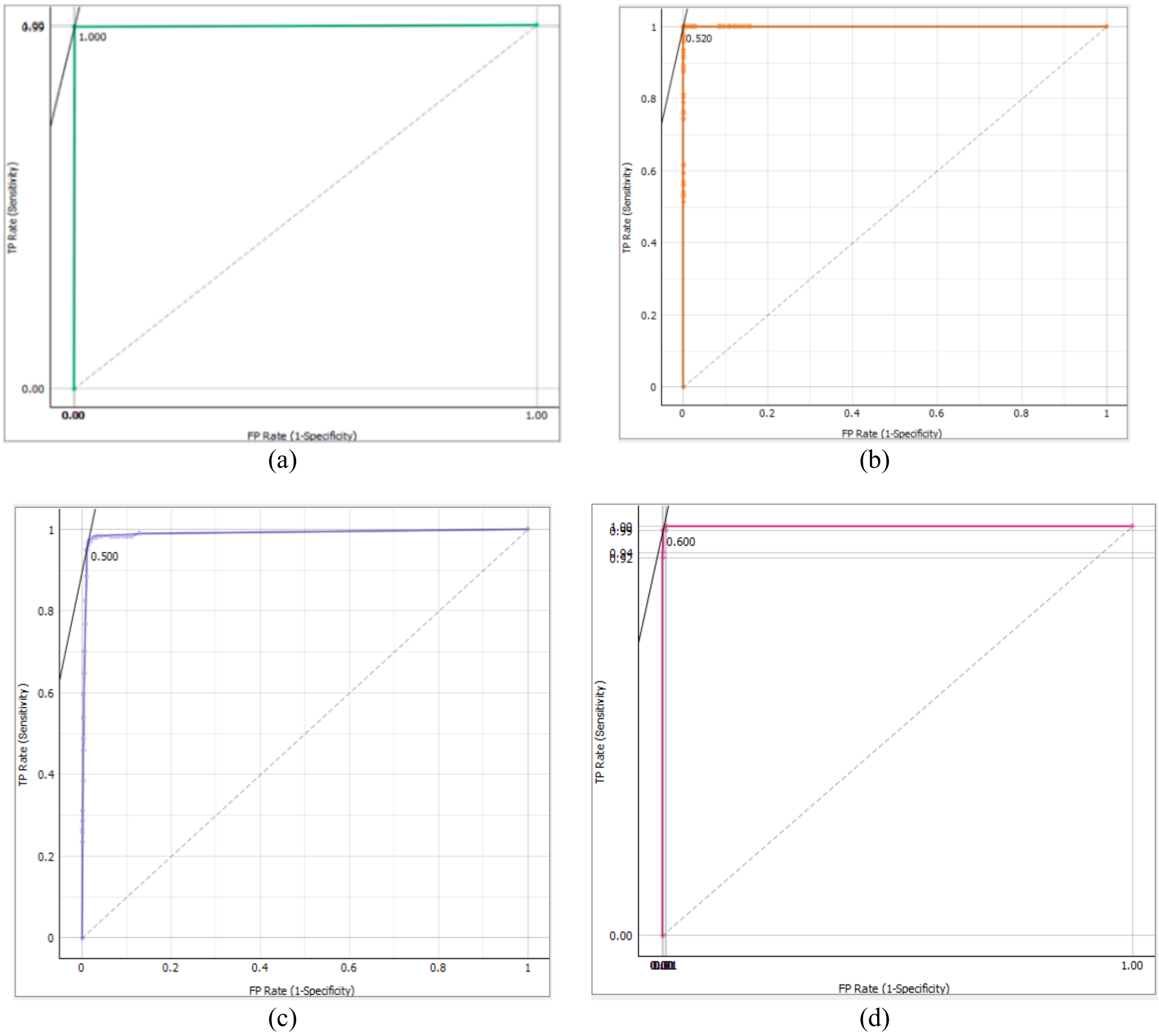


Figure 9. ROC curves for the evaluated classifiers of (a) AdaBoost, (b) RF, (c) DT, and (d) KNN

ROC curve and AUC are useful tools for evaluating binary classifier performance. These facilitate informed decision-making when selecting or enhancing models, as well as understanding how well a model may differentiate across classes. It can enhance one's capacity to assess and examine model performance. While the ROC curve shows how well the classifier performs at various levels, the AUC often highlights its performance. A higher AUC indicates a more successful classifier. The AUC is calculated using the ROC curve graphing process. AUC values of 0.85 or more indicate that the model performs well in distinguishing between traffic signs. Using ROC curve analysis, Orange3 simplifies the evaluation of binary classifiers, facilitating rapid experimentation, informed decision-making, and graphical representation. This improves classification opportunities and maximizes data science projects.

5. CONCLUSION

In this research, four ML models are compared in the study RF, AdaBoost, KNN, and DT. DT's accuracy percentage is 94.90%. In contrast, KNN offers comparable metrics and an accuracy percentage of 98.30%. While KNN works well for classification applications, depending on distance calculations can raise the cost of computing and make it less scalable for larger datasets. On the other side, AdaBoost's ability to modify weights on cases that were incorrectly classified allowed it to attain an accuracy of 99.20%. It is a strong option for classification problems since it controls FP and FN well. With an accuracy of 99.80%, RF emerged as the best performer, demonstrating its dependability and capacity for generalization. The findings highlight the relationship between interpretability and predictive accuracy by demonstrating a positive correlation between higher model complexity and improved performance measures. The best model for applications involving great accuracy is RF. Parameter optimization and cross-validation approaches should be investigated in future studies to improve the models' generalizability over a larger variety of datasets.

6. FUTURE WORK

To enhance model performance and applicability, future traffic sign recognition research might focus on incorporating DL methods, especially CNNs. Autonomous cars require real-time processing methods, and combining different datasets can improve model performance and generalization. To guarantee dependability and security, it is essential to look at how resilient recognition systems are to hostile attacks and changes in sign design. However, enhancing detection skills under a variety of circumstances can be achieved by investigating multi-modal systems that use many sensors, user-centric design, and continuous learning systems that adjust to modifications in traffic sign designs or laws. Cross-domain adaptation techniques can help make these systems more broadly applicable in other countries. Also, autonomous vehicles that incorporate traffic sign recognition technologies can greatly improve user experience, safety, and operational efficacy, resulting in more dependable and successful ITS. Creating explainability strategies can improve user comprehension and trust, especially in applications that are safety-critical. Working together, industry and academia may encourage the application of cutting-edge recognition technology in practical settings.

7. LIMITATION

The difficulties that traffic sign recognition systems encounter include changes in the surroundings, background complexity, processing speed, attack vulnerability, integration problems, user involvement, and ethical and legal considerations. Environmental elements that raise FP or FN include poor clarity, a variety of sign designs, and intricate backgrounds. For real-time recognition, processing speed is essential, but overfitting might result in delays. By addressing these constraints, further study can improve the efficiency of systems of traffic sign recognition.

ACKNOWLEDGMENTS

We are especially grateful to Al Ahliyya Amman University for their support with this study.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Areen Arabiat	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	
Muneera Altayeb	✓		✓	✓			✓			✓	✓	✓	✓	
Sanaa Salama						✓	✓		✓		✓			

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The dataset that supports the findings of this study is openly available in Kaggle at https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification, reference [16].

REFERENCES

[1] V. N. Sichkar and S. A. Kolyubin, “Real time detection and classification of traffic signs based on YOLO version 3 algorithm,” *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, vol. 20, no. 3, pp. 418–424, 2020, doi: 10.17586/2226-1494-2020-20-3-418-424.

[2] C. Liu, S. Li, F. Chang, and Y. Wang, “Machine vision based traffic sign detection methods: review, analyses and perspectives,” *IEEE Access*, vol. 7, pp. 86578–86596, 2019, doi: 10.1109/ACCESS.2019.2924947.

[3] Y. Zeng, J. Lan, B. Ran, Q. Wang, and J. Gao, “Restoration of motion-blurred image based on border deformation detection: a traffic sign restoration model,” *PLoS ONE*, vol. 10, no. 4, 2015, doi: 10.1371/journal.pone.0120885.

[4] A. H. Alomari and T. I. Alhadidi, “A stochastic approach for modeling the saturation flow rate at traffic signals in Jordan,” *Iranian Journal of Science and Technology - Transactions of Civil Engineering*, vol. 48, no. 5, pp. 3775–3786, 2024, doi: 10.1007/s40996-024-01441-6.

[5] R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, and N. Paparoditis, “Road sign detection in images: a case study,” in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 484–488, doi: 10.1109/ICPR.2010.1125.

[6] R. Al-Hindawi, T. I. Alhadidi T, and M. Adas, “Evaluation and optimization of adaptive cruise control in autonomous vehicles using the CARLA simulator: a study on performance under wet and dry weather conditions,” in *2024 IEEE International Conference on Advanced Systems and Emergent Technologies*, 2024, pp. 1–6, doi: 10.1109/IC_ASET61847.2024.10596222.

[7] Y. Lai, N. Wang, Y. Yang, and L. Lin, “Traffic signs recognition and classification based on deep feature learning,” in *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, 2018, pp. 622–629, doi: 10.5220/0006718806220629.

[8] D. Tabernik and D. Skocaj, “Deep learning for large-scale traffic-sign detection and recognition,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1427–1440, 2020, doi: 10.1109/TITS.2019.2913588.

[9] J. H. Chung, D. W. Kim, T. K. Kang, and M. T. Lim, “Traffic sign recognition in harsh environment using attention based convolutional pooling neural network,” *Neural Processing Letters*, vol. 51, no. 3, pp. 2551–2573, 2020, doi: 10.1007/s11063-020-10211-0.

[10] S. Mehta, C. Paunwala, and B. Vaidya, “CNN based traffic sign classification using adam optimizer,” in *2019 International Conference on Intelligent Computing and Control Systems*, 2019, pp. 1293–1298, doi: 10.1109/ICCS45141.2019.9065537.

[11] V. Sharma, V. Kumar, and H. Aditya, “Traffic sign detection and classification,” in *2023 IEEE International Conference on Computer Vision and Machine Intelligence*, 2023, pp. 1–6, doi: 10.1109/CVMI59935.2023.10464876.

[12] Y. Yang, H. Luo, H. Xu, and F. Wu, “Towards real-time traffic sign detection and classification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2022–2031, 2016, doi: 10.1109/TITS.2015.2482461.

[13] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, “Real-time traffic sign detection and classification towards real traffic scene,” *Multimedia Tools and Applications*, vol. 79, no. 25–26, pp. 18201–18219, 2020, doi: 10.1007/s11042-020-08722-y.

[14] N. Youssouf, “Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4,” *Heliyon*, vol. 8, no. 12, 2022, doi: 10.1016/j.heliyon.2022.e11792.

[15] N. Triki, M. Karray, and M. Ksantini, “A real-time traffic sign recognition method using a new attention-based deep convolutional neural network for smart vehicles,” *Applied Sciences*, vol. 13, no. 8, 2023, doi: 10.3390/app13084793.

[16] A. V. N. M. Hemateja, “Traffic sign dataset - classification,” *Kaggle*. 2022. Accessed: Sep. 05, 2024. [Online]. Available: https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification

[17] Z. Dobesova, “Evaluation of Orange data mining software and examples for lecturing machine learning tasks in geoinformatics,” *Computer Applications in Engineering Education*, vol. 32, no. 4, 2024, doi: 10.1002/cae.22735.

[18] “Line plot — orange visual programming 3 documentation,” *Orange Data Mining*. 2015. Accessed: Sep. 01, 2024. [Online]. Available: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/evaluate/predictions.html

[19] J. K. Tsai and C. H. Hung, “Improving AdaBoost classifier to predict enterprise performance after COVID-19,” *Mathematics*, vol. 9, no. 18, 2021, doi: 10.3390/math9182215.

[20] G. An *et al.*, “Short-term wind power prediction based on particle swarm optimization-extreme learning machine model combined with AdaBoost algorithm,” *IEEE Access*, vol. 9, pp. 94040–94052, 2021, doi: 10.1109/ACCESS.2021.3093646.

[21] C. Wang, S. Xu, and J. Yang, “AdaBoost algorithm in artificial intelligence for optimizing the IRI prediction accuracy of asphalt concrete pavement,” *Sensors*, vol. 21, no. 17, 2021, doi: 10.3390/s21175682.

[22] C. Bentéjac, A. Csörgő, and G. M.-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, 2021, doi: 10.1007/s10462-020-09896-5.

[23] Z. Shao, M. N. Ahmad, and A. Javed, “Comparison of random forest and XGBoost classifiers using integrated optical and SAR features for mapping urban impervious surface,” *Remote Sensing*, vol. 16, no. 4, 2024, doi: 10.3390/rs16040665.

[24] A. Lachaud, M. Adam, and I. Mišković, “Comparative study of random forest and support vector machine algorithms in mineral prospectivity mapping with limited training data,” *Minerals*, vol. 13, no. 8, 2023, doi: 10.3390/min13081073.

[25] J. Wang, S. Ma, P. Jiao, L. Ji, X. Sun, and H. Lu, “Analyzing the risk factors of traffic accident severity using a combination of random forest and association rules,” *Applied Sciences*, vol. 13, no. 14, 2023, doi: 10.3390/app13148559.

[26] M. A. Alhariri, “Early detection of similar fake accounts on twitter using the random forest algorithm,” *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 12, pp. 611–620, 2020, doi: 10.34218/IJARET.11.12.2020.064.

[27] Y. Narayan, “SEMG signal classification using KNN classifier with FD and TFD features,” *Materials Today: Proceedings*, vol. 37, no. Part 2, pp. 3219–3225, 2020, doi: 10.1016/j.matpr.2020.09.089.

[28] A. Juwaied, L. J.-Strumillo, and A. Sierszeń, “Enhancing clustering efficiency in heterogeneous wireless sensor network protocols using the k-nearest neighbours algorithm,” *Sensors*, vol. 25, no. 4, 2025, doi: 10.3390/s25041029.

[29] T. Oberski, B. Walendzik, and M. Szejnfeld, “The monitoring of macroplastic waste in selected environment with UAV and multispectral imaging,” *Sustainability*, vol. 17, no. 5, 2025, doi: 10.3390/su17051997.

[30] C. Karras, A. Karras, and S. Sioutas, “Pattern recognition and event detection on IoT data-streams,” *arXiv-Computer Science*, pp. 1–24, 2022.

[31] K. Alnowaiser, “Improving healthcare prediction of diabetic patients using KNN imputed features and tri-ensemble model,” *IEEE Access*, vol. 12, pp. 16783–16793, 2024, doi: 10.1109/ACCESS.2024.3359760.

[32] Q. Xin, R. Song, Z. Wang, Z. Xu, and F. Zhao, “Enhancing bank credit risk management using the C5.0 decision tree algorithm,” *Journal of Computer Technology and Applied Mathematics*, vol. 1, no. 4, pp. 100–107, 2024, doi: 10.5281/zenodo.14032041.

[33] L. Tensen and K. Fischer, “Evaluating hybrid speciation and swamping in wild carnivores with a decision-tree approach,” *Conservation Biology*, vol. 38, no. 1, 2024, doi: 10.1111/cobi.14197.

[34] E. Liu, T. Hu, T. T. Allen, and C. Hermes, “Optimal classification trees with leaf-branch and binary constraints,” *Computers and Operations Research*, vol. 166, 2024, doi: 10.1016/j.cor.2024.106629.

[35] S. M. C. Cindra, "Machine learning with Python for beginner: DQLab career track module," *Medium*. 2020. [Online]. Available: https://michelle-cindra1998.medium.com/machine-learning-with-python-for-beginner-adec066f187c

[36] S. Nafaa *et al.*, "Advancing roadway sign detection with YOLO models and transfer learning," in *2024 IEEE 3rd International Conference on Computing and Machine Intelligence*, 2024, pp. 1–4, doi: 10.1109/ICMI60790.2024.10586105.

[37] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of k-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-10358-x.

[38] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315–1316, 2010, doi: 10.1097/JTO.0b013e3181ec173d.

[39] H. E. Moore, O. Andlauer, N. Simon, and E. Mignot, "Exploring medical diagnostic performance using interactive, multi-parameter sourced receiver operating characteristic scatter plots," *Computers in Biology and Medicine*, vol. 47, no. 1, pp. 120–129, 2014, doi: 10.1016/j.compbiomed.2014.01.012.

[40] Q. Yun, W. B. Han, Y. Y. Guo, H. Wang, and M. Du, "The detection, extraction and parameter estimation of extreme-mass-ratio inspirals with deep learning," *Science China: Physics, Mechanics and Astronomy*, vol. 68, no. 1, 2025, doi: 10.1007/s11433-024-2500-x.

[41] F. Movahedi, R. Padman, and J. F. Antaki, "Limitations of receiver operating characteristic curve on imbalanced data: assist device mortality risk scores," *Journal of Thoracic and Cardiovascular Surgery*, vol. 165, no. 4, pp. 1433-1442.e2, 2023, doi: 10.1016/j.jtcvs.2021.07.041.

[42] N. Alzboun, M. Alhur, H. Khawaldah, and M. T. Alshurideh, "Assessing gastronomic tourism using machine learning approach: The case of google review," *International Journal of Data and Network Science*, vol. 7, no. 3, pp. 1131–1142, 2023, doi: 10.5267/j.ijdns.2023.5.010.

[43] J. Muschelli, "ROC and AUC with a binary predictor: a potentially misleading metric," *Journal of Classification*, vol. 37, no. 3, pp. 696–708, 2020, doi: 10.1007/s00357-019-09345-1.

[44] M. Hassanzad and K. H. -Tilaki, "Methods of determining optimal cut-point of diagnostic biomarkers with application of clinical data in ROC analysis: an update review," *BMC Medical Research Methodology*, vol. 24, no. 1, 2024, doi: 10.1186/s12874-024-02198-2.

BIOGRAPHIES OF AUTHORS

Areen Arabiat obtained B.Sc. in Computer Engineering in 2005 from Al Balqa Applied University, and her M.Sc. in Intelligent Transportation Systems from Al-Ahliyya Amman University in 2022. She is currently a part-time lecturer, researcher and computer lab supervisor at the Faculty of Engineering/Al-Ahliyya Amman University since 2013. Her research interests are focused on the following areas: machine learning, data mining, artificial intelligence, IoT, cyber security, and image processing. She can be contacted at email: a.arabiat@ammanu.edu.jo.

Muneera Altayeb obtained a bachelor's degree in Computer Engineering in 2007, and a master's degree in Communications Engineering from the University of Jordan in 2010. She has been working as a lecturer in the Department of Communications and Computer Engineering at Al-Ahliyya Amman University since 2015. She currently holds the position of Assistant Dean at the Faculty of Engineering at Amman Al-Ahliyya University. Her research interests focus on the following areas: digital signals and image processing, machine learning, robotics, and artificial intelligence. She can be contacted at email: m.altayeb@ammanu.edu.jo.

Dr. Sanaa Salama is an Associate Professor at the Department of Biomedical Engineering at Arab American University. She received the B.Sc. degree in Telecommunication Technology from the Arab American University, Palestine, in 2006 (excellent evaluation GPA 3.96/4), the M.Sc. degree in Electrical Engineering from the University of Jordan, Jordan, in 2009 (excellent evaluation GPA 3.83/4), and she received the Ph.D. degree in Electrical Engineering from Duisburg-Essen University, Germany, in 2015. Her research interests are characteristic chassis wavemodes, MIMO antenna design and beam-forming antenna array design, reconfigurable antenna, coupling-element based antenna structures, mutual coupling, chassis wavemodes coupling, ports and patterns isolation, her recent field of research is design of RF coils for 7T MRI system, and design of decoupling and matching networks. She can be contacted at email: sanaa.salama@aaup.edu.