Network Slicing in CBRS System using Non-Linear Classification

AHMED N. H. ALNUAIMY, IHAM TARIQ ISMAIL AL-NUAIMI
Department of Mechatronics Engineering,
Alkhawarizmi College of Engineering,
University of Baghdad,
Baghdad 10013,
IRAQ

Abstract: - The Citizens Broadband Radio Service (CBRS) system requires an optimal network slicing approach to allocate resources and manage quality of service (QoS) for various applications. This paper focuses on refinement of network slicing through non-linear classification methods: Decision Trees, Random Forests, and K-Nearest Neighbours (KNN). We study how these algorithms meet accuracy and processing time benchmarks best suited for dynamic spectrum access situations. With the aid of machine learning, we build a model that predicts channel conditions and classifies network slices in real-time based on traffic flows. The results indicate that Random Forest achieved the highest accuracy but Decision Trees provided a favourable balance between computation and precision. KNN performed better than many of the other algorithms in terms of accuracy but cost more time in processing. The balance between best accuracy and lowest computational time highlights classification methods that could be implemented for real-time CBRS network slicing. Consequently, the proposals made enhance spectral efficiency while reducing latency and improving overall performance across the network.

Key-Words: - CBRS, Network Slicing, KNN, Decision Tree, Random Forest.

Received: December 28, 2025. Revised: March 6, 2026. Accepted: May 7, 2026. Published: July 6, 2026.

1 Introduction

The growth of wireless communication has been very fast over the last decades, and one thing that is becoming very promising is CBRS in enhancing spectrum utilization efficiency for a wide range of applications. Other factors that play in are dynamic spectrum availability and heterogeneous user requirements, which impose more constraints on channel allocation within the CBRS scheme, making it really challenging, [1], [2]. The previous resource management procedures, whether linear or simplistic, did not do justice to the complexities of spectrum management; hence, there emerged a very suboptimal performance scenario with an unfairness streak that practically always presented itself to the users, [3], [4].

Hence, in improving the old model of resource optimization with a new one that stresses network slicing, we suggest a solution grounded on non-linear classification. The main objective here is to optimize resource allocation and ensure fair access for every user. Through classification and making predictions, machine learning methods will help in forecasting the channel conditions among multiple network slices based on up-to-date traffic patterns. This study tries to close the gap of smart spectrum administration in CBRS for more adaptable and resilient wireless networks

2 Citizen Broadband Radio Services

CBRS constitutes an agile spectrum-sharing approach devised by the Federal Communications Commission (FCC) to foster the activities of wireless communications through the more efficient use of a 3.5 GHz (3550-3700 MHz) band of frequencies. In this manner, all types of users-from mobile network operators to enterprises, to private networks will have dynamic access to the spectrum while ensuring interference protection for incumbent users such as U.S. Navy radar systems and fixed satellite service applications, [1].

CBRS offers a three-tiered access framework which consists of the Incumbent Access, Priority Access (PA) and the General Authorized Access (GAA) [2]. On the user side, spectrum is provided to the users by 3 methods. In-network users (currently in the network together with the priorities) are the highest priority users experiencing interference-free availability. PA licensees can obtain the right to use the frequency via auction, and GAA-authorized users are entitled to access with opportunist privilege in case the frequency is not being used by a higher tier user. This hierarchical resource is managed by a Spectrum Access System (SAS) which schedules the spectrum resources in order to provide fair and efficient use of the resources.

The CBRS has shifted the landscape of wireless, now making it feasible to deploy private 5G networks, industrial IoT applications, and deliver better broadband service. These many operators and application in the same spectrum band, naturally enables the new concept of network slicing, wherein virtualized network instances can be customized for applications with specific requirements. However, its use to guide smart decision making is necessary to achieve efficient spectrum allocation and network slicing. In this context, complex machine learning strategies are needed to optimize performance, minimal interference and latency, [2].

This paper examines the various non-linear classifiers namely Decision Trees, Random Forests and KNN for the purpose of improving network slicing in the context of CBRS. These algorithms will be tested in classification accuracy and processing time— And, proceed with the final selection of the most adequate integration method for Real-Time CBRS Resource Management. The study describes the trade-off in terms of accuracy against computation efficiency thereby contributing towards intelligent CBRS-based solutions for network slicing strategies.

3 Application of Classification

In this paper, classification refers to the cognitive discernment of allocating network slicing requests into relevant network slices depending on their service needs. In the context of CBRS (Citizens Broadband Radio Service), users and services may have differences in their bandwidth, latency, reliability, and priority needs. With a classification model based on non-linear Random Forest or Decision Trees or even K-Nearest Neighbors, these input features are processed, instantiating the most applicable slice for fulfilling the given request. This increases overall resource allocation efficiency and ensures that each service is paired with a slice that is guaranteed to fulfil its qualitative service level requirements.

4 Network Slicing and Machine Learning Techniques

Within the CBRS scheme, network slicing forms a practical and effective means of resource allocation since multiple virtualized networks can be implemented over the same physical infrastructure. Much more than through software-defined networking (SDN) and network function virtualization (NFV) [3], such a scheme allows for meeting special service provisioning created specifically for applications with user needs: low latency for industrial automation, high throughput in the mobile broadband context, or huge connectivity in IoT applications, [2], [3]. Again, in the CBRS ecosystem, different priority tiers (for instance: Incumbent Access, Priority Access (PA), and General Authorized Access (GAA)) can coexist harmoniously while having their distinct QoS requirements through slicing.

Dynamic availability of spectrum poses a major challenge in CBRS network slicing, [4]. Compared to licensed spectrum, wherein the spectrum resources get accessed dynamically according to availability and interference conditions, CBRS operates under the sharing access model, [1], [2], [3]. Hence, effective spectrum management is indispensable since this entails implementing proper intelligent classification techniques followed by resource allocations in assigning slices optimally. SAS is abstracting such centralized coordination of spectrum assignment, however, integration of other decision-making techniques through advanced machine learning can further increase network efficiencies in traffic demand prediction, slice allocation optimization, and latency reduction, [2].

Employing algorithms of classification in a non-linear framework such as Decision Trees, Random Forests, and KNN will help to handle user demand classification and enable resource allocation within the scope of a manageable network slicing in CBRS. Furthermore, these machine learning models will be able to leverage real-time parameters of network traffic, user mobility, and interference levels to configuration slices. While Random Forest exploits the property of ensemble learning and achieves high accuracy, both Decision Trees are faster in making decisions because of their non-complex computation, and KNN can work in adapting to changing network conditions but it might cost higher processing time, [2].

The application of AI-enabled classification techniques for CBRS network slicing can significantly enhance resource utilization, service differentiation, and QoS enforcement. By intelligently mapping user requirements to suitable slices, operators could achieve seamless connectivity while minimizing spectral inefficiencies, [1], [2], [3], [4]. Moreover, as CBRS deployments continue to ramp up, the incorporation of advanced classification models would be central to maintaining network speed and the diversity of such applications as 5G private networks, smart cities, industrial IoT, and emergency communications, [5].

5 Non-Linear Classification

Non-linear classification entails the categorization or labelling of samples when the information itself—the decision boundary between classes—cannot be represented by either a straight line or a simple linear function, [5]. The general idea is that linear classifiers assume some sort of separation between input samples via a hyperplane while the non-linear ones could understand complex relationships hidden in the data, making them applicable to real-world applications like network slicing in CBRS, where spectrum availability, user demand, and interference patterns are inherently non-linear. These classifiers use methods such as decision trees, kernel-based methods, and ensemble learning to discover non-trivial decision boundaries, [6].

In CBRS network slicing, the proper selection of the right non-linear classification technique, notably Decision Trees, Random Forests, and KNN is dependent on the trade-offs between accuracy, processing time, and scalability. Decision Trees require fast classification with interpretable rules, Random Forest balances accuracy and robustness, and KNN accommodates adaptivity, thus requiring higher computation effort, [7], [8], [9].

A decision tree is an example of a simple non-linear classifier, which is a series of feature-based conditions used in decision-making. The DT is constructed of internal nodes that represent features or tests, edges that represent values for each feature, and leaf nodes that indicate the class labels, [9]. In general, DT's idea would split the instance set recursively based on feature values that can best separate the target classes. Metrics like Gini index, entropy, or information gain often quantify the goodness of such splits, [10].

Equation (1) defines Gini impurity, a criterion that is generally used for judging dissection within a tree decision. The definition of Gini impurity for a fixed node is:

$$Gini(t) = 1 - \sum_{i=1}^{k} p_i^2 \qquad (1)$$

where:
p_i is the probability of an element into a set of symbols.
k is the number of classes.

The purpose of decision trees is to minimize Gini's impurity at every node to achieve the maximum possible homogeneity in class distribution for subsets, [10]. The one that contributes the least to the Gini impurity will be selected for the node position, and the recursive splitting will continue on the resulting subsets until a stopping criterion has occurred (for instance, maximum tree depth or minimum sample leaf size). Another well-known criterion for evaluation is information gain, which quantifies the reduction in uncertainty/entropy attributed to a given split. Equation (2) defines the entropy of a set of samples S:

$$H(S) = -\sum_{i=1}^{k} p_i log_2(p_i) \qquad (2)$$

where p_i is the probability of the element in the set S. Information Gain (IG) is then the difference between the entropy of the parent node and the weighted average entropy of the child nodes after a split, as shown in Equation (3):

$$IG = H(S) - \sum_{i=1}^{m} \frac{|S_i|}{|S|} H(S_i) \qquad (3)$$

where S_i represents the subsets resulting from the split and $|S|$ and $|S_i|$ denote the sizes of the parent and child sets, respectively. A split that yields the maximum information gain is then selected. In addition to the said splitting criteria, the decision tree has proved to be a very powerful tool for capturing non-linear relationships, allowing the resultant tree model to be precise while still remaining interpretable.

Random Forest is an ensemble learning technique that unites output from many decision trees to get higher accuracy in classification with lowered risk due to overfitting. While a decision tree would ordinarily compromise on accuracy by making a classification mistake, Random Forest creates many gently overfitted models that average their outputs for regression or majority voting for classification. Randomness, brought into play by random sampling along with feature selection for each tree, contributes to reducing variance and thus enhances generalization on new data, [11].

The training of each tree in the forest is done on a different subset of features to introduce additional diversity. Further, to increase this diversity, each tree is trained by selecting a random subset of features at each split rather than taking into account all features therefore permitting fewer correlations between trees, [11]. The feature selection process serves to avoid overfitting and to render one tree less correlated with another. Nevertheless, Gini impurity or entropy is still used in choosing the best feature for a particular split by any of the individual trees in training; yet the random selection step makes sure that no attribute rules the model. Equation (4) shows the Gini impurity at a given split:

$$Gini = 1 - \sum_{i=1}^{k} p_i^2 \tag{4}$$

where pi is the proportion of class ii in node t, and k is the number of classes. The goal is to minimize the Gini impurity across all trees by selecting the best splits.

Once all trees in the forest are trained, Random Forest predicts the class label by taking a majority vote across the trees for classification problems. If T trees are trained, and the output of each tree titi is a class y_i, the Random Forest prediction y_{RF} is the class that appears most frequently among the individual tree predictions, as given in Equation (5):

$$y_{RF} = majority\ vote\ (y_1, y_2, \ldots, y_T) \tag{5}$$

In regression tasks, the prediction is the average of the outputs of all trees, as shown in Equation (6):

$$y_{RF} = \frac{1}{T} \sum_{i=1}^{T} y_i \tag{6}$$

The KNN Classification Technique is a simple yet very powerful non-linear classification method that assumes similar data points generally reside in the same class. In KNN classification, it is found in feature space and assigned to the class that most frequently occurs among those neighbors, [12]. Distance calculation between points usually occurred through a distance metric such as Euclidean distance. Equation (7) gives the Euclidean distance between two points $x=(x_1,x_2,\ldots,x_d)$ and $y=(y_1,y_2,\ldots,y_d)$ in a d-dimensional feature space is given by:

$$Distance = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{7}$$

where x_i and y_i are the feature values of the points x and y, respectively. Once the distances from the query point to all points in the dataset are computed, the k smallest distances are selected, and the class with the highest frequency among these neighbors is assigned to the query point.

To select the optimal value of k, cross-validation techniques are often used to balance the bias-variance trade-off, [12]. A smaller k can lead to higher variance (overfitting), as the model may be overly sensitive to noise in the data, while a larger k tends to smooth out predictions, possibly leading to higher bias (underfitting). The predicted class for a given query point x_q with K-Nearest Neighbors is determined by the majority voting rule for classification, as shown in Equation (8):

$$\hat{y}(x_q) = mode(y_1, y_2, \ldots, y_k) \tag{8}$$

where $y_1,y_2,\ldots,y_k$ are the class labels of the KNN of x_q. In the case of regression, the prediction would be the average of the target values of the neighbors, as given in Equation (9):

$$\hat{y}(x_q) = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{9}$$

While KNN is intuitive and effective for many non-linear classification problems, it does suffer from high computational complexity, especially in large datasets, as it requires calculating the distance between the query point and all points in the dataset.

5.1 Non-Linear Mapping Interpretation

The classification is modelled as a mapping from feature space to classes, with non-linear boundaries formed through ensemble and distance-based methods.

6 Non-Linear Classification

In this section, we describe the methodological framework to realize network slicing in the context of a CBRS system with non-linear classifiers. Our goal is to classify incoming media flows into three separate network slices, eMBB (Enhanced Mobile Broadband) for throughput insensitive media, (URLLC Ultra-Reliable Low Latency Communication slice) for delay-sensitive applications, and mMTC (Massive Machine Type Communication slice) for applications which are sensitive to loss/overhead based on application semantics. Classification is performed by considering random Forest (RF), Decision Tree (DT), and K-Nearest Neighbors (KNN) classifiers. Figure 1, gives a suggested block diagram for the CBRS system with network classification for three network slices. The block diagram presented in Figure 1 is an original conceptual design proposed by the authors for this study.

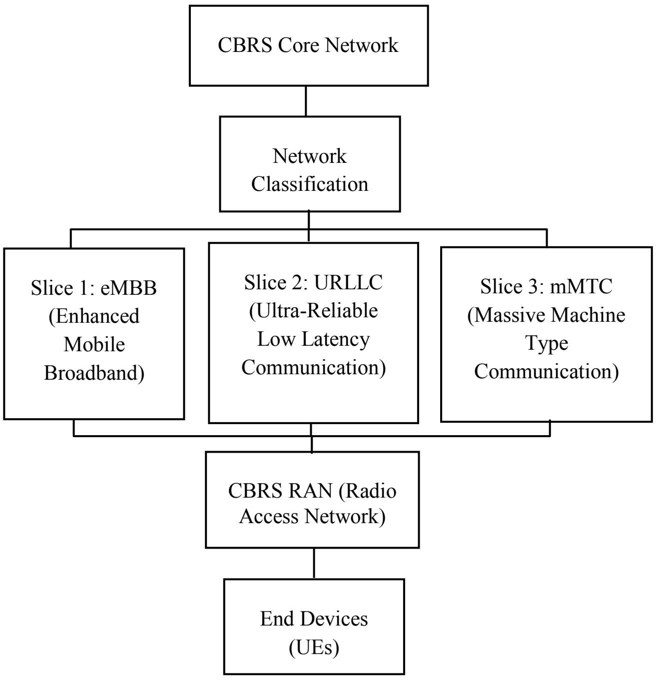

Fig. 1: CBRS Network Slicing Block Diagram
Source: created by the authors

6.1 System Model

Let the CBRS system consist of users and spectrum resources. Each user is assigned a priority level corresponding to Incumbent, PA, and GAA tiers. The allocation must satisfy priority, interference, and availability constraints.

6.2 Optimization Problem

The network slicing problem is formulated as given in Equation (10):

Maximize:

$$\sum_{i=1}^{n} \alpha \cdot SEi - \beta \cdot Li - \delta \cdot C \qquad (10)$$

Subject to:

- CBRS priority constraints
- QoS requirements
- Spectrum availability

Where SE is spectral efficiency, L latency, and C computational cost.

7 Data Preparation

The dataset in this work includes more than 30,000 traffic samples that correspond to a particular application setting in a CBRS enabled environment [13]. It includes synthetic traffic profiles emulating the UE traffic characteristics for eMBB, URLLC, and mMTC use-cases. These samples are synthetic in nature and reflect the communication requirements of the real world in various verticals.

Features include:

- Latency requirements (ms).
- Data rate (Mbps).
- Packet arrival rate (pps).
- Mobility level (static, pedestrian, vehicular).
- QoS priority level.
- Reliability requirement.
- Application type (categorical: smartphone, healthcare, IoT, etc.)

The dataset is first labelled using predefined QoS thresholds based on 3GPP standards into one of three categories. It is then pre-processed by normalization and stratified sampling is implemented to retain class distribution into training and testing sets of 70% and 30%, respectively.

8 Results and discussion

To study the performance of the proposed non-linear classification technique for network slicing in the CBRS system, MATLAB was used as the primary simulation tool. The classification model was implemented using MATLAB's machine learning software, which facilitated training and testing various non-linear classifiers. The MATLAB simulation environment and classification models were independently developed for this study, and no previously published code or implementation frameworks were reused. MATLAB's visualization and performance measurement capabilities enabled comprehensive analysis of classification accuracy, confusion matrices, and resource allocation efficiency in various scenarios.

The accuracy with which the non-linear classification algorithms (DT, RF, and KNN) perform network slicing under the CBRS system is plotted in Figure 2. The predictive allocation capabilities are assessed by the comparison of the actual allocation with that predicted by real-time network parameters.

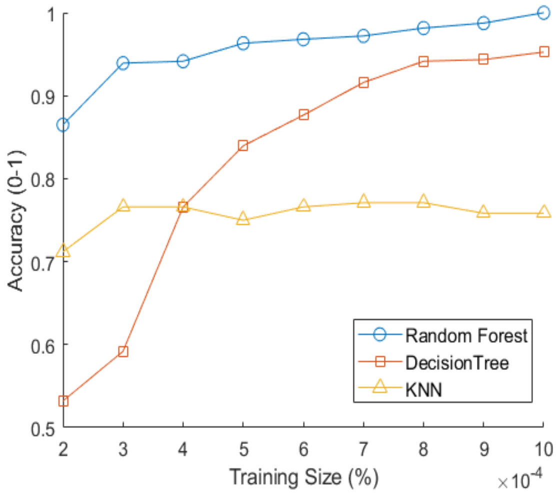

Fig. 2: Classification Accuracy
Source: created by the authors

Random Forest (RF) was thus able to realize its higher accuracy as compared to the other two algorithms, which is capable of variance reduction to yield optimum classification results, especially in CBRS environments facing complex, dynamic variations. It can be said that the former could handle overfitting, mainly due to how the algorithm utilizes feature randomness at each split, better than both DT and KNN. Decision Trees (DTs) performed competently but had slightly less accuracy than RF. However, when model simplicity is taken into consideration, DT continues to be a powerful candidate. While Random Forest achieves higher accuracy, it incurs increased processing time due to its ensemble nature. In contrast, Decision Trees offer faster predictions with lower computational overhead. KNN does quite well in some cases, especially when the training data has a high density and feature distributions are more uniform. Overall, however, its accuracies are typically lower compared to RF and DT due to the challenges of distance computation in high-dimensional space and its high sensitivity to noise in the data. Table 1, shows the classification accuracy for the three different three non-linear classification algorithms.

Table 1. Classification Accuracy

Classifier	Accuracy (%)
Random Forest (RF)	94.7%
Decision Tree (DT)	88.3%
K-Nearest Neighbours (KNN)	78.1%

Source: created by the authors

Figure 3 shows the time for each algorithm, in seconds/decision slice allocation. The time indicates how fast the algorithms can classify new CBRS network traffic data.

Decision Trees (DT) have the faster processing time, because of the simplicity of its single tree. Once trained, they are efficient in making predictions with little computation overhead. This gives DT an edge in free environments where quick decisions are required with little computational resource use. Random Forest (RF), it should be known, has a better mapping accuracy. Still, this model faces issues due to the performance required to process several trees included in the ensemble. The more trees, the longer the time required by RF. Nonetheless, for scenarios other than the performance of the CBRS, RF is still feasible to implement even if a slight delay is acceptable with a better prediction accuracy. Nonetheless, for scenarios other than the performance of the CBRS, RF is still feasible to implement even if a slight delay is acceptable with a better prediction accuracy.

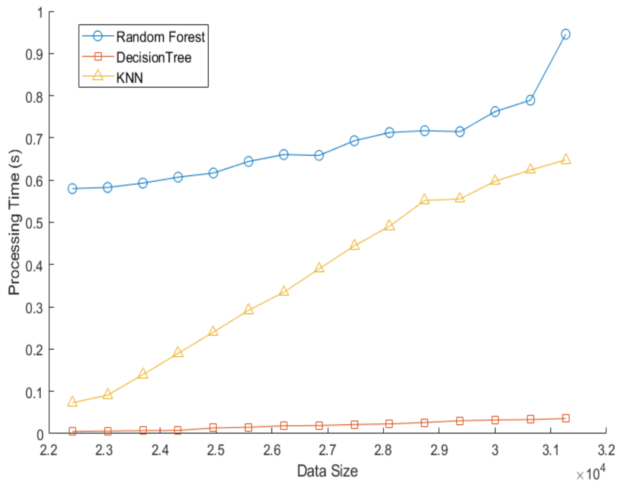

Fig. 3: Processing Time
Source: created by the authors

KNN processing time is long; it computes the best distance between the class, with all data points always put in the dataset. Computationally, it has very demanding requirements as the data size escalates. It would mean that KNN could cause significant possible bottlenecks in real-time CBRS applications since quick decisions matter the most.

9 Conclusions

In summary, Random Forest is superior in terms of accuracy among CBRS network slicing algorithms, while it incurs a higher computational time. To balance accuracy with processing time, Decision Trees would be amicable in scenarios where interpretability of the model and speed of the decision-making process are critical parameters.

Ahmed N. H. Alnuaimy,
Iham Tariq Ismail Al-Nuaimi

Although KNN is suitable in varying network conditions such as the presence of noise, changing traffic patterns, and varying degrees of channel conditions, it has slow processing time, especially when used in large CBRS scenarios, and will be less than helpful when real-time decisions are needed.

References:

[1] Jai, Naru, et al., Optimal channel allocation in the CBRS band with shipborne radar incumbents, *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). IEEE*, 2021.

[2] Jai, Naru, et al., Modeling and Optimization of Channel Allocation for PAL and GAA Users in the CBRS Band, *IEEE Transactions on Cognitive Communications and Networking* (2023).

[3] Zhang, Shunliang. ,An overview of network slicing for 5G, *IEEE Wireless Communications* 26.3 (2019): 111-117.

[4]] Li, Xin, et al., Network slicing for 5G: Challenges and opportunities., *IEEE Internet Computing* 21.5 (2017): 20-27.

[5] Zhang, Haijun, et al., Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges, *IEEE communications magazine* 55.8 (2017): 138-145.

[6] Khan, Latif U., et al., Network slicing: Recent advances, taxonomy, requirements, and open research challenges, *IEEE Access 8* (2020): 36009-36028.

[7] Malkoc, Mirsad, and Hisham A. Kholidy, 5g network slicing: Analysis of multiple machine learning classifiers, *arXiv preprint arXiv*:2310.01747 (2023).

[8] Phyu, Hnin Pann, Diala Naboulsi, and Razvan Stanica, Machine learning in network slicing—a survey, *IEEE Access 11* (2023): 39123-39153.

[9] Le, Luong-Vy, et al., SDN/NFV, machine learning, and big data driven network slicing for 5G, *2018 IEEE 5G World Forum (5GWF). IEEE*, 2018.

[10] Song, Yan-Yan, and L. U. Ying, Decision tree methods: applications for classification and prediction, *Shanghai archives of psychiatry* 27.2 (2015): 130.

[11] Parmar, Aakash, Rakesh Katariya, and Vatsal Patel, A review on random forest: An ensemble classifier, *International conference on intelligent data communication technologies and internet of things (ICICI) 2018. Springer International Publishing*, 2019.

[12] Zhang, Shichao, et al., Learning k for knn classification, *ACM Transactions on Intelligent Systems and Technology (TIST)* 8.3 (2017): 1-19.

[13] Kaggle User. 5G Network Slicing Analysis; Kaggle: 2023; Available at: https://www.kaggle.com/code/amohankumar/5g-network-slicing-analysis (Accessed: 28 May 2025).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.