BİTLİS EREN ÜNİVERSİTESİ
FEN BİLİMLERİ DERGİSİ

ISSN: 2147-3129 / e-ISSN: 2147-3188

Article Type : Research Article
Received : March 3, 2025
Revised : April 7, 2025
Accepted : August 9, 2025
DOI : 10.17798/bitlisfen.1650456

Year : 2025
Volume : 14
Issue : 3
Pages : 1420-1439

PREDICTING BID VERIFICATION IN SPECTRUM AUCTIONS: A DATA-DRIVEN APPROACH

Ceren Nisa AVCU [1], **Ali DEĞİRMENCİ** [1,*], Ömer KARAL [1]

[1] *Ankara Yıldırım Beyazıt University, Electrical and Electronics Engineering Department, Ankara, Türkiye*

**** Corresponding Author:*** *alidegirmenci@aybu.edu.tr*

ABSTRACT

Spectrum auctions are very important for the strategic allocation of frequency bands in the telecommunications industry, ensuring efficient and fair access to this valuable resource. However, the complexity of auction environments—characterized by vast state spaces and multidimensional bid attributes—renders manual bid verification infeasible. This study introduces an innovative, data-driven approach by utilizing machine learning models, including k-nearest neighbors, support vector machines, decision trees, and stochastic gradient descent classifiers, to automate the verification process. Through hyperparameter tuning and rigorous k-fold cross-validation, the decision tree model emerged as the most effective, achieving an F1-score of 96% and a G-Mean of 97%. These results demonstrate the practical viability of AI-enhanced verification systems in spectrum auctions and suggest broader applicability across various high-stakes auction platforms where real-time, reliable validation is essential.

Keywords: Classification, Machine learning, Spectrum auctions.

1 INTRODUCTION

Spectrum auctions play a crucial role in frequency allocations, especially in the telecommunications sector, ensuring the efficient and fair allocation of frequency bands. However, these processes are quite complex and require processing large amounts of data to verify bids from bidders. Traditional spectrum auction methods are computationally expensive and time-consuming, and errors in these processes can negatively impact auction outcomes. Therefore, optimizing verification of spectrum auctions for speed and reliability is essential to improving the overall efficiency and fairness of the auction process.

Numerous studies have explored spectrum auction in literature. Bailey investigated price prediction in art auctions and its impact on the art market [1]. The study compared image-based models, such as Convolutional Neural Networks (CNNs), with text-based and numerical data analysis models. Results showed that image-based models had lower performance. Rodríguez et al. examined collusion in public sector auctions, where companies secretly agree on bid prices for future auctions [2]. Using datasets from five countries—Brazil, Italy, Japan, Switzerland, and the United States—the study applied eleven machine learning algorithms. The top-performing models were Extra Trees (83%–86% precision), Random Forest (RF) (80%–84%), and AdaBoost (78%–82%). Imhof et al. focused on detecting cartel involvement in auctions by developing an algorithm to identify bid manipulation [3]. Datasets from Japan, Italy, and Switzerland were analyzed using four machine learning algorithms. The Super Learner algorithm achieved the highest classification accuracy at 90.5%. Abidi et al. studied shill bidding—fraudulent participation in auctions to drive up prices [4]. They developed the Dynamic Fused Machine Learning for Shill Bidding (DFM-SB) model, which combines multiple algorithms, including Artificial Neural Network (ANN), Support Vector Machines (SVM), Decision Tree (DT), and RF. The DFM-SB model achieved an accuracy of 99.8%. Zhang et al. addressed low participation and irrational bidding in IoT auctions by introducing a multi-round bidding mechanism based on a second-price sealed bid auction [5]. They implemented five machine learning algorithms: Ordered Value (OV) regression, OV neural networks, an auction screening model, a difference-of-convex algorithm, and a double deep Q-network. OV regression performed best on linear datasets (mean deviation: 35.3 ± 0.4), while the auction screening model excelled on non-linear datasets (mean deviation: 39.1 ± 0.7). Prathuri et al. applied machine learning to estimate the cost of selling players in sports leagues [6]. Their framework incorporated six regression models: DT regressor, k-Nearest Neighbors (kNN), Linear Regression, Stochastic Gradient Descent (SGD), RF regressor, and Support Vector Regression (SVR). SVR and Linear Regression provided the best results. Kusonkhum et al. used classification techniques to predict over-budget prices in Thai government construction projects [7]. The study applied kNN, ANN, and DT algorithms, achieving accuracies of 75%, 77.6%, and 77.3%, respectively.

As evident from the detailed literature review, while there are studies related to the Spectrum Auction dataset, most have focused on auction duration. Some studies focus on prediction of outcomes as well, e.g. [8] applied Fuzzy Neural Networks to the same dataset with mentionable accuracy, but their work did not particularize on analyzing model behaviors under

class imbalances. Similarly, [9] addressed spectrum auctions by swarm learning method, however their work did not analyze general machine learning algorithms such as SVM. Apart from these studies, our study, aims to predict verification results in spectrum auctions (i.e., verification.result), with efficient machine learning algorithms. To achieve this, four machine learning algorithms— SGDClassifier, DT, kNN, and SVM—were applied. Algorithm-specific hyperparameter optimization was performed to enhance performance. The model's effectiveness was evaluated using F1-Score and G-Mean metrics since those metrics are more compatible with imbalanced data.

The structure of this article is as follows: Section 2, a literature review of the dataset used in this study is given. Section 3 provides a detailed explanation of the dataset. Section 4 introduces the benchmarked machine learning algorithms. Section 5 discusses performance metrics, feature importance, and experimental results. Finally, Section 6 presents conclusions and recommendations for future research.

2 LITERATURE REVIEW

Similar studies have been conducted in literature using the dataset employed in this research. In spectrum auction, two key applications can be performed: estimating the auction duration and predicting the auction outcome.

Ordoni et al. conducted a study focusing on dataset explanation and the prediction of both auction duration and auction outcomes [10]. The performance of the RF method was evaluated using the Matthews Correlation Coefficient (MCC) metric. In another study, Ordoni et al. proposed an algorithm for validating process models that support data value changes [11]. This algorithm addresses the state space explosion problem by utilizing binary coding and Binary Decision Diagrams (BDD) to transform data value functions into Petri Nets. Fischer et al. analyzed auction duration in their OpenML-CTR23 benchmark package [12]. They evaluated five machine learning models—XGBoost, RF, a generalized additive model, ridge regression, and a regression tree—using root mean square error (RMSE) values, which were 0.394, 2.972, 6.140, 6.301, and 3.155, respectively. Tan proposed a method to enhance regression tree performance by introducing outlier detection [13]. Initially, a regression tree model was built using training data, and predictions significantly deviating from the terminal node’s mean were labeled as outliers. These outliers were used in an outlier detection algorithm, and during testing, their predictions were discarded, improving the model’s performance. The Mean Absolute Error (MAE) decreased from 1508 ± 105 to 1157 ± 189 after removing outliers.

Tajabadi et al. introduced a framework that enables nodes to obtain personalized models based on their contribution rates, serving as a fair reward mechanism in Swarm Learning (SL) systems [9]. Two machine learning methods—Deep Learning (DL) and RF—were used. DL models, including VGG8 and EfficientNetV2S, were trained on datasets such as CIFAR-10, CIFAR-100, Fashion-MNIST, and Reuters Newsletter using transfer learning. RF performance was assessed using the MCC metric, showing a 23% increase in MCC values as contribution rates rose in the spectrum auction dataset. The study demonstrated that higher contributions led to better model performance and encouraged collaboration. Campos Souza et al. investigated spectrum auction and the use of Fuzzy Neural Networks (FNNs) for improved fraud detection and expertise extraction [8]. They implemented a three-layer FNN integrating Gaussian neurons, fuzzy rules, and Leaky-ReLU activation functions. Comparative analysis with conventional machine learning methods—Naive Bayes, Neural Networks, SVM, kNN, and DT—showed that FNN achieved superior performance, with an accuracy of 96.21%, specificity of 90.4%, and an area under the curve (AUC) of 93.7%.

3 DATASET

This study utilizes the Spectrum Auction dataset from the UCI Machine Learning Repository [10], which contains extensive data on the German 4G spectrum auction. While the dataset includes approximately 130,000 validation records defining specific targets and features, it comprises only 2,043 unique feature-value combinations. Approximately 13% of the samples had correct verification results, while the rest were incorrect.

The dataset captures various data objects and their values, representing different stages of the auction process. Each object is described by a set of properties reflecting its state at a given phase of the auction. For example, attributes such as the frequency band may be modified or supplemented with additional attributes as the auction progresses. This dynamic structure enables the dataset to reflect the evolving nature of auction objects.

The dataset includes two target variables: verification.time, representing the auction duration, and verification.result, indicating the auction outcome. This study focuses on predicting verification.result. Details of the dataset features are provided in Table 1.

Table 1. Details of the dataset

Variable Name	Description
process.b1.capacity	Maximum number of bids Bidder 1 can win.
process.b2.capacity	Maximum number of bids Bidder 2 can win.
process.b3.capacity	Maximum number of bids Bidder 3 can win.
process.b4.capacity	Maximum number of bids Bidder 4 can win.
property.price	Verified price of the product.
property.product	Verified product in the auction.
property.winner	Winner of the bid.
verification.result	Verification outcome whether the verified outcome is possible or not.
verification.time	Duration of the auction.

The pre-processing steps that are used in this study contain different approaches than other studies. This study focuses on verification.result, rather than verification.time, which [10] and [8] have been focused. This required special attention to the unbalanced class structure of the dataset (around 13% correct validation results). To address this imbalance dataset, traditional balancing methods (e.g., over-sampling or under-sampling) were not applied, instead opting for metrics for model evaluation that are more compatible with imbalanced datasets, such as F1-Score and G-Mean. The normalization steps applied to the dataset in other studies were deliberately restricted to evaluate the performance of algorithms such as SGDClassifier, DT, kNN and SVM on raw data, which is the main focus of this study. This approach aims to better reflect the robustness of the selected algorithms in the face of unbalanced and large-scale raw data and their applicability in real-world scenarios. Thus, unlike some studies in the literature, no extensive preprocessing (e.g., outlier removal, feature scaling) was applied to the dataset in this study, a methodological choice adopted to evaluate the ability of the models to learn patterns directly from the raw data.

4 METHODS

This study investigates the performance of four machine learning algorithms—SGDClassifier, kNN, DT, and Support Vector Machine (SVM)—for predicting verification outcomes in spectrum auctions. To improve model accuracy, hyperparameter tuning was conducted. The overall experimental design is presented schematically in Figure 1. The following sections provide a detailed explanation of the machine learning techniques, offering a structured insight into the analytical procedures employed throughout the study.

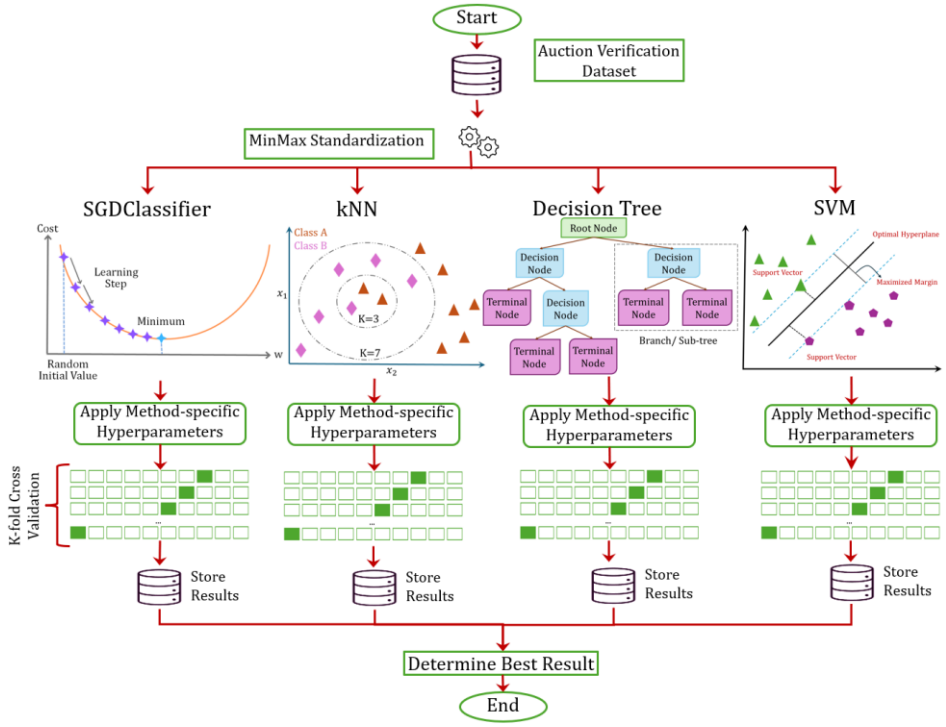

Figure 1. The flowchart of the study.

4.1 Stochastic Gradient Descent-Based Classifier (SGDClassifier)

In this study SGDClassifier is used, which is a linear classifier provided by the scikit-learn library. SGD Classifier is an algorithm that is trained using SGD, by taking the gradient of the loss of each sample at a time and updating the model along the way. SGDClassifier is a supervised machine learning algorithm commonly used for classification tasks, particularly in large datasets. It is a variant of the gradient descent optimization algorithm, which minimizes the loss function by updating model parameters iteratively. On the other hand, SGDClassifier utilizes this algorithm to solve especially big and sparse datasets. Unlike the traditional gradient descent, SGDClassifier updates parameters based on the whole dataset in each iteration. This approach significantly reduces computational load, making the algorithm well-suited for real-time systems and memory-efficient applications. Despite its advantages, SGD-based algorithms have some limitations. The learning rate must be carefully tuned—if set too high, the algorithm may overshoot the optimal solution, whereas a low learning rate can result in slow convergence. Additionally, SGD-based algorithms are sensitive to the initial parameter values, which can affect the outcome. To mitigate these challenges, several variants of SGD have been developed, such as mini-batch gradient descent, which processes small batches of data points to balance the benefits of both SGD and batch gradient descent [14]. A schematic diagram of SGD, which underlies the training of the SGDClassifier is shown in Figure 2.

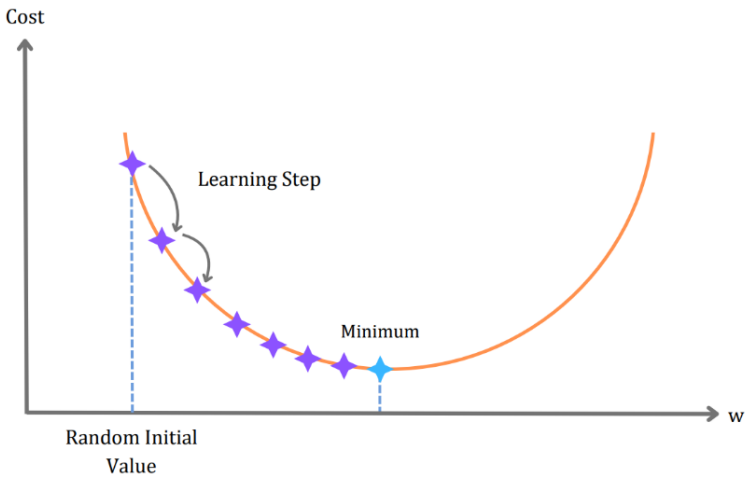

Figure 2. Schematic Diagram of SGD, which underlies the training process in SGDClassifier.

4.2 k-Nearest Neighbors (kNN)

kNN is a lazy learning algorithm that does not explicitly learn a model from training data. Instead, it stores all data points and makes predictions based on their proximity to new data points. The core principle of kNN is that closer data points tend to have similar characteristics.

When a new data point is encountered, kNN identifies the *k* nearest neighbors from the training data using a specified distance metric. In classification problems, the new data point is assigned to the most common class label among its *k* nearest neighbors. In regression problems, the predicted value is typically the average of the *k* nearest neighbors [15, 16].

The choice of distance metric significantly impacts kNN's performance. The most used distance metrics include:

- Euclidean distance
- Manhattan distance
- Minkowski distance

These metrics are formally defined as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{1}$$

$$d(x,y) = \sum_{i=1}^{k} |x_i - y_i| \tag{2}$$

$$d(x,y) = \left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q} \tag{3}$$

where, x and y represent two data points, x_i and y_i are the i^{th} feature of the data points. k is the number of the dimensions, and it can be said that in Minkowski distance metric, q is a parameter that determines the type of distance calculation. When q is 2, it becomes the Euclidean distance, and when q is 1, it becomes the Manhattan distance [17].

In the kNN algorithm, k represents the number of nearest neighbors considered for making a prediction. The optimal value of k should be adjusted based on the dataset and the specific problem at hand. A schematic representation of kNN is shown in Figure 3. As illustrated in the figure, when k = 3, the query sample is assigned to the class represented by the red triangles. However, when k = 7, the query sample is assigned to the class represented by the pink diamonds.

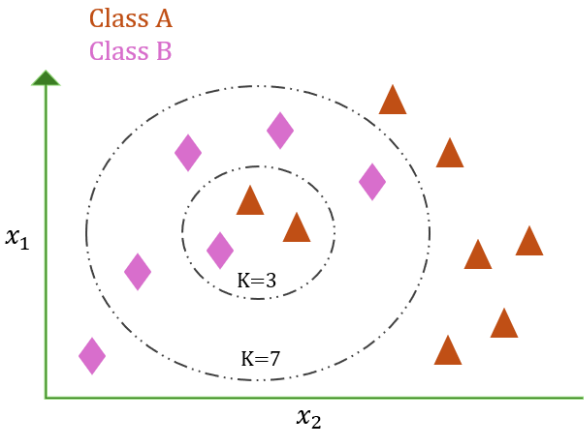

Figure 3. Schematic Representation of kNN.

4.3 Decision Tree (DT)

DT algorithm is a supervised learning method commonly used for classification tasks. It organizes data into a hierarchical structure of nodes and branches, which helps make decisions by evaluating multiple conditions step by step [18, 19].

In a Decision Tree, there are three types of nodes:

1. Root Node (Decision Node): The topmost node representing the entire dataset.

2. Internal Nodes: These nodes represent the points where the dataset is further split based on additional features.
3. Leaf Nodes (Terminal Nodes): These nodes represent the outcome or final decision.

Branches in the tree connect nodes and indicate the relationship between decisions and possible outcomes. A schematic representation of the Decision Tree structure is shown in Figure 4.

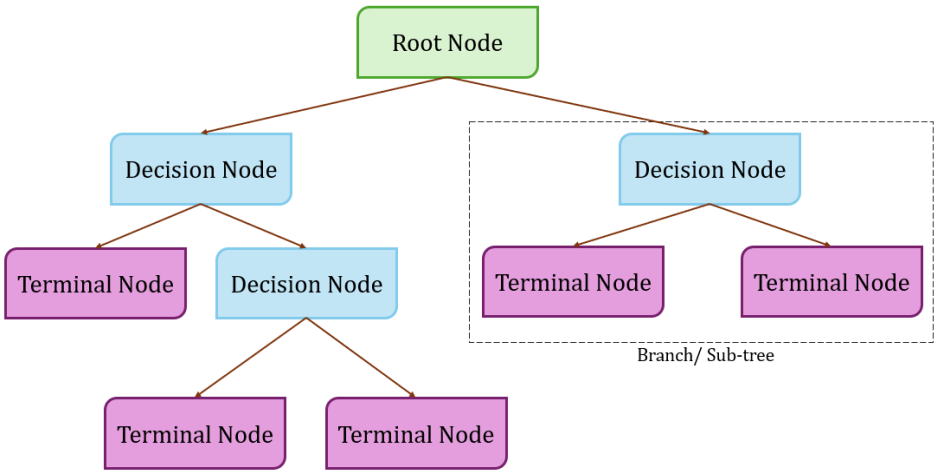

Figure 4. Structural representation of Decision Tree.

There are several advantages to using the Decision Tree (DT) algorithm. It effectively handles missing values and is robust against outliers. Additionally, DT is a non-parametric algorithm, meaning it does not rely on predefined functional forms. However, despite its advantages, DT also has some drawbacks. One key issue is that overfitting—the model can become too complex, leading to poor generalization on unseen data. Moreover, DT is prone to instability: a minor change in the dataset can result in a completely different model.

4.4 Support Vector Machine (SVM)

SVM is a machine learning algorithm used to assign labels to objects by learning from examples. The primary goal of SVM is to find a maximum margin hyperplane that best divides the dataset into distinct classes. The ideal hyperplane is the one with the maximum distance from both classes [20].

In a two-dimensional space, this hyperplane is a line, but in higher dimensions, it becomes a plane. Therefore, SVM is highly effective when working with multidimensional datasets [21].

A linear SVM assumes that the data is linearly separable, meaning it can be divided by a straight line. However, in real-world scenarios, it is not always guaranteed that datasets are linearly separable. In such cases, a linear SVM struggles to find an appropriate hyperplane.

To overcome this limitation, SVM maps the data into higher-dimensional spaces, allowing for the separation of classes in a non-linear manner. This transformation is accomplished using the kernel trick, which enables SVM to find non-linear decision boundaries.

Several kernel functions are available for SVM, including:

- Radial Basis Function (RBF)
- Polynomial
- Sigmoid

Among these, RBF is one of the most widely used kernels. It has two hyperparameters:

1. Gamma: Determines how far the influence of a single training example reaches. A low gamma results in a smoother decision boundary, while a high gamma gives each training example a more localized area of influence.
2. C: Controls the trade-off between margin size and classification errors. It also acts as a regularization parameter in RBF-SVM, balancing the complexity of the model and overfitting.

A schematic representation of the optimal hyperplane in SVM is shown in Figure 5.

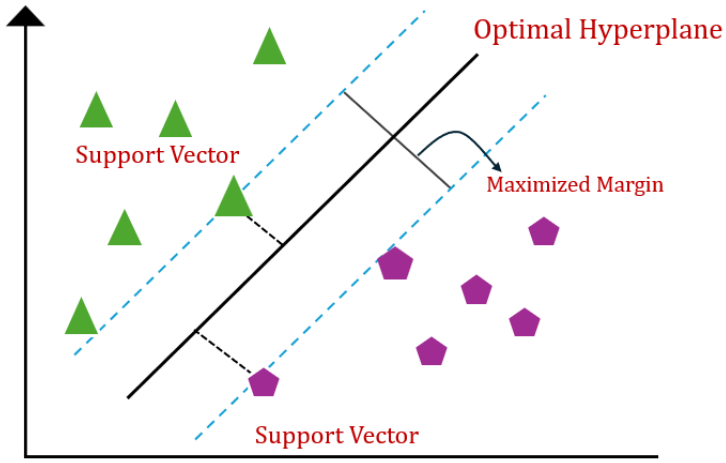

Figure 5. Optimal hyperplane of SVM.

5 RESULTS AND DISCUSSION

5.1 Performance Criteria

Five different performance metrics have been used to evaluate the effectiveness of the machine learning algorithms. The commonly used performance metrics for classification tasks include:

- Accuracy
- Precision
- Recall
- F1-Score
- G-Mean

Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome. Accuracy can be written as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{4}$$

which also can be written as,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP stands for true positive that is true prediction of an event. TN stands for true negative that is an event which has not occurred in prediction. FP stands for false positive, and it represents positive prediction for an event that has not occurred. FN stands for false negative and represents negative prediction of an event that has not occurred.

Precision is the quality of a positive prediction made by the model and is represented as:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F1-Score evaluates the performance of a model by observing recall and precision values. F1-Score is given by:

$$F1 - Score = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (7)$$

G-Mean measures the ability of the model to accurately predict both positive and negative classes in a balanced manner. It is calculated by taking geometric means of sensitivity and specificity. Sensitivity measures the proportion of data points with positive labels that are correctly classified by the model whereas specificity measures the proportion of data points with a negative label that are classified by the model. Sensitivity is defined as:

$$Sensitivity = \frac{TP}{TP + FN} \quad (8)$$

Specificity is defined as:

$$Specificity = \frac{TN}{TN + FP} \quad (9)$$

Finally, G-Mean is represented as:

$$G - Mean = \sqrt{Sensitivity * Specificity} \quad (10)$$

5.2 K-Fold Cross Validation

The k-fold cross-validation technique is a reliable and effective method for evaluating the performance of machine learning algorithms. Its primary goal is to enhance the generalization ability of the model by using the data efficiently.

The technique involves dividing the dataset into K parts. Each part is designated as a test set, while the remaining parts are used as the training set. In each iteration, one part serves as the validation data, and the other parts are used for training. The model is trained in the training data and tested on the validation data. This process is repeated k times, with a different validation set used in each iteration [22].

The performance metrics obtained in each iteration are recorded, and at the end of the K iterations, the average of these metrics is calculated. This average value provides an overall assessment of the model's performance across different data splits. By testing the model on different parts of the data, k-fold cross-validation helps reduce the risk of overfitting.

Overall, k-fold cross-validation is a widely preferred method for model evaluation because it makes efficient use of data and offers a more reliable performance assessment.

5.3 Feature Importance

Assessing feature importance is essential for interpreting machine learning models, especially in fields that prioritize transparency and explainability. This process measures how much each input variable contributes to the model's predictive accuracy, allowing researchers to pinpoint the key factors influencing results. By ranking these variables, one can improve model interpretability and optimize performance through dimensionality reduction. To evaluate the contribution of individual features toward the classification task, mutual information (MI) is applied. MI quantifies the amount of information obtained about the target variable through each input feature, capturing both linear and non-linear dependencies [23]. Figure 6 demonstrates the MI scores computed for each feature in the spectrum auction dataset. The analysis reveals that property.winner and property.price are the most informative features, contributing substantially more than the others. Specifically, property.winner exhibits the highest mutual information score, indicating a strong dependency with the target variable. In contrast, features related to process capacities, such as process.b1.capacity, process.b2.capacity, and process.b4.capacity, show relatively lower MI scores, suggesting limited direct informational value for the spectrum auction. The process.b4.capacity feature contributed the least, which may indicate redundancy or weak association with the output variable.

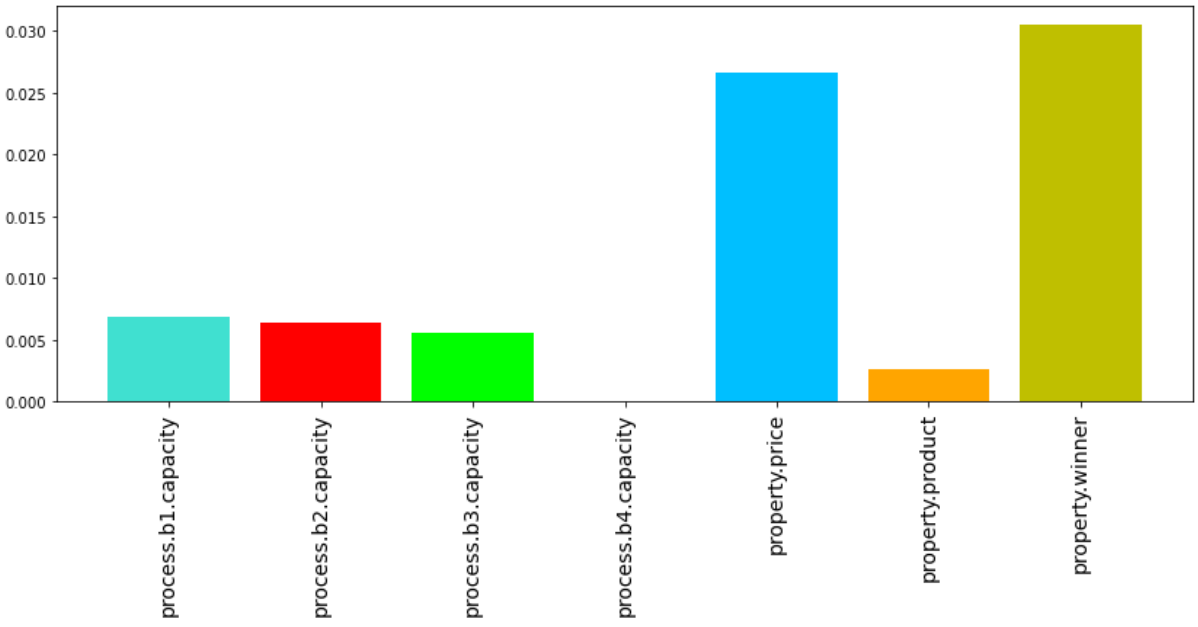

Figure 6. Mutual information score of the features.

5.4 Experimental Results

SGDClassifier, kNN, DT, and SVM were implemented to determine the verification of auction outcomes. Each of these machine learning algorithms has hyperparameters that control the structure of the model and the training process, and these hyperparameters need to be fine-tuned to achieve optimal performance.

To ensure more consistent and reliable results, the number of folds (K) was set to 10, and 10 iterations were performed for each algorithm. The dataset used in the study exhibits an imbalanced class distribution, which can lead to misleading performance evaluations. For example, the accuracy metric can be deceptive in imbalanced datasets, as it may show high scores by predominantly favoring the majority class, while failing to capture the model's performance on the minority class.

To address this issue and provide clearer insights, the F1-Score and G-Mean metrics were selected and plotted for each benchmarked algorithm, as they are more sensitive to imbalanced data [24].

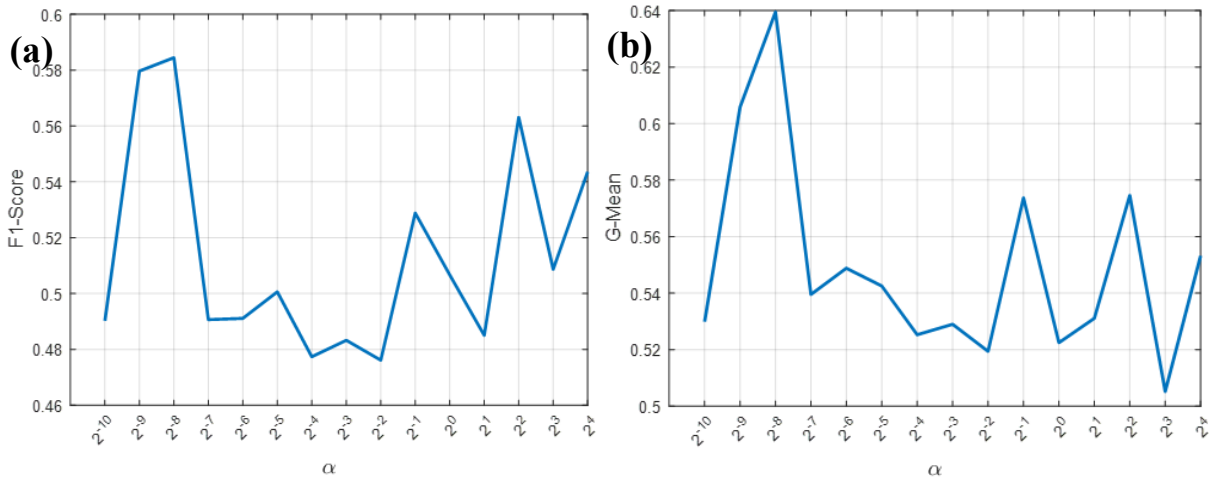

Figure 7. (a) Results of SGDClassifier F1-Score; (b) G-Mean.

SGDClassifier is an efficient and easily implemented machine learning algorithm. It uses the α parameter to prevent overfitting. The α parameter applies a penalty to limit the size of the model's weights, thus helping to avoid overfitting. For this study, the values of α ranged from 2^{-10} to 2^{4}. The results for the F1-Score and G-Mean are shown in Figure 7(a) and Figure 7(b), respectively. Both model performances initially improved as α increased and peaked at 2^{-8}, and then mostly gradually decreased. This indicates that moderate regularization is optimal.

The kNN algorithm uses the k parameter to determine the number of neighbors considered when making decisions for the test sample. In kNN, models with small k values are more sensitive to noise and are more prone to overfitting, whereas models with large k values are more generalized but may be more prone to underfitting. To identify the best results, k was analyzed within the range of 1 to 30, with values increased one by one.

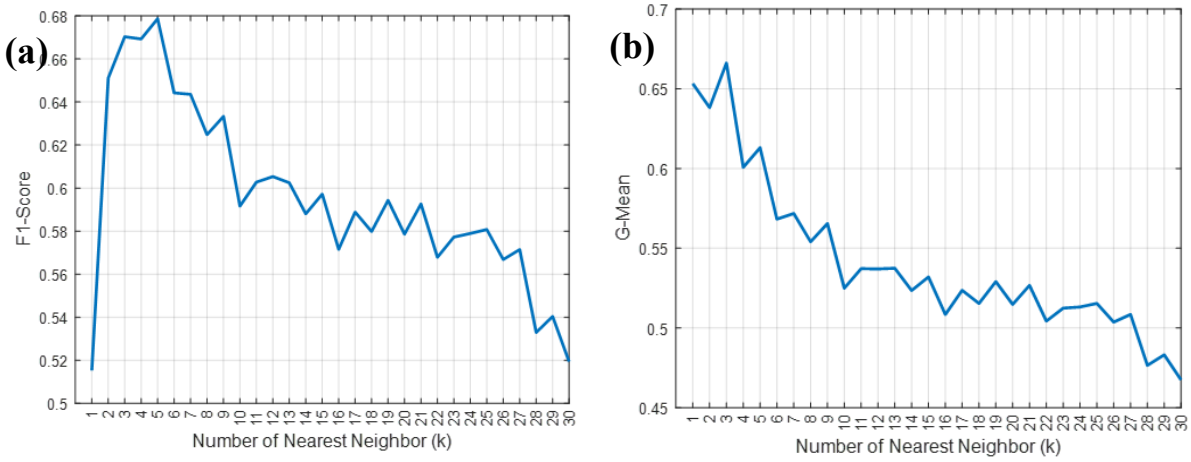

Figure 8. (a) Performance of kNN F1-Score; (b) G-Mean.

As shown in Figure 8(a) and Figure 8(b), the algorithm achieved the best F1-Score when k was set to 5, while the best G-Mean result was obtained when k was set to 3. For both metrics, the results gradually decreased as the k value increased further. This is convenient with the behavior of kNN, since smaller k values can capture local class patterns. The higher G-Mean score when k was set to 3 means that the sensitivity was improved in both classes.

DT uses the min_sample_split and max_depth hyperparameters to control the structure of the tree and the model's generalization ability. The min_sample_split parameter determines the minimum number of samples required to split an internal node, while max_depth sets the maximum depth of the tree. For this study, min_sample_split values ranged from 1 to 17, and max_depth values ranged from 1 to 20.

The results for F1-Score and G-Mean metrics in the DT method are shown in Figure 9(a) and Figure 9(b), respectively. For both metrics, the best results were obtained with the same hyperparameters: max_depth = 9 and min_sample_split = 4. When the max_depth parameter exceeded 9, the results remained nearly constant which indicates that increasing the tree depth beyond 9 does not contribute to further improvements. This also indicates increasing the depth may risk overfitting.

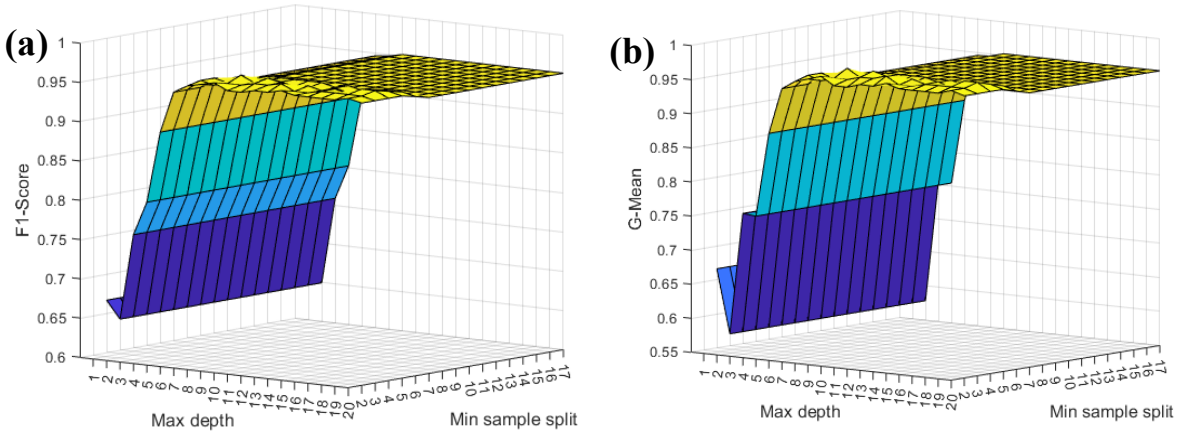

Figure 9. (a) Results of DT F1-Score; (b) G-Mean.

SVM with an RBF kernel takes the C and γ hyperparameters to balance the generalization performance of the algorithm. C value is taken between the range of 2^3 to 2^{14}, whereas γ value is taken between the range of 2^{-4} to 2^5. Figures 10(a) and 10(b) illustrate the F1-Score and G-Mean results of the SVM method for the RBF kernel, respectively. As illustrated in both figures, the performance improves with the increase of C, indicating that the model benefits from placing higher penalties on misclassification. This is consistent with the nature of the dataset, where correct instances are scarce and need to be captured decisively. On the contrary, extreme values of γ tends to degrade performance, suggesting that overly localized or overly generalized decision boundaries are detrimental in this context. The highest F1-Score and G-Mean values were obtained when C is 2^{14} and γ is 2^{-1} in both metrics.

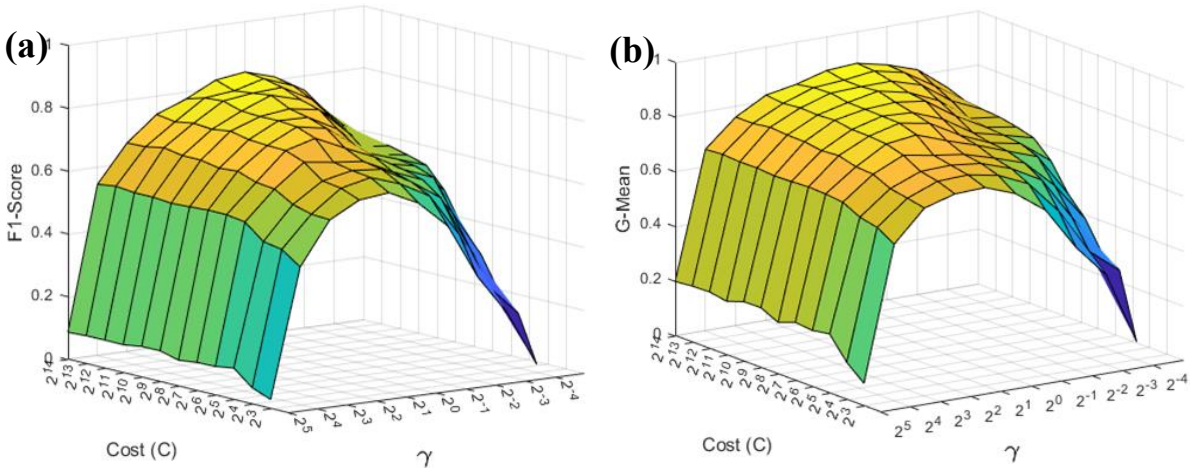

Figure 10. (a) Performance of SVM F1-Score; (b) G-Mean.

Table 2 presents the performance metrics—accuracy, precision, recall, G-Mean, and F1-Score—for the benchmarked algorithms. The hyperparameters chosen for each algorithm were based on the results that yielded the highest G-Mean for that specific algorithm. The choice to prioritize the G-Mean metric is due to the imbalance in the dataset used in this study.

Table 2. Performance results

Methods	Accuracy	Precision	Recall	G-Mean	F1-Score
SGDClassifier α= 0,00390625	0.8821	0.7341	0.6123	0.6394	0.5844
kNN k=3	0.9013	0.7537	0.6498	0.6661	0.6703
Decision Tree max_depth= 9, min_sample_split=4	0.9911	0.9609	0.9807	0.9757	0.9679
RBF-SVM C=2^{14}, γ=2^{-1}	0.9619	0.8901	0.7925	0.8807	0.8385

According to Table 2, the DT algorithm achieved the best overall performance among the benchmarked algorithms. Given the characteristics of the dataset, G-Mean performance results were used for evaluation. It can be observed that the highest G-Mean result (0.97) was obtained with the Decision Tree. Similarly, for the F1-Score, the highest result (0.96) was also achieved by the Decision Tree. On the other hand, the lowest results for all metrics were observed with SGDClassifer, where the G-Mean performance was 0.63, and the F1-Score was 0.58. This result can be explained by its linear nature and sensitivity which makes SGDClassifier less compatible for complex and imbalanced datasets.

The performance gap between the DT and SGDClassifier can be explained by the models' differences in their modeling capacities. While the DT algorithm is more suitable for structure datasets such as Spectrum Auction dataset, as it can identify important features and handle non-linear relationships. Furthermore, adjusting min_sample_split and max_depth to their optimal range allows the DT algorithm to generalize more efficiently without overfitting.

On the contrary, SGDClassifier assumes linear separability and is highly sensitive to the learning rate α and other relevant parameters. Those features make the SGDClassifier less suitable for this particular dataset.

The best performance result of DT can be explained by several reasons. Firstly, to achieve a balance between model complexity and generalization, comprehensive hyperparameter tuning was implemented with max_depth and min_samples_split. Conversely, previous studies did not point out this kind of hyperparameter tuning nor utilize expanded optimization techniques. Secondly, while the authors [8] employed 3-fold-cross-validation, we employed 10-fold-cross-validation over 10 iterations which makes the model results more reliable. Furthermore, even though RF and Fuzzy Neural Network (FNN) employed by [9] and

[8] are powerful ML algorithms, those algorithms also can be prone to overfitting and require more data to generalize to model effectively. The structure of DT can be especially convenient for this dataset since Spectrum Auction dataset is relatively structured and interpretable, and DT can capture discrete decision boundaries without the risk of excessive complexity.

6 CONCLUSION AND SUGGESTION

This study investigates the prediction of spectrum auction results using four distinct machine learning algorithms: SGDClassifier, kNN, DT, and SVM. Each algorithm's specific hyperparameters were optimized to achieve the best performance. To ensure the reliability of the results, 10-fold cross-validation was employed, and a range of performance metrics—accuracy, recall, precision, G-Mean, and F1-Score—were utilized for evaluation.

Among the benchmarked algorithms, the Decision Tree (DT) method with a max_depth of 9 and min_samples_split of 4 provided the best overall results. The DT model achieved approximately 99% accuracy, 96% precision, 98% recall, 97% G-Mean, and 96% F1-Score. The second-best results were observed using the SVM with an RBF kernel. In contrast, the SGDClassifier algorithm delivered the lowest performance across all metrics. Although the accuracy metric was relatively high across all methods, significant disparities were observed in other metrics. The gap between the best and worst results in terms of accuracy was around 11%, but it widened to 34% in G-Mean and 39.62% in F1-Score.

Future studies could explore other machine learning algorithms to enhance classification performance. Additionally, deep learning models have the potential to capture more complex patterns within the dataset. Advanced approaches, such as transfer learning or fine-tuning pre-trained models, hold promises for improving results by leveraging insights from similar datasets, which warrants further exploration.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Artificial Intelligence (AI) Contribution Statement

Authors All components of this study, including its design, data analysis, and scientific contributions, were conducted solely by the authors. ChatGPT® was utilized exclusively to enhance the grammar and clarity of a few selected sentences.

Contributions of the Authors

Ceren Nisa Avcu: Writing – original draft, Writing – review & editing, Conceptualization, Investigation, Methodology,

Ali Değirmenci: Methodology, Software, Writing – original draft, Writing – review & editing, Visualization, Validation

Ömer Karal: Conceptualization, Supervision, Writing – review & editing

REFERENCES

[1] J. Bailey, "Can machine learning predict the price of art at auction?," *Harvard Data Science Review,* vol. 2, no. 2, pp. 1-8,2020.

[2] M. J. G. Rodríguez, V. Rodríguez-Montequín, P. Ballesteros-Pérez, P. E., Love, and R. Signor, "Collusion detection in public procurement auctions with machine learning algorithms," *Automation in Construction,* vol. 133, p. 104047, 2022. doi:10.1016/j.autcon.2021.104047

[3] D. Imhof and H. Wallimann, "Detecting bid-rigging coalitions in different countries and auction formats," *International Review of Law and Economics,* vol. 68, p. 106016, 2021. doi: 10.1016/j.irle.2021.106016

[4] W. U. H. Abidi, M. S. Daoud, B. Ihnaini, M. A. Khan, T. Alyas, A. Fatima, and M. Ahmad, "Real-time shill bidding fraud detection empowered with fussed machine learning," *IEEE Access,* vol. 9, pp. 113612-113621, 2021. doi: 10.1109/ACCESS.2021.3098628.

[5] R. Zhang, C. Jiang, J. Zhang, J. Fan, J. Ren, and H. Xia, "Reinvigorating sustainability in Internet of Things marketing: Framework for multi-round real-time bidding with game machine learning," *Internet of Things,* vol. 24, p. 100921, 2023. doi: 10.1016/j.iot.2023.100921

[6] J. Rani P., A. Kulkarni, A. V. Kamath, A. Menon, P. Dhatwalia and D. Rishabh, "Prediction of Player Price in IPL Auction Using Machine Learning Regression Algorithms," *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT),* Bangalore, India, pp. 1-6, 2020. doi: 10.1109/CONECCT50063.2020.9198668

[7] W. Kusonkhum, K. Srinavin, and T. Chaitongrat, "The adoption of a big data approach using machine learning to predict bidding behavior in procurement management for a construction project," *Sustainability,* vol. 15, no. 17, p. 12836, 2023. doi: 10.3390/su151712836

[8] P. V. d. C. Souza and M. Dragoni, "Knowledge extraction in auction verification employing techniques from machine learning and fuzzy neural networks.," *2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, Madrid, Spain, pp. 1-8, 2024. doi: 10.1109/EAIS58494.2024.10569109.

[9] M. Tajabadi and D. Heider, "Fair swarm learning: Improving incentives for collaboration by a fair reward mechanism," *Knowledge-Based Systems,* vol. 304, p. 112451, 2024. doi: 10.1016/j.knosys.2024.112451

[10] E. Ordoni, J. Bach and A. -K. Fleck, "Analyzing and Predicting Verification of Data-Aware Process Models–A Case Study With Spectrum Auctions," *IEEE Access,* vol. 10, pp. 31699-31713, 2022. doi: 10.1109/ACCESS.2022.3154445

[11] E. Ordoni, J. Mülle, K. Yang, and K. Böhm, "Efficient Verification of Process Models Supporting Modifications of Data Values," *2022 IEEE 24th Conference on Business Informatics (CBI),* Amsterdam, Netherlands, pp. 21-30, 2022. doi: 10.1109/CBI54897.2022.00010.

[12] S. F. Fischer, M. Feurer, and B. Bischl, "OpenML-CTR23–a curated tabular regression benchmarking suite," *Proc. AutoML Conf. 2023 (Workshop)*, 2023.

[13] S. C. Tan, "Enhancing regression tree predictions with terminal-node anomaly detection," *2023 6th Artificial Intelligence and Cloud Computing Conf.,* pp. 21-26, 2023. doi: 10.1145/3639592.3639596

[14] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review,* vol. 60, no. 2, pp. 223-311, 2018. doi:10.1137/16M1080173

[15] A. Ozbay, A. Degirmenci, and O. Karal, "A Comparative Analysis of Machine Learning Algorithms for Accurate Step Detection in Wrist Worn Devices," *2023 Innovations in Intelligent Systems and Applications Conference (ASYU),* Sivas, Turkiye, pp. 1-5, 2023. doi: 10.1109/ASYU58738.2023.10296741.

[16] A. Degirmenci and O. Karal, "iMCOD: Incremental multi-class outlier detection model in data streams," *Knowledge-Based Systems,* vol. 258, p. 109950, 2022. doi: 10.1016/j.knosys.2022.109950

[17] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, vol. 2, New York: Springer, 2009, pp. 1-758.

[18] M. Çakır, A. Degirmenci, and O. Karal, "Exploring the behavioural factors of cervical cancer using ANOVA and machine learning techniques," *Int. Conf. Science, Engineering Management and Information Technology,* Feb. 2022, pp. 249-260, Cham: Springer Nature Switzerland.

[19] A. N. Karaoglu, H. Caglar, A. Degirmenci, and O. Karal, "Performance Improvement with Decision Tree in Predicting Heart Failure," *2021 6th International Conference on Computer Science and Engineering (UBMK)*, Ankara, Turkey, pp. 781-784, 2021. doi: 10.1109/UBMK52708.2021.9558939.

[20] M. Muttaqi, A. Degirmenci, and O. Karal, "US Accent Recognition Using Machine Learning Methods," *2022 Innovations in Intelligent Systems and Applications Conference (ASYU),* Antalya, Turkey, pp. 1-6, 2022. doi: 10.1109/ASYU56188.2022.9925265.

[21] W. S. Noble, "What is a support vector machine?," *Nature Biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006. doi: 10.1038/nbt1206-1565

[22] Ö. Karal, "Performance comparison of different kernel functions in SVM for different k value in k-fold cross-validation," *2020 Innovations in Intelligent Systems and Applications Conference (ASYU),* Istanbul, Turkey, pp. 1-5, 2020. doi: 10.1109/ASYU50717.2020.9259880

[23] Değirmenci, A. "Machine Learning Models for Accurate Prediction of Obesity: A Data-Driven Approach," *Turkish Journal of Science and Technology,* vol. 20, no. 1, pp. 77-90, 2025. doi: 10.55525/tjst.1572382

[24] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evolutionary Computation,* vol. 17, no. 3, pp. 275-306, 2009. doi: 10.1162/evco.2009.17.3.275

[25] E. Ordoni, J. Bach, A.-K. Fleck, and J. Bach, "Auction verification," *UCI Machine Learning Repository,* 2022. [Online]. Available: https://doi.org/10.24432/C52K6N.Ge