

# A Comparative Analysis of Machine Learning Algorithms for Intelligent Spectrum Sensing in Cognitive Radio Networks

Md. Tofail Ahmed\*, Mousumi Haque\*\*, Yosuke Sugiura\*, Tetsuya Shimamura\*

\*Dept. of Information and Computer Sciences, Saitama University, Saitama, 338-8570, Japan

\*\*Dept. of Information & Communication Engineering, University of Rajshahi, Rajshahi 6205, Bangladesh

md.t.a.812@ms.saitama-u.ac.jp, mouice12@gmail.com, ysugiura@mail.saitama-u.ac.jp,  
shima@mail.saitama-u.ac.jp

**Abstract**—A branch of artificial intelligence (AI), machine learning (ML) is a rapidly advancing technology that is driving significant transformations across numerous fields. In the field of spectrum sensing, researchers have introduced ML techniques to enhance sensing accuracy and overcome the limitations of conventional methods. This study presents a performance comparison between two main categories of ML algorithms, which are classification and regression for ML-based spectrum sensing. As classification algorithms, we consider naive Bayes (NB), support vector machine (SVM), decision tree (DT), k-nearest neighbors (KNN), linear discriminant analysis (LDA), generalized additive model (GAM), binary linear classifier (BLC) with logistic learner, logitboost ensemble classifier (LEC), binary Gaussian kernel (BGK) with logistic learner, and neural network (NN) with sigmoid function. In contrast, regression methods include linear regression learner (LRL) with least squares, SVM for regression (SVMR), Gaussian process regression (GPR), DT for regression (DTR), least-squares boosting (LSBoost) ensemble for regression (LER), and NN for regression (NNR). Performance is analyzed through simulations using quadrature amplitude modulation (QAM) in a low signal-to-noise ratio (SNR) environment over an additive white Gaussian noise (AWGN) channel. Evaluation metrics such as accuracy, precision, probability of detection, probability of false alarm, and sensing error are used to assess the effectiveness of all the ML algorithms.

**Index Terms**—Spectrum sensing, GAM, LEC, BGK, LRL, DTR, SVM, NN, machine learning, and cognitive radio.

## I. INTRODUCTION

In this modern age, global mobile data traffic is expanding significantly. Current estimates from the international telecommunication union (ITU) predict that worldwide monthly mobile data traffic will reach 607 exabytes (EB) by 2025 and 5016 EB by 2030. It is also projected that approximately 70% of the global population will use mobile services by 2025, with 60% of those users accessing the mobile internet [1]. The massive growth in data traffic will lead to increased demands for diverse services and ultra-high-speed wireless communications characterized by ultra-high reliability and ultra-low latency. Cognitive radio (CR) devices or applications, such as computers, laptops, smartphones, sensing equipment, and internet of everything (IoE), are the primary sources of

this tremendous increase in data traffic. Therefore, handling the huge data traffic requires the intelligent and efficient use of the valuable spectrum bands in wireless communication systems. Spectrum sensing in CR detects whether licensed spectrum bands are occupied by primary users (PUs). If the bands are vacant, meaning unoccupied by PUs, then the CR devices, known as secondary users (SUs), are permitted to access the bands.

Machine learning (ML) is now a leading approach in spectrum sensing to address the shortcomings of conventional methods. In [2], a spectrum prediction model is demonstrated by employing several established ML classifiers, including random forest (RF), logistic regression (LR), support vector machine (SVM), k-nearest neighbors (KNN), decision tree (DT), and artificial neural network (ANN). The study in [3] compares traditional energy detection (ED) with ML-based approaches, specifically LR, KNN, and neural networks (NN). Their findings show that the NN model performs better than the others, which is suitable for the non-linear nature of the data and does not rely on a preset threshold. The performance of the k-means classifier is compared to non-cooperative, AND-based, and OR-based spectrum sensing approaches, illustrating that the k-means-based model marginally outperforms the OR-based approach in strong noisy environments [4]. A spectrum sensing method is presented in [5] that uses ANN and SVM models as binary classifiers to identify PU presence in low signal-to-noise ratio (SNR) environments while reducing computing complexity compared to threshold-based ED. In [6], the performance of smart spectrum sensing is compared among four popular supervised ML models, including naive Bayes (NB), SVM, gradient boosting machine (GBM), and random forest (RF). To handle the complex dataset, they used principal component analysis (PCA) to reduce dimensionality while preserving critical information. The study [7] compares ML models like multilayer perceptron (MLP), SVM, and NB against traditional approaches like maximum ratio combining (MRC) and AND/OR rules for channel occupancy identification in CNR. A key finding in [7] is that the MLP model provides the best trade-off between rapid training with highly effective channel occupancy detection.

While existing research in ML-based spectrum sensing has primarily focused on classification methods, there is a lack of a comprehensive comparison between the major categories of classification and regression algorithms.

This study presents a comprehensive comparative analysis of the most popular classification and regression algorithms for ML-based spectrum sensing. On the classification side, we evaluate a suite of algorithms, including NB, SVM, DT, KNN, linear discriminant analysis (LDA), generalized additive model (GAM), binary linear classifier (BLC) with logistic learner, a logitBoost ensemble classifier (LEC), a binary Gaussian kernel (BGK) with logistic learner, and a NN with a sigmoid function. For regression, we examine linear regression learner (LRL) with least squares, SVM for regression (SVMR), Gaussian process regression (GPR), DT for regression (DTR), a least-squares boosting (LSBoost) ensemble for regression (LER), and a NN for regression (NNR). Performance is evaluated using accuracy, precision, probability of detection, probability of false alarm, and sensing error.

## II. FUNDAMENTALS OF SPECTRUM SENSING

The primary goal of the spectrum sensing method is to determine whether a PU occupies a specific frequency band or if it is free. The free band, also known as the spectrum hole, can be used by an SU without causing harmful interference. The problem can be formulated as a classic binary hypothesis test, with null and alternative hypotheses denoted as  $H_0$  and  $H_1$ , respectively.  $H_0$  indicates that the PU signal is missing, and just noise is present. In contrast,  $H_1$  indicates the presence of the PU signal as well as noise. The fundamental equation for the received signal,  $y(n)$ , at the  $n$ -th sample (where  $n \in [0, 1, 2, \dots, N-1]$ ) is expressed as follows [8]:

$$y(n) = \begin{cases} H_0: & y_0(n) = u(n), \\ H_1: & y_1(n) = x(n) + u(n), \end{cases} \quad (1)$$

where  $x(n)$  represents the clean modulated signal of the PU for a total of  $N$  samples and  $u(n)$  is additive white Gaussian noise (AWGN) with mean zero and variance  $\sigma_u^2$ , i.e.,  $u(n) \sim \mathcal{N}(0, \sigma_u^2)$ .  $y_0(n)$  and  $y_1(n)$  are the received signals by the SU when the PU is inactive and active, respectively. This research considers the AWGN channel, indicating that the channel coefficient equals one. The received signal at an SU is initially processed, and features are extracted to determine whether the PU is active or not. The decision is then made by either a classical (e.g., ED) detector, which applies a hypothesis test against a predefined threshold, or a learning-based model (e.g., ML), which operates as a classifier reliant on the accuracy of its prediction of the model against the actual spectrum state.

## III. DATASET, ML IMPLEMENTATION, AND PERFORMANCE METRICS

Modern spectrum sensing techniques rely on high-quality datasets and the selection of a suitable ML model. The following subsections provide a short description of dataset preparation, ML implementation, and performance assessment measures.

### A. Dataset

It is challenging to obtain a real-world dataset for spectrum sensing that meets the desired requirements. To address this, we generate a synthetic dataset using the methodology outlined in [9], which is based on the energy and standard deviation of the real and imaginary components of the fast Fourier transform (FFT) of received signals. For each SNR value, the dataset is generated using a 4-QAM modulation scheme with a total of  $N = 1000$  samples. The dataset dimensions are  $O \times 5$ , where  $O$  is the number of samples or observations, and 5 represents dataset attributes, where the first four columns are features or independent variables  $X$  belonging to energy real, energy imaginary, standard deviation real, and standard deviation imaginary, and the last column is the response or class variable  $Y$ , which belongs to 0 or 1, denoted as  $Y \in [0, 1]$ , where the value of  $Y$  is set to 0 when the PU is inactive and  $Y$  set to 1 when the PU is active.

### B. Implementation of ML Algorithms

In this paper, the performance of the most popular ML classification and regression models is chosen and compared. ML classification and regression are the two primary techniques of supervised learning, each developed for a certain kind of predicting problem. Classification is largely concerned with predicting discrete categorical labels, answering "what" questions, such as detecting whether an email is spam or not, and generating a medical diagnosis based on a set of possibilities. The goal of a classification model is to learn a mapping from inputs  $X$  to outputs  $Y$ , where  $Y \in [1, \dots, C]$ , with  $C$  being the number of classes and  $X \in \mathbb{R}$ , where  $\mathbb{R}$  and  $X$  represent real numbers and features of the dataset, respectively. If  $C = 2$ , this is referred to as binary classification (in which case it assumes  $Y \in [0, 1]$ ); if  $C > 2$ , this is referred to as multiclass classification [10]. Meanwhile, regression in ML is concerned with predicting continuous numerical values, answering "how much" problems such as estimating property prices, projecting sales income, and predicting stock market movements. The regression model is similar to classification, except that the response variable  $Y$  is continuous. It maps the real-valued input  $X \in \mathbb{R}$  to the real-valued outputs  $Y \in \mathbb{R}$  [10]. A regression model is selected based on the dataset. For example, linear regression is employed for continuous response variables, logit regression for binary outcomes ( $Y \in [0, 1]$ ), and polynomial regression for non-linear relationships. The logit regression algorithm and the binary ( $C = 2$ ) classification algorithm are both appropriate for spectrum sensing, as it is a binary classification problem.

In Algorithm 1, we show the implementation process of the ML model on the dataset. All classification and regression algorithms are trained through MATLAB functions; for instance, *fitcsvm*(.) and *fitrsvm*(.) functions are used for SVM classification model and the SVM regression model, respectively. The *predict*(.) function is employed to predict the output of a model for the testing data set, while *confusionmat*(.) generates the confusion matrix by comparing the predicted class with the actual class. After that, the performance metrics,

TABLE I  
BINARY CONFUSION MATRIX WITH MATHEMATICAL NOTATION

| Confusion Matrix |                      |                      |
|------------------|----------------------|----------------------|
| Predicted values | Actual values        |                      |
|                  | $Y = 1$              | $Y = 0$              |
| $\hat{Y} = 1$    | True [+ve] ( $TP$ )  | False [+ve] ( $FP$ ) |
| $\hat{Y} = 0$    | False [-ve] ( $FN$ ) | True [-ve] ( $TN$ )  |

such as accuracy,  $A_c$ , precision,  $P_c$ , probability of detection,  $P_d$ , probability of false alarm,  $P_f$ , and sensing error,  $S_e$ , are calculated from the confusion matrix. As a dataset preprocessing process, the functions  $cvpartition(\cdot)$ ,  $training(\cdot)$ , and  $test(\cdot)$  are used to split the dataset into training and testing data sets with a percentage of 80% for training and 20% for testing. Additionally, the min-max normalization method is applied to rescale the dataset features within 0 and 1 to achieve accurate and reliable results of the models, and it also leads to reducing training time.

### C. Performance Assessment Metrics

The important parameters, such as the confusion matrix,  $A_c$ ,  $P_c$ ,  $P_d$ ,  $P_f$ , and  $S_e$ , are used to assess the performance of all algorithms. Table I shows the binary confusion matrix, which compares the actual  $Y$  and predicted  $\hat{Y}$  outcomes of classification and regression models. The  $A_c$ ,  $P_c$ ,  $P_d$ ,  $P_f$ , and  $S_e$  can be computed from Table I, where the  $S_e$  is the average of the probability of miss detection  $P_m = 1 - P_d$  and  $P_f$ , formulated as  $S_e = (P_m + P_f)/2$  [11]. The mathematical equations of others are defined as follows [12], [13]:

$$A_c = \frac{TP + TN}{TP + TN + FP + FN}, P_c = \frac{TP}{TP + FP}, \quad (2)$$

$$P_d = \frac{TP}{TP + FN}, P_f = \frac{FP}{FP + TN}.$$

## IV. SIMULATION FINDINGS AND ANALYSIS

The validation of this work is performed via simulations in MATLAB R2023a, with the ML algorithms being implemented using the integrated statistics and machine learning toolbox. The simulations are performed on a computer running Windows 10 Home (64-bit), equipped with an Intel Core i7-9700K eight-core processor (3.60 GHz base clock) and 32 GB of random access memory (RAM). The PU signal is modulated through a 4-QAM modulation scheme with  $N = 1000$  and transmitted through an AWGN channel. For each SNR value, a dataset of  $O = 8000$  samples is generated. The data is then divided into an 80/20 split, resulting in 6400 samples (80% of 8000) for the training set and 1600 samples (20% of 8000) for the testing set.

Fig. 1 illustrates the visual representation of two features from the generated dataset, which are the standard deviation of the real and imaginary components of the received signal after FFT. The dataset, generated with an SNR of -13 dB, contains 8000 observations. Of these, 6400 samples (80% of

### Algorithm 1 Algorithm for ML Classification and Regression

- 1: **Input:** Dataset  $data\_set$  with features  $X = data\_set(:, 1:4)$  and labels  $Y = data\_set(:, 5)$
- 2: **Output:** Performance metrics for all algorithms
- 3: Split and preprocess the dataset:  
**Function** DATAPREPARATION():  
 a.  $X \leftarrow data\_set(:, 1:4)$  // Feature matrix  
 b.  $Y \leftarrow data\_set(:, 5)$  // Actual Label vector  
 c.  $cv \leftarrow cvpartition(Y, 'Holdout', 0.2)$  // 80-20 split  
 d.  $idxTrain \leftarrow training(cv)$  // Training set  
 e.  $idxTest \leftarrow test(cv)$  // Testing set  
 f.  $XTrain \leftarrow X(idxTrain, :)$   
 g.  $YTrain \leftarrow Y(idxTrain)$   
 h.  $XTest \leftarrow X(idxTest, :)$   
 i.  $YTest \leftarrow Y(idxTest)$   
 j. Normalize:  $XTrain$  and  $XTest$  // Apply min-max normalization  
**return**  $XTrain, YTrain, XTest, YTest$
- 4: Train and evaluate the model:  
**Function** TRAINANDEVALUATE( $class\_reg, XTrain, YTrain, XTest, YTest$ ):  
 a. Train model  $mdl$  on  $XTrain, YTrain$   
 b.  $\hat{YTest} \leftarrow predict(mdl, XTest)$  // Predicted Label vector for testing data set  
 c.  $[conf\_mat, order] \leftarrow confusionmat(YTest, \hat{YTest})$  // Generate confusion matrix  
 d. Extract:  $TN, FP, FN, TP$  from  $conf\_mat$   
 e. Compute:  $A_c, P_c, P_d, P_f, S_e$  using  $TN, FP, FN, TP$   
**return**  $A_c, P_c, P_d, P_f, S_e$
- 5: Start the main function:  
**Function** MAIN():  
 a. DATAPREPARATION // Call data preparation method  
 b. Initialize classifiers and regressions:  
   Classifiers\_Regressions = [ "NB", "fitcnb", "normal", "SVM", "fitsvm", "standardized", "DT", "fitctree", "max splits", "KNN", "fitcknn", "5 neighbors", "LDA", "fitcdiscr", "quadratic", "GAM", "fitcgam", "all interactions", "BLC", "fitclinear", "logistic", "LEC", "fitcensemble", "LogitBoost", "BGK", "fitckernel", "logistic", "NN", "fitcnet", "sigmoid", "LRL", "fitrlinear", "leastsquares", "SVMR", "fitsvm", "linear", "GPR", "fitrgp", "standardized", "DTR", "fitrtree", "max splits", "LER", "fitrensemble", "LSBoost", "NNR", "fitrnet", "sigmoid" ]  
**for each**  $class\_reg$  in  $Classifiers\_Regressions$  **do**  
   a.  $A_c, P_c, P_d, P_f, S_e \leftarrow TRAINANDEVALUATE(class\_reg, XTrain, YTrain, XTest, YTest)$   
   b. Display performance metrics for all algorithms  
**end**

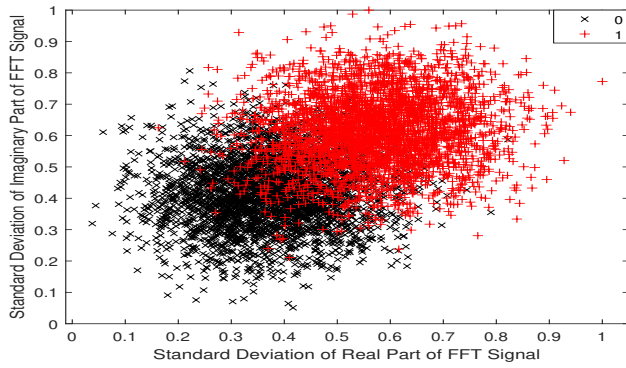


Fig. 1. Scatter plot of the generated dataset (training set) at SNR = -13 dB.

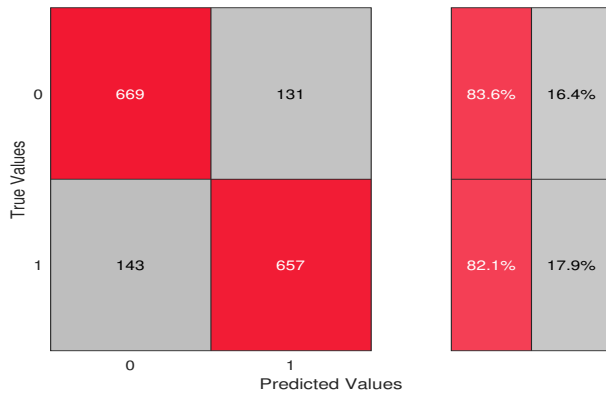


Fig. 2. Confusion matrix for testing set using NB model at SNR = -14 dB.

8000) are selected for the training set, as displayed in Fig. 1. A min-max normalization is applied to all features, rescaling the data so that  $X \in [0, 1]$ .

The effectiveness of all algorithms is evaluated by calculating the confusion matrix for the testing data set, which consists of 1600 samples (20% of 8000). For simplification, only the confusion matrix for the NB algorithm at SNR = -14 dB is illustrated in Fig. 2. The NB model provides robust and well-balanced outcomes, which are illustrated in the confusion matrix, which accurately identifies the majority of the 1600 total instances. The model exhibits nearly identical performance for both classes, with the ability to accurately predict class 0 being comparable to its capability to accurately predict class 1. This symmetry indicates the model does not favor one class over the other and maintains a high level of predictive reliability despite of the challenging low SNR.

Table II represents the performance comparison of various classification and regression models for testing data set taken from the generated dataset with SNR = -12 dB. The top-performing algorithms are BGK and NN classification models, which achieve the highest overall accuracy of 94.12%, coupled with high  $P_d$  above 94%,  $P_f$  near 6%, and  $S_e$  below 6%. Several regression models, including LRL and GPR, demonstrate competitive performance with accuracies around 93.90%. However, GAM is the least effective classifier in

TABLE II  
RESULTS OF PERFORMANCE ASSESSMENT METRICS

| Classification and Regression Algorithms |              |           |           |           |           |           |
|------------------------------------------|--------------|-----------|-----------|-----------|-----------|-----------|
| Model (Type)                             | Model (Name) | $A_c$ (%) | $P_c$ (%) | $P_d$ (%) | $P_f$ (%) | $S_e$ (%) |
| Classification                           | NB           | 93.88     | 93.33     | 94.50     | 6.75      | 6.13      |
|                                          | SVM          | 94.00     | 93.46     | 94.62     | 6.62      | 6.00      |
|                                          | DT           | 92.12     | 92.44     | 91.75     | 7.50      | 7.88      |
|                                          | KNN          | 93.62     | 92.87     | 94.50     | 7.25      | 6.38      |
|                                          | LDA          | 93.56     | 92.66     | 94.62     | 7.50      | 6.44      |
|                                          | GAM          | 91.12     | 90.72     | 91.62     | 9.38      | 8.88      |
|                                          | BLC          | 93.88     | 94.54     | 93.12     | 5.38      | 6.12      |
|                                          | LEC          | 92.69     | 92.63     | 92.75     | 7.38      | 7.31      |
|                                          | BGK          | 94.12     | 93.47     | 94.88     | 6.62      | 5.88      |
|                                          | NN           | 94.12     | 93.91     | 94.38     | 6.12      | 5.88      |
| Regression                               | LRL          | 93.81     | 93.65     | 94.00     | 6.38      | 6.19      |
|                                          | SVMR         | 93.62     | 93.73     | 93.50     | 6.25      | 6.37      |
|                                          | GPR          | 93.88     | 93.23     | 94.62     | 6.88      | 6.12      |
|                                          | DTR          | 92.12     | 92.44     | 91.75     | 7.50      | 7.88      |
|                                          | LER          | 92.19     | 91.72     | 92.75     | 8.38      | 7.81      |
|                                          | NNR          | 94.00     | 93.46     | 94.62     | 6.62      | 6.00      |

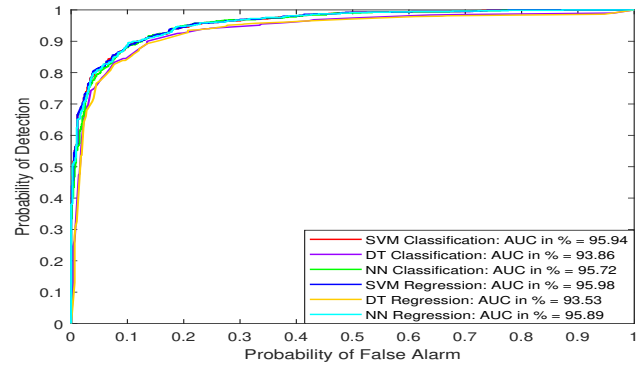


Fig. 3. ROC with AUC of ML models for testing set at SNR = -13 dB.

the comparison of classification models, with an accuracy of 91.12% and an elevated  $P_f$  of 9.38%. Overall, the findings show that both model types are suitable for spectrum sensing, with the best models establishing a strong balance between  $A_c$ ,  $P_c$ , and  $P_d$ .

Fig. 3 shows the performance of classification and regression ML models in a low environment with SNR = -13 dB using receiver operating characteristic (ROC) curves, which present the distinction between the  $P_d$  (finding actual signals) and the  $P_f$  (mistaking noise for a signal). Another important statistic is the area under the curve (AUC) in ROC, where a greater percentage shows the ability of a model to distinguish between 1 and 0 classes across all possible classification thresholds. All models perform quite well, while the best performers are the SVM and NN models, with the SVM regression approach providing the highest AUC of 95.98%, followed by its classification counterpart and neural network models. While the DT algorithms are slightly less effective in this scenario, their results are still excellent, confirming that all evaluated methods are capable of spectrum sensing.

TABLE III  
PERFORMANCE IN DISTINCT LOW SNR ENVIRONMENTS

| Accuracy (%) for Each SNR |              |          |          |          |          |          |
|---------------------------|--------------|----------|----------|----------|----------|----------|
| Model (Type)              | Model (Name) | -19 (dB) | -17 (dB) | -15 (dB) | -13 (dB) | -11 (dB) |
| Classification            | NB           | 60.00    | 69.25    | 79.56    | 88.69    | 97.50    |
|                           | SVM          | 60.50    | 69.00    | 79.62    | 88.50    | 97.56    |
|                           | DT           | 58.94    | 67.81    | 77.44    | 87.38    | 96.56    |
|                           | KNN          | 54.37    | 63.56    | 75.75    | 86.94    | 96.88    |
|                           | LDA          | 60.19    | 68.94    | 79.12    | 88.62    | 97.44    |
|                           | GAM          | 53.94    | 63.00    | 74.69    | 85.25    | 95.94    |
|                           | BLC          | 60.69    | 68.75    | 79.44    | 88.56    | 97.56    |
|                           | LEC          | 59.06    | 66.75    | 78.00    | 88.06    | 95.94    |
|                           | BGK          | 60.06    | 68.62    | 79.56    | 88.62    | 97.56    |
|                           | NN           | 60.50    | 69.00    | 79.31    | 87.69    | 97.31    |
| Regression                | LRL          | 60.69    | 68.88    | 79.44    | 88.69    | 97.44    |
|                           | SVMR         | 60.44    | 69.19    | 79.62    | 88.56    | 97.62    |
|                           | GPR          | 60.12    | 68.50    | 79.56    | 88.50    | 97.56    |
|                           | DTR          | 58.94    | 67.81    | 77.38    | 87.31    | 96.56    |
|                           | LER          | 54.19    | 61.94    | 73.00    | 86.50    | 96.12    |
|                           | NNR          | 60.88    | 68.31    | 78.88    | 88.50    | 97.50    |

TABLE IV  
COMPUTATIONAL COMPLEXITY OF ALL ALGORITHMS

| Computational Costs |              |                         |                        |                          |
|---------------------|--------------|-------------------------|------------------------|--------------------------|
| Model (Type)        | Model (Name) | Training time (Seconds) | Testing time (Seconds) | Memory usage (Kilobytes) |
| Classification      | NB           | 0.2149                  | 0.0242                 | 354.208                  |
|                     | SVM          | 0.4985                  | 0.0256                 | 469.72                   |
|                     | DT           | 0.1312                  | 0.0118                 | 322.65                   |
|                     | KNN          | 0.1372                  | 0.0415                 | 352.90                   |
|                     | LDA          | 0.1998                  | 0.0171                 | 307.79                   |
|                     | GAM          | 1.2902                  | 0.1396                 | 1711.493                 |
|                     | BLC          | 0.1905                  | 0.0083                 | 010.56                   |
|                     | LEC          | 0.8617                  | 0.1713                 | 201.73                   |
|                     | BGK          | 0.9070                  | 0.0119                 | 008.81                   |
|                     | NN           | 1.8753                  | 0.0137                 | 364.29                   |
| Regression          | LRL          | 0.0611                  | 0.0037                 | 009.78                   |
|                     | SVMR         | 0.4686                  | 0.0127                 | 672.25                   |
|                     | GPR          | 3.5799                  | 0.0465                 | 320836.37                |
|                     | DTR          | 0.0429                  | 0.0079                 | 321.74                   |
|                     | LER          | 0.6474                  | 0.0361                 | 2021.07                  |
|                     | NNR          | 0.9299                  | 0.0052                 | 363.37                   |

Table III compares the  $A_c$  of all ML models under low SNR conditions, ranging from -19 dB to -11 dB. The results demonstrate a clear and expected trend where the  $A_c$  of every model consistently improves as the SNR increases. Among the 16 models tested, several, including SVM, BLC, BGK, and NNR, consistently provide robust performance, maintaining a slight but noticeable performance advantage over others like KNN, DT, and GAM.

The training time, testing (or inference) time, and memory usage, namely computational costs, are critical factors that influence the choice of models for particular ML-based tasks. The computational costs for training (on 6400 samples) and testing (on 1600 samples) of all ML-based spectrum sensing algorithms are compared in Table IV. The findings show that simpler models for both classification and regression, such as DT/DTR, and simple classification, like NB, are very fast

to train and test, while more complicated models, such as GPR and NN/NNR, require significantly longer training times and greater memory resources. On the other hand, the linear models, like BLC, BGK, and LRL, demonstrate that high prediction speed and low memory usage during testing are achievable, even if their training times are moderate.

## V. CONCLUSION

This study evaluated the performance of various classification and regression algorithms for ML-based spectrum sensing. The findings indicate that logit regression models achieve performance nearly equivalent to that of common binary classification models. Among the tested algorithms, the NN model provides marginally higher results for both classification and regression tasks since its use of a sigmoid function, which is well-suited for binary outcomes. However, while complex models such as NN and GPR delivered higher accuracy, they required significantly more computational resources, in terms of both training time and memory, than simpler models like BLC, LEC, and LRL.

## REFERENCES

- [1] M. Alsabah, M. A. Naser, B. M. Mahmmod, S. H. Abdulhussain, M. R. Eissa, and A. Al-Baidhani, "6G wireless communications networks: a comprehensive survey," *IEEE Access*, vol. 9, pp. 148191–148243, Nov. 2021.
- [2] P. Pandian, C. Selvaraj, N. Bhalaji, K. G. Arun Depak, and S. Saikrishnan, "Machine learning based spectrum prediction in cognitive radio networks," in *2023 Int. Conf. on Netw. and Commun. (ICNWC)*, Chennai, India, Apr. 2023, pp. 1–6.
- [3] P. Nimudomsuk, M. Sanguanwattananaraks, K. Srisomboon, and W. Lee, "A performance comparison of spectrum sensing exploiting machine learning algorithms," in *18th Int. Conf. on Elect. Eng./Electron., Comput., Telecommun. and Inf. Technol. (ECTI-CON)*, Chiang Mai, Thailand, May 2021, pp. 102–105.
- [4] M. A. Abusubaih and S. Khamayseh, "Performance of machine learning-based techniques for spectrum sensing in mobile cognitive radio networks," *IEEE Access*, vol. 10, pp. 1410–1418, Dec. 2022.
- [5] R. Kumar, "Optimization of spectrum sensing in cognitive radio using machine learning approach," in *2nd Int. Conf. on Device Intell., Comput. and Commun. Technol. (DICCT)*, Dehradun, India, Mar. 2024, pp. 1–5.
- [6] S. Aghabeiki, C. Hallet, N. E. -R. Nouthou, N. Rassem, I. Adjali, and M. B. Mabrouk, "Machine-learning-based spectrum sensing enhancement for software-defined radio applications," in *IEEE Cognitive Commun. for Aerosp. Appl. Workshop (CCAAS)*, Cleveland, OH, USA, Jun. 2021, pp. 1–6.
- [7] C. H. A. Tavares, J. C. Marinello, M. L. Proenca Jr, and T. Abrao, "Machine learning-based models for spectrum sensing in cooperative radio networks," *IET Commun.*, vol. 14, no. 18, pp. 3102–3109, Nov. 2020.
- [8] A. Wang, Q. Meng, and M. Wang, "Spectrum sensing method based on residual dense network and attention," *Sensors*, vol. 23, no. 18, p. 7791, pp. 1–14, Sep. 2023.
- [9] M. T. Ahmed, M. Haque, Y. Sugiura, and T. Shimamura, "An algorithm to generate synthetic dataset for modern intelligent spectrum sensing in cognitive radio networks," in *9th Int. Symp. on Innov. Approaches in Smart Technol. (ISAS)*, Gaziantep, Turkiye, Jun. 2025, pp. 1–5.
- [10] K. P. Murphy, "Machine learning: a probabilistic perspective," Cambridge (Massachusetts, London): MIT Press, 2012.
- [11] E. V. Vijay and K. Aparna, "RNN-BIRNN-LSTM based spectrum sensing for proficient data transmission in cognitive radio," *e-Prime - Adv. in Elect. Eng., Electron. and Energy*, vol. 6, pp. 1–8, Dec. 2023.
- [12] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. of Data Mining & Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.
- [13] T. Fawcett, "An introduction to roc analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.