I. MODELS, METHODOLOGY AND RESULTS

In this section, we'll cover data preprocessing, the implementation of both machine learning and deep learning models on the dataset, the metrics employed to assess the model performance, and ultimately, the presentation of the obtained results.

A. Data preprocessing

Initially, the data set undergoes a crucial process of splitting into two subsets: the training set, comprising 80% of the data, and the test set, constituting the remaining 20%. This division ensures that the model is trained on a substantial portion of the data but also evaluated on unseen data to gauge its real-world performance.

Prior to introducing our data set to machine learning and deep learning models, we initiated preprocessing steps to optimize the information. Our data set includes five distinct features, namely, "land_cover", "slope", "elevation", "solrad_annual" and "min_distance" with 971 instances. Among these, the "min_distance" feature exhibited two instances of missing data. To handle missing value, a common imputation method was employed, utilizing the mean strategy. This strategy involved replacing the missing values within the "min_distance" feature with the mean value derived from the available data points within that specific feature. As the "min_distance" feature represents continuous data, this approach was considered appropriate and helped mitigate the impact of missing values.

Additionally, to further enhance the data set's suitability for modeling, we performed standardization on certain features characterized by continuous data. Standardization involved a transformation process wherein the mean of each feature was subtracted, resulting in a data set where the mean of each feature became zero. Following this, scaling to unit variance was implemented, ensuring that the variance of each feature reached a uniform scale. This process aimed to equalize the impact of different features on the models, promoting fair comparisons and preventing certain features from exerting undue influence due to their scale differences. Furthermore, standardization can often enhance the performance of certain machine learning algorithms and may positively influence computational speed.

B. Models

Our approach involved deploying an ensemble of machine learning and deep learning models for two primary reasons. Firstly, it aimed to familiarize us with a spectrum of algorithms, ranging from simple to intricate ones. This diverse exposure provided comprehensive experience across different model complexities. Secondly, in case individual models didn't perform optimally, leveraging a voting method over an ensemble of weak learner classifiers could aggregate their outputs, potentially creating a stronger learner classifier with enhanced accuracy. This ensemble strategy provided a safety net, harnessing combined insights from multiple models to enhance predictive power.

Our initial choice was Logistic Regression (LR) due to its linearity and interpretability, offering transparency in showcasing feature influences on the outcome. LR's interpretability helped understand linear relationships between features and the target variable, yielding valuable insights into initial predictive patterns.

Transitioning beyond LR, we incorporated Support Vector Machines (SVM) and Decision Trees (DT) into our strategy. Both offer advantages over LR by detecting non-linear relationships and complex feature interactions. SVM excels in identifying non-linear boundaries by mapping data into higher-dimensional spaces, while DT capture complex decision-making processes via branching conditions.

Additionally, we implemented XGBoost models, leveraging ensemble approaches that combine weak learner models, primarily decision trees. This sequential improvement aims to correct errors from previous models using boosting algorithms. XGBoost reduces overall error in the loss function by fitting subsequent trees to ensemble residuals. Finally, a simple neural network (NN) with only two layers—a hidden layer with 16 neurons and an output layer with one neuron—was implemented using the Adam optimizer and binary cross-entropy loss function.

C. Results

After the model selection phase, a comprehensive evaluation of the model's effectiveness and robustness is vital. This evaluation process involves employing a 5-fold cross-validation technique on the training set. This approach partitions the training data into five equally sized subsets. It iteratively uses four subsets for training the model and reserves the remaining subset for validation. This cyclic process ensures each data subset acts as both training and validation data, providing a more reliable estimation of the model's generalization performance.

Dataset	Logistic Regression	Decision Tree	Support Vector Machine	XGBoost	Neural Network
Training Set (Accuracy)	94.9%	93.2%	95.4%	96.3%	95.4%
Test Set (Accuracy)	93.8%	94.8%	95.8%	95.3%	94.8%

TABLE I. Performance of logistic regression, decision tree, support vector machine, XGBoost and neural network models for training an test sets.

Hyperparameters play a pivotal role in a model's performance. To optimize these crucial settings and enhance model performance, a grid search technique is employed. This method systematically explores various combinations of hyperparameters within predefined ranges. By exhaustively testing these combinations, the grid search identifies the parameter set that yields the best performance based on the chosen evaluation metric.

In this context, the data set displays a balanced distribution in the target label 'Solar_farm_present.' Given this balance, accuracy serves as a suitable metric for evaluating model performance. Accuracy measures the ratio of correctly predicted instances to the total number of instances, making it particularly relevant when dealing with balanced classes. However, in scenarios involving imbalanced data sets, metrics like precision, recall, or F1-score might provide a more comprehensive evaluation of model performance. In this study, due to the balanced nature of the data set, these additional metrics are not necessary for assessing model performance.

Taken all into consideration, the results are depicted in Table I. Notably, all models exhibit similar accuracy levels on both the training and test sets. This consistency across models suggests a potential absence of substantial non-linear behavior among the features in the current dataset, given the existing number of instances and features. However, SVM model demonstrates slightly superior performance compared to others, indicating a subtle presence of non-linearity. This indication might intensify with the inclusion of more data or additional features.

Moreover, we generated confusion matrices for all five models, representing both training and test sets, as shown in Figs. 1 to 5. In these plots, the x-axis signifies the predicted label, while the y-axis represents the true label. Here, label zero signifies the absence of a solar farm, while label one denotes the presence of a solar farm in the respective location. Observing these plots reveals a significant concentration along the diagonal of the confusion matrix, indicating the model's ability to accurately predict labels, whether they are zeros or ones. In the off-diagonal, the confusion matrix shows cases that the model confuse the label. The plots reveal a subtle observation: the instances where the model predicts the existence of a solar farm in locations where the true label was zero outnumber the cases where the model predicts the absence of a solar farm while the true label was one.

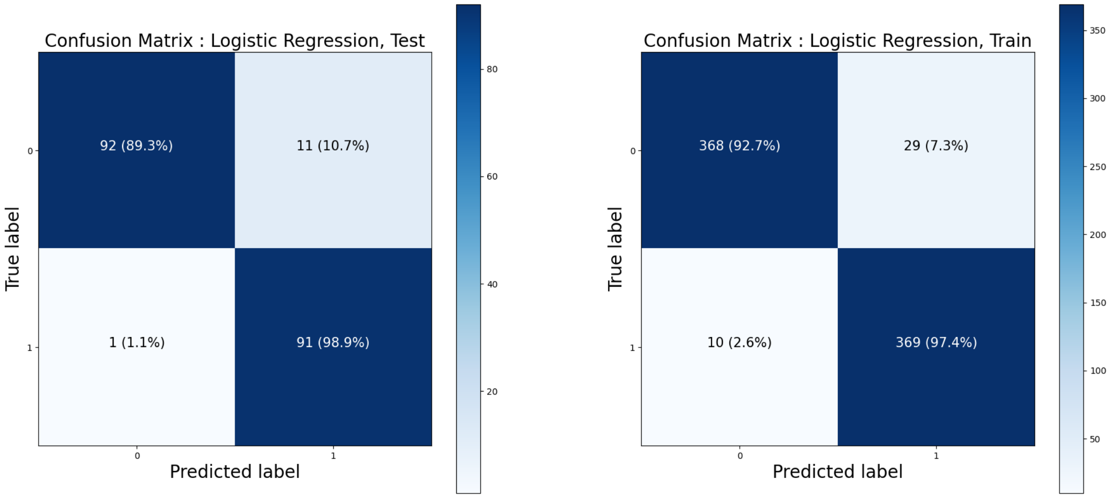

FIG. 1. Confusion matrix for Logistic Regression model for training set (left) and test set (right).

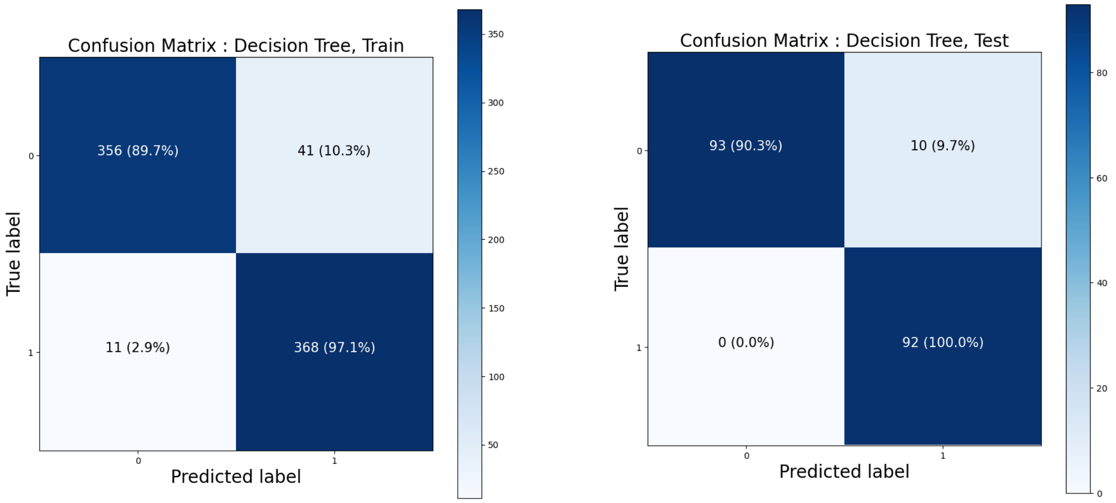

FIG. 2. Confusion matrix for decision tree model for training set (left) and test set (right).

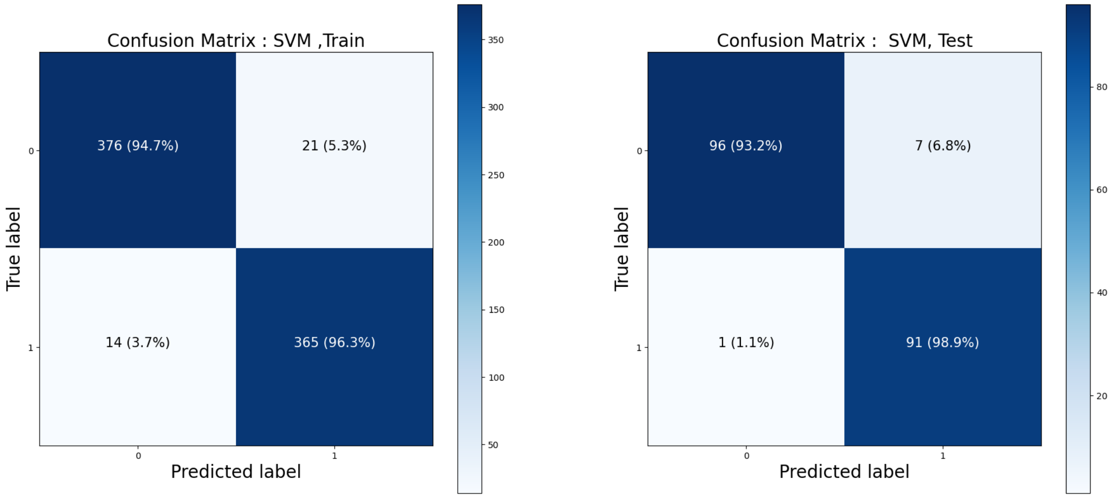

FIG. 3. Confusion matrix for support vector machine model for training set (left) and test set (right).

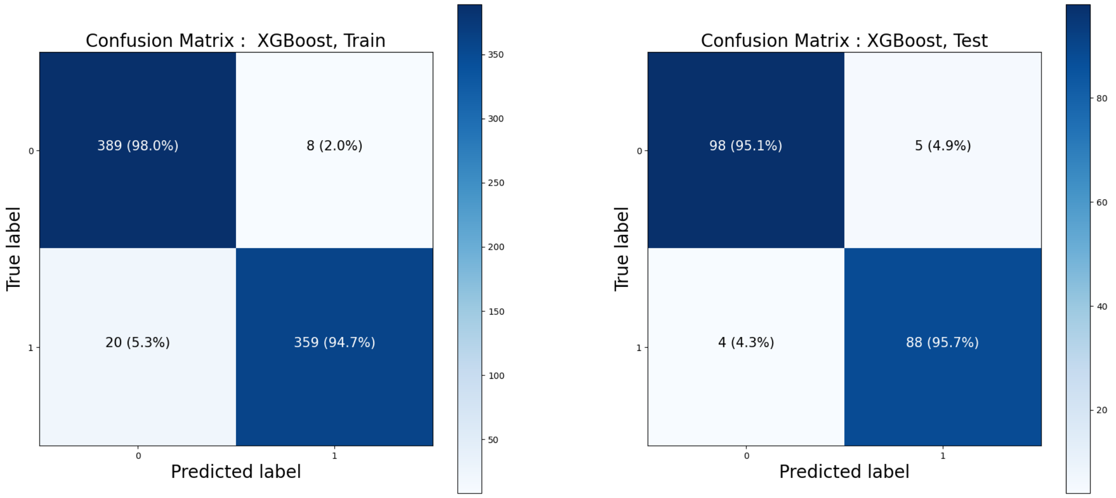

FIG. 4. Confusion matrix for XGBoost model for training set (left) and test set (right).

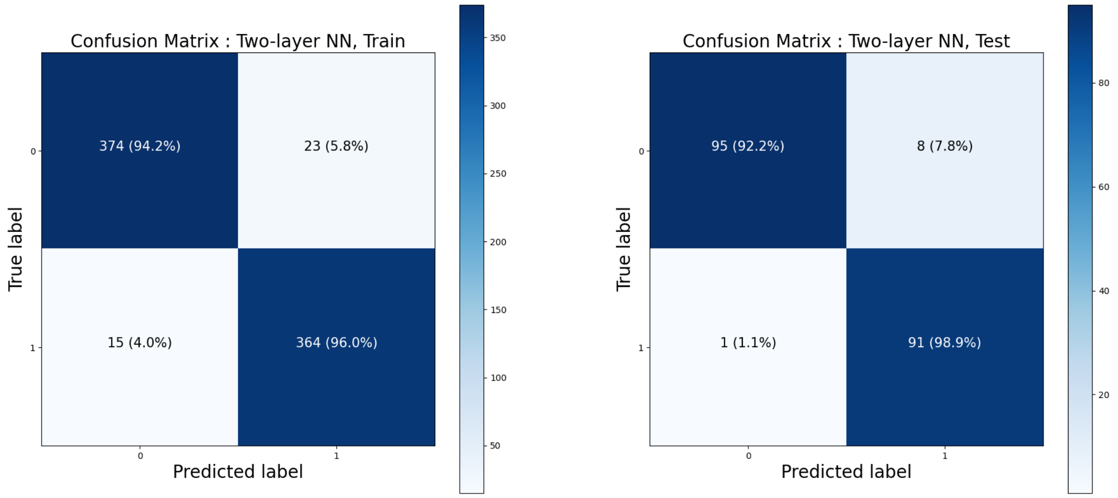

FIG. 5. Confusion matrix for two layer neural network model for training set (left) and test set (right).