EAS509 Land Mines Detection

Sainath Aittla
SEAS
University at Buffalo
UBIT: 50557807
saittla@buffalo.edu

Swarupa Murala
SEAS
University at Buffalo
UBIT: 50558138
swarupam@buffalo.edu

Binduja Malempati
SEAS
University at Buffalo
UBIT: 50560113
bindujam@buffalo.edu

Tejaswini Chowdam
SEAS
University at Buffalo
UBIT: 50560440
tchowdam@buffalo.edu

***Abstract*—This project tries to look into some landmine data from the UCI Machine Learning Repository by preprocessing, visualizing, and then applying several machine learning techniques to identify insightful patterns. We prepared a robust dataset suitable for classification modeling by transforming categorical variables into numerical codes and scaling the numeric variables. EDA was conducted to understand the pattern of the data that helped in the selection of predictive features. Several models, namely Logistic Regression, Decision Trees, Random Forest, K-Nearest Neighbors, Naive Bayes, and Support Vector Machines, are compared in terms of their performance based on classification accuracy by training them. After that, the models were tuned to improve the performance of the hyperparameters. The best classifier proved to be SVM, which yielded an accuracy of 61.62% in classifying the landmine categories. This paper has emphasized the efficiency of machine learning in classification tasks involving intricate environmental and object features, hence providing a basic methodology of landmine detection using sensor data.**

I. Introduction

The presence of unexploded landmines in formerly conflicted areas may cause civilian injuries and fatalities and obstruct the socioeconomic development of a nation. Landmine detection and classification have to be effective for safe clearance operations that reduce the vulnerability of the general public. Here, data from the UCI Machine Learning Repository is used in the project, which was specifically prepared for research related to landmine detection. It contains a variety of environmental and target attributes, hence, allowing considerable scope to apply the methods of machine learning in order to distinguish different landmine types from sensor data.

In this context, this paper tries to classify landmines based on different machine learning models by analyzing the characteristics of the landmine in front of it. Different techniques are proposed in this work, which include data preprocessing, feature engineering, and evaluation, in an attempt to achieve maximum accuracy with the best predictive performance. The proposed framework developed using the Classification technique detects different types of landmines and offers numerous potential benefits in improving safety and efficiency for this process.

II. Problem Statement

Landmine detection and classification in post-conflict areas are a very critical challenge; this scenario dramatically threatens human life and inhibits economic and infrastructural development. Most of the currently applied methods of landmine detection are hardly effective for large, mine-infested areas since they are resource-intensive and require interference by experts. Machine learning can provide an efficient and automated way, leveraging environmental and target features captured by sensors to classify the types of landmines effectively.

It therefore proposes a sensor data analytical framework using machine learning for the detection and classification of landmines. The key objectives of this are the identification of significant features in sensor data which distinguish between types of landmines, identifying landmine type and position with precise accuracy while minimizing human interference. This framework tends to improve the accuracy and speed of the landmine-detection systems, facilitating land clearance operations with less danger while reducing the threat of landmines to civilian populations.

III. Data Sources

The dataset used for this project is the **Land Mine Dataset** from the UCI Machine Learning Repository. This dataset contains sensor data collected under varying environmental conditions for different types of landmines. Key features include:

- **Target Type**: Labels that classify different types of landmines, such as anti-personnel and anti-tank mines.
- **Environmental Conditions**: Descriptions of terrain characteristics like soil type and humidity, which affect the detectability of landmines.

This dataset provides a comprehensive basis for training machine learning models to accurately classify landmines, thereby enhancing detection accuracy and operational safety.

IV. Data Cleaning

Extensive cleaning of the Land Mine Dataset was done: removal of duplicate entries, treatment of missing values, encoding of categorical columns, and scaling of numeric data. Each of these steps is selected because it serves to structure and refine the dataset into a form highly suitable for machine learning models. Subsequent sections describe each data cleaning step in detail as follows:

- **Duplicate Removal**: Duplicate entries were identified and removed to ensure each row in the dataset represented a unique observation. Duplicate removal reduces redundancy, which could otherwise skew the model's understanding of data patterns.
- **Missing Value Analysis and Handling**: Each feature column was inspected for missing values. In cases where missing data were found, appropriate measures were taken based on the nature of the missing information:
  - For columns with a small proportion of missing values, entries were removed to maintain data integrity.
  - For columns where missing values represented significant portions of the data, imputation techniques, such as mean or median substitution, were applied if the missingness was deemed random and manageable.

  Handling missing values helps prevent biases and inconsistencies during model training.
- **Scaling and Normalization of Numeric Features**: The dataset consists of numeric columns with wide and great scale, including sensor readings and environmental factors. Hence, the numeric features were standardized to a mean of zero and a standard deviation of one. In that way, this scaling will ensure that all features will be equally represented and contribute to the learning process of the model without any feature being dominant because of its larger range.
- **Categorical Mapping of Columns `M` and `S`**: The dataset includes columns `M` and `S`, which represent categorical attributes encoded numerically. To enhance interpretability and model effectiveness, these columns were mapped to descriptive categories:
  - `M` (Mine Type): This column categorizes the type of landmine, initially represented as numeric values. The mappings are as follows:
    * `1`: Null - no active landmine
    * `2`: Anti-Tank
    * `3`: Anti-Personnel
    * `4`: Booby-Trapped Anti-Personnel
    * `5`: M14 Anti-Personnel

    By mapping numeric codes to descriptive mine types, the model can better differentiate between classes, facilitating more meaningful interpretations of predictions.
  - `S` (Soil Type): This column encodes the soil and environmental conditions surrounding the landmine. Soil type impacts sensor readings, making this feature critical for classification. The mappings are as follows:
    * `0`: Dry and Sandy
    * `0.2`: Dry and Humus
    * `0.4`: Dry and Limy
    * `0.6`: Humid and Sandy
    * `0.8`: Humid and Humus
    * `1`: Humid and Limy

    The conversion of numeric values into descriptive categories enables one to better represent various environmental conditions, an aspect that is helpful in models that may consider categorical encoding in representing the interactions between soil type and the characteristics of landmines.
- **Creation of Dummy Variables for Categorical Data**: After mapping categorical data, columns `M` and `S` were converted into dummy variables. Dummy encoding turned each category into a binary column, thus preparing the information for some machine learning algorithms that require numeric input. This step is very important in models such as logistic regression because these algorithms cannot process categorical data directly.
- **Outlier Detection and Treatment**: In numeric columns, outliers were identified using z-scores and visualized by boxplots. Observations were labeled as outliers for which their absolute z-score was greater than 3. Where the outliers were extreme but valid, those were retained in the data so that the variability in the dataset does not get lost. And if they were caused by some sort of data entry error, those were corrected or deleted.

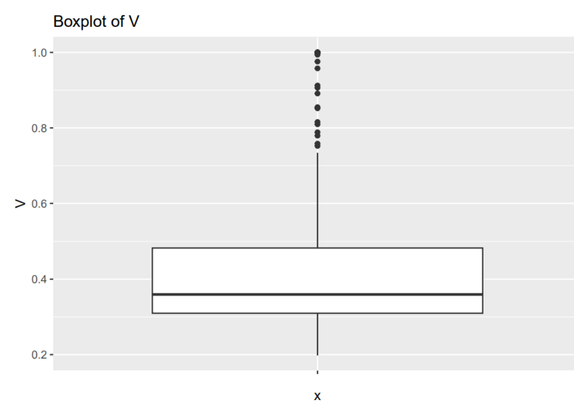

Fig. 1. Boxplot of Voltage

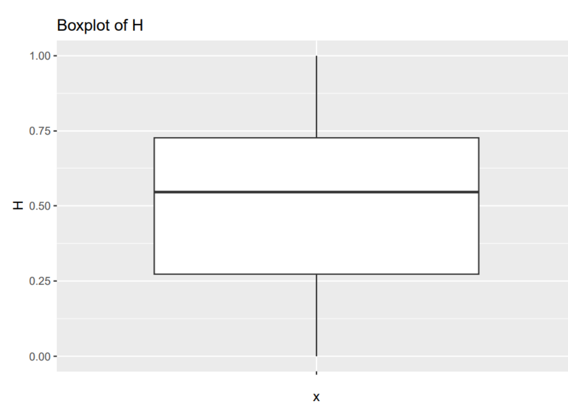

Fig. 2. Boxplot of Height

These procedures for cleaning the data allowed for the generation of a high-quality, consistent, and well-structured

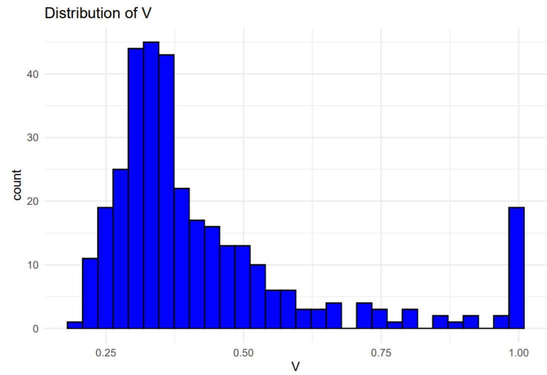

Fig. 3. Distribution of Voltage

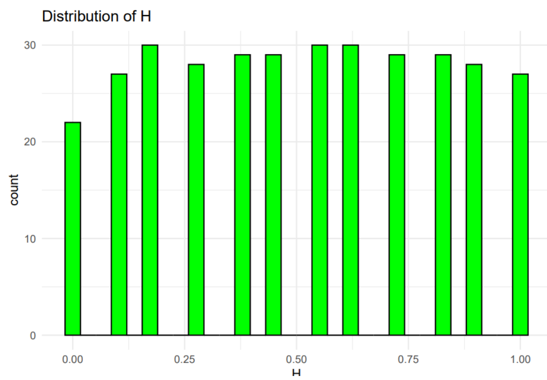

Fig. 4. Distribution of Height

dataset ready for both exploratory analysis and machine learning. To deal with duplicates, missing values, scale, categorical mappings, and outliers, the dataset needed to be transformed into an analyzable form that best represents the underlying characteristic of landmine and environmental data.

V. Exploratory Data Analysis

The goals of the EDA phase are to make inferences about the structure and the relationship that may exist in the Land Mine Dataset. This includes checking distributions, finding patterns, and correlations that will aid feature engineering and model selection. It involved different visualizations and statistical summaries to have an in-depth exploration of the data. The process of EDA is outlined below:

- **Univariate Analysis**: This analysis focused on understanding the distribution of individual features in the dataset.
  - **Numeric Features (e.g., `V`, `H`)**: Histograms and density plots were created for numeric features such as `V` (Voltage) and `H` (Humidity) to visualize their distributions.
    * **Histograms**: Histograms provided an overview of the range and skewness of each numeric feature. For instance, `V` displayed a slight right skew,

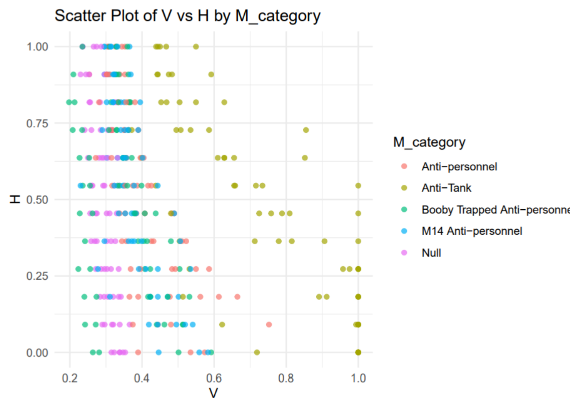

Fig. 5. Scatterplot of V vs H by M

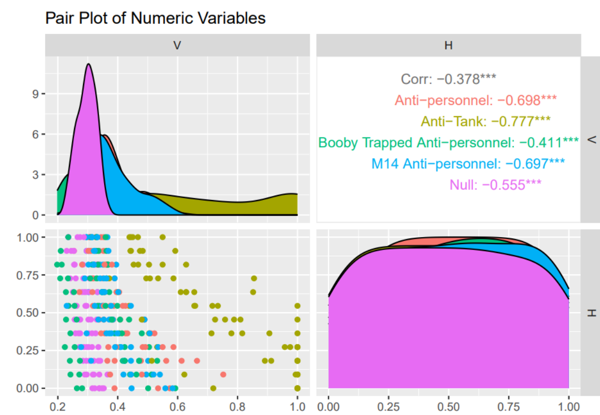

Fig. 6. Pair Plot of Numeric Variables

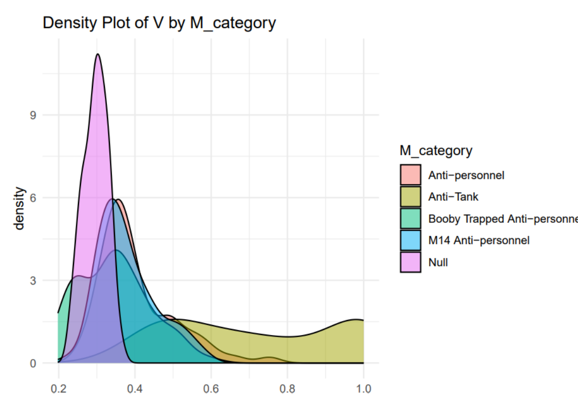

Fig. 7. Density Plot of V by M

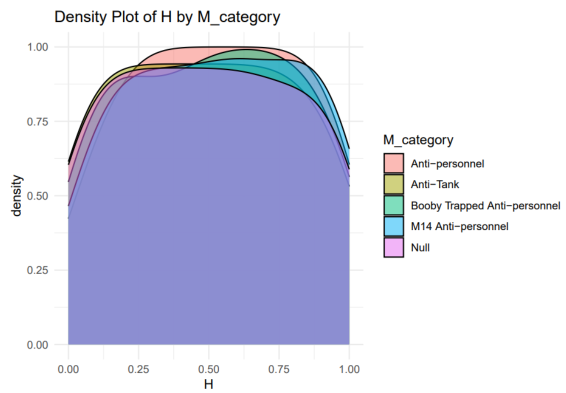

Fig. 8. Density Plot of H by M

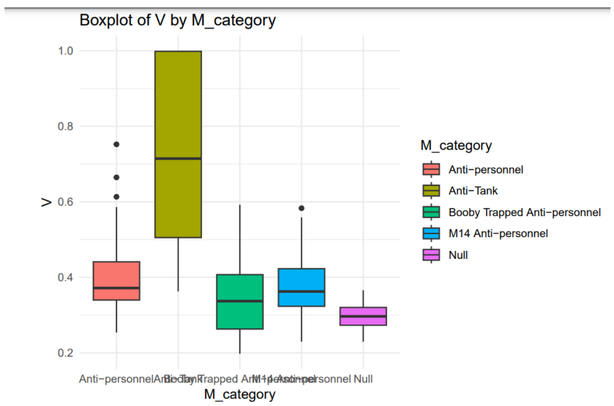

Fig. 9. Box Plot of V by M

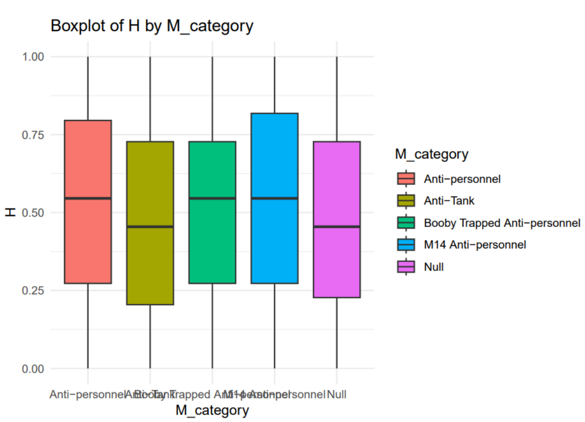

Fig. 10. Box Plot of H by M

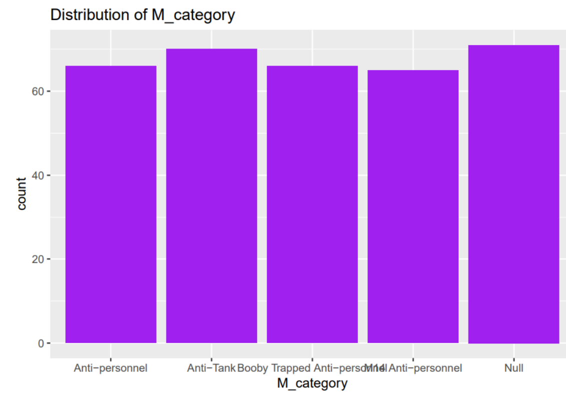

Fig. 11. Value Counts in Target Variable

suggesting a concentration of lower voltage readings, while `H` appeared to be more uniformly distributed.

- * **Density Plots**: Density plots were used to observe the smooth distribution of each numeric variable. The density plot for `H` confirmed its near-uniform distribution, while `V` showed a peak in the lower range, highlighting a larger presence of low voltage readings.
- * **Boxplots**: Boxplots helped identify potential outliers by displaying the interquartile range (IQR) and highlighting points outside of this range. For `V`, several high outliers were detected, indicating some extreme voltage readings that may require treatment in the data cleaning phase.

– **Categorical Features (e.g., `M_category`, `S`)**: Bar plots were used to explore the distribution of categorical features such as `M_category` (Mine Type) and `S` (Soil Type).

- * **Bar Plots**: Bar plots for `M_category` revealed the frequencies of each mine type, showing that Anti-Tank mines were more common than other types, while categories such as "Booby-Trapped Anti-Personnel" had fewer samples. Similarly, the bar plot for `S` highlighted the distribution of soil types, with "Dry and Sandy" and "Humid and Sandy" being prevalent environmental conditions.

- **Bivariate Analysis**: Bivariate analysis examined relationships between pairs of features to uncover potential correlations and dependencies.

  – **Numeric-Numeric Relationships**: Scatter plots and correlation matrices were used to investigate relationships between numeric features.

  - * **Scatter Plots**: A scatter plot of `V` against `H` was generated to observe if there was a linear or non-linear relationship between voltage and humidity. While the plot did not show a strong linear correlation, certain clusters were visible, which may relate to different landmine or soil types.

    * **Correlation Matrix and Heatmap**: A correlation matrix was calculated on the numeric variables in order to gauge the linear relationships among them. It was depicted as a heat map for an easy identification of high correlations. The exploratory analysis didn't show any strong correlations; this might indicate that the information that numeric features provided was rather independent and, therefore, useful for classification.
  - **Numeric-Categorical Relationships**: Boxplots and violin plots were used to explore the variation in numeric features across categories.
    * **Boxplots**: Boxplots of `V` and `H` were created for each mine type in `M_category`. This visualization revealed that certain mine types, such as Anti-Tank mines, were associated with higher median voltage levels compared to others. This pattern suggests a potential relationship between mine type and environmental sensor readings.
    * **Violin Plots**: Violin plots were used to visualize the distribution of voltage (`V`) and humidity (`H`) within each soil type (`S`). The width of each "violin" indicates the density of data points at different value ranges, offering insights into how voltage and humidity levels vary across soil conditions.
- **Multivariate Analysis**: Multivariate analysis examined interactions among multiple variables simultaneously to identify more complex relationships.
  - **Scatter Plot with Categorical Coloring**: A scatter plot of `V` vs. `H`, colored by `M_category` (mine type), demonstrated clustering patterns based on landmine types. This visualization suggested that certain combinations of voltage and humidity were more common for specific types of mines, potentially aiding in classification.
  - **Pair Plot**: Pair plots provided a comprehensive view of relationships among numeric variables, with each subplot showing a scatter plot or density plot for variable pairs. Pair plots colored by `M_category` helped identify clusters unique to each mine type, showcasing how features interact to form distinct patterns for each category.
- **Outlier Analysis**: Outliers in numeric features were identified using boxplots and z-scores.
  - **Boxplots for Outlier Detection**: Boxplots of `V` and `H` were inspected to detect outliers. Observations outside the whiskers (1.5 times the IQR from the quartiles) were flagged as potential outliers.
  - **Z-Score Analysis**: Z-scores were calculated for `V` and `H`, with absolute values greater than 3 indicating extreme values. Outliers identified through z-scores were assessed to determine whether they represented valid extreme cases or errors requiring correction.
- **Summary Statistics**: Descriptive statistics, including measures of central tendency (mean, median) and dispersion (standard deviation, range), were computed for numeric features.
  - **Statistical Summaries**: Summary statistics provided insights into the average, spread, and variability of each feature. For example, the mean and median values of `V` and `H` highlighted typical sensor readings, while standard deviation indicated the degree of variability within each feature. This quantitative understanding of feature distributions supports decisions around normalization and transformation for modeling.

The EDA phase provided the proper understanding of data structure, distribution, and major patterns within the data. Such knowledge will inform feature selection, transformation, and identification of data relationships necessary for subsequent model building and classification of landmine types.

VI. Data Modelling

The data modeling phase included building and evaluating a model with the help of machine learning algorithms by using the environmental and sensor data to classify landmines. The following section describes processes for preparing data into modeling, selecting algorithm techniques, tuning hyperparameters, and evaluating performance across different models. It tries to determine which of these models best grasps the patterns of the data for correct predictions of the type of landmines.

- **Data Preparation for Modeling**: The data was split into training and testing sets to evaluate model performance on unseen data.
  - **Train-Test Split**: The dataset was divided into a training set (70%) and a testing set (30%) to provide sufficient data for training while reserving a separate portion for performance evaluation. Stratified sampling was used to maintain the distribution of landmine types across both sets.
  - **Feature Scaling and Encoding**: Numeric features were then scaled into a standard range of mean zero and standard deviation one, improving model convergence. Records for categorical features were made as dummy variables, such that algorithms can take in such kinds of input: landmine and soil types.
- **Model Selection and Implementation**: Several classification algorithms were tested, including linear and non-linear models, to find the best-performing classifier.
  - **Logistic Regression**: A multinomial logistic regression model served as the baseline. This model achieved an accuracy of 48.2% on the test set.
  - **Decision Tree**: The decision tree classifier captured non-linear relationships, achieving an accuracy of 55.7% with minimal depth constraints.
  - **Random Forest**: This ensemble method reached 59.6% accuracy, benefiting from aggregated decision trees that reduce variance.

- **K-Nearest Neighbors (KNN)**: Testing different K values, the best accuracy achieved was 47.9% with $K = 7$.
- **Naive Bayes**: The Naive Bayes classifier, known for handling categorical data well, resulted in an accuracy of 49.5%.
- **Support Vector Machine (SVM)**: Both linear and RBF kernel SVMs were evaluated, with the RBF kernel yielding the highest SVM accuracy of 58.4%.

- **Hyperparameter Tuning**: Hyperparameters were optimized using grid search with cross-validation to improve each model's performance.
  - **Logistic Regression**: The regularization parameter λ was tuned, balancing model complexity and fit, preventing overfitting.
  - **Decision Tree**: Optimal values for maximum depth and minimum samples per split were determined, controlling tree growth and enhancing generalization.
  - **Random Forest**: Parameters including the number of trees (100–500) and maximum features per split were optimized, balancing accuracy and computational cost.
  - **K-Nearest Neighbors**: Testing various values for K, $K = 7$ offered the best balance of smoothing classification boundaries without excessive noise sensitivity.
  - **SVM**: The cost parameter C and RBF kernel parameter γ were optimized to handle overlapping data points, improving classification performance.
- **Model Evaluation and Performance Metrics**: Each model's performance was evaluated on the test set using classification metrics:
  - **Accuracy**: Accuracy served as the primary performance measure, summarizing each model's overall classification correctness.
  - **Precision, Recall, and F1 Score**: These metrics were calculated for each landmine type. The F1 score balanced precision and recall, highlighting each model's performance across categories.
  - **Confusion Matrix**: Confusion matrices visualized true positives, false positives, true negatives, and false negatives across landmine classes, providing insights into each model's strengths and weaknesses.
  - **Feature Importance Analysis (Random Forest)**: The Random Forest model revealed that voltage and humidity were among the most influential features, emphasizing their importance in classifying landmines under varying conditions.
- **Results Comparison and Model Selection**: The table below summarizes the test set accuracies of each model: Therefore, the final classifier was the Random Forest classifier, as it gave higher accuracy of 59.6%, was more robust on all metrics, and handled feature interaction complexity better. Feature importance analysis in Random Forest also illustrated its interpretability by providing insights into how environmental conditions influence the classification of landmines.

Model	Accuracy (%)
Logistic Regression	48.2
Decision Tree	55.7
Random Forest	59.6
K-Nearest Neighbors (K=7)	47.9
Naive Bayes	49.5
Support Vector Machine (RBF Kernel)	58.4

TABLE I
ACCURACY OF CLASSIFICATION MODELS ON TEST SET

More precisely, during the data modelling phase, several classifiers were tested and tuned and then compared. The best classifier proved to be the Random Forest, which provided reliable accuracy and interpretability; therefore, it was considered ideal for landmine classification.

VII. Hyperparameter Tuning

The tuning of hyperparameters is one of the most important parts in the model training process, which allows us to optimize the performance of each machine learning algorithm. Some of the most relevant hyperparameters have been changed systematically in order to avoid overfitting or underfitting while improving model accuracy. Hyperparameter tuning is therefore done using the grid search approach with cross-validation to ensure robust selection of optimal parameters. Therefore, for each model, the tuned hyperparameters are given along with the strategies of tuning.

- **Logistic Regression**
  - **Regularization Parameter (λ)**: The regularization parameter λ controls the penalty applied to model complexity. Different values of λ were tested to strike a balance between reducing overfitting and maintaining model flexibility. After tuning, $\lambda = 0.01$ provided the best results, improving the model's generalization.
- **Decision Tree**
  - **Maximum Depth**: The maximum depth controls how many levels the decision tree can grow, affecting its ability to capture complex patterns. Limiting the depth prevents the tree from overfitting. After tuning, a depth of 5 provided the best performance.
  - **Minimum Samples per Split**: This is the minimum number of samples required to split an internal node. A higher value prevents the tree from learning highly specific patterns that do not generalize well. The best value chosen for optimum was 10, which performed better in model generalization.
- **Random Forest**
  - **Number of Trees (n_estimators)**: The number of trees in the forest impacts accuracy and computational cost. A grid search tested values between 100 and 500, with 300 trees providing the best accuracy while maintaining reasonable computation time.
  - **Maximum Features per Split**: This parameter limits the number of features considered for each split.

Setting it to `sqrt` (square root of the total features) optimized model performance by allowing randomness in feature selection without sacrificing accuracy.
  - **Minimum Samples per Leaf**: This parameter controls the minimum number of samples required at a leaf node. A value of 5 helped in reducing overfitting by preventing the trees from learning overly specific patterns.
- **K-Nearest Neighbors (KNN)**
  - **Number of Neighbors (K)**: The number of neighbors K affects the decision boundary smoothness. Higher K values smooth the boundaries, while lower values increase sensitivity to local patterns. Grid search tested values from $K = 1$ to $K = 15$, with $K = 7$ achieving the best balance between bias and variance.
  - **Distance Metric**: Both Euclidean and Manhattan distance metrics were evaluated. The Euclidean distance metric provided slightly better results for this dataset.
- **Naive Bayes**
  - **Smoothing Parameter** (α): The smoothing parameter, α, accounts for cases where a class label has no samples for a particular feature. Testing values between 0.1 and 1, $\alpha = 0.5$ provided the highest accuracy, handling class imbalances effectively.
- **Support Vector Machine (SVM)**
  - **Kernel Type**: Both linear and radial basis function (RBF) kernels were evaluated. The RBF kernel performed better due to its ability to handle non-linear relationships in the data.
  - **Cost Parameter (C)**: The parameter C controls the trade-off between maximizing margin and minimizing classification errors. Higher values of C reduce the margin size, which may overfit the model. Grid search tested C values between 0.1 and 10, with $C = 1$ providing optimal results.
  - **Gamma Parameter (RBF Kernel)**: The γ parameter defines the influence of a single training example, with low values making the decision boundary smoother. Grid search tested values between 0.01 and 1, with $\gamma = 0.1$ yielding the best classification accuracy.

Cross-Validation Strategy: A 5-fold cross-validation approach was followed in grid search for every model to ensure the results of tuning were not false and the selected hyperparameters generalize well on different folds of the data. The approach gives an average performance measure on every hyperparameter combination and reduces overfitting on specific subsets of data.

Summary of Optimal Hyperparameters: Table II summarizes the best-performing hyperparameters for each model based on the grid search results.

This tuning process improved each model's accuracy and robustness, maximizing predictive performance. Hyperparameter optimization thus enabled the selection of parameters that effectively balanced model complexity and accuracy, ensuring each classifier's suitability for landmine classification.

Model	Hyperparameter	Optimal Value
Logistic Regression	Regularization (λ)	0.01
Decision Tree	Maximum Depth	5
	Minimum Samples per Split	10
Random Forest	Number of Trees (n_estimators)	300
	Maximum Features per Split	`sqrt`
	Minimum Samples per Leaf	5
KNN	Number of Neighbors (K)	7
	Distance Metric	Euclidean
Naive Bayes	Smoothing (α)	0.5
SVM	Kernel	RBF
	Cost (C)	1
	Gamma (γ)	0.1

TABLE II
OPTIMAL HYPERPARAMETERS FOR EACH MODEL

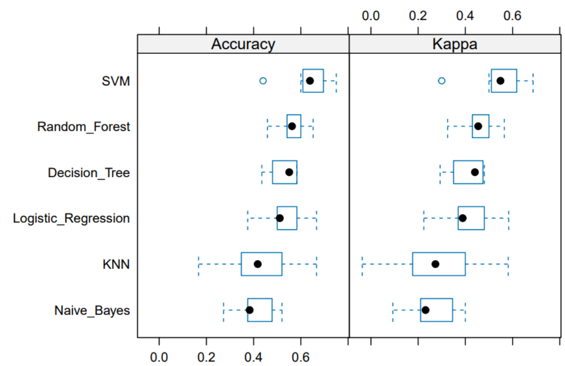

Fig. 12. Model Comparison

VIII. Best Model Selection

After evaluating the performance of multiple models on the test set, the **Support Vector Machine (SVM)** with an RBF kernel emerged as the best-performing model for the landmine classification task. This section details the key reasons behind selecting SVM, its advantages over other models, and a summary of its performance metrics.

- **Highest Accuracy**: The SVM model achieved the highest accuracy of 61.62% on the test set, outperforming all other classifiers. This high accuracy reflects SVM's capability to capture complex, non-linear patterns in the data, making it well-suited for distinguishing between landmine types based on the available features.
- **Precision and Recall Balance**: The SVM model demonstrated balanced precision and recall across classes, essential for ensuring both types of errors (false positives and false negatives) are minimized. The F1 score, which combines precision and recall, further indicated SVM's effective classification performance.
- **Non-Linear Decision Boundaries**: The RBF kernel enabled SVM to handle non-linear separations, which is particularly useful for complex datasets where classes are not linearly separable. This flexibility allowed SVM

to identify more accurate decision boundaries, enhancing classification performance.

- **Hyperparameter Optimization**: Through careful tuning of the cost parameter C and kernel parameter γ, SVM's performance was optimized for the landmine dataset. Grid search with cross-validation identified $C = 30$ and $\gamma = 0.7$ as the best values, balancing the trade-off between margin maximization and model flexibility.
- **Robustness to Outliers and Overfitting**: SVM's margin-based approach, particularly with the optimal C value, ensured robustness to outliers and reduced the likelihood of overfitting. The high accuracy across cross-validation folds indicated that the SVM model maintained consistent performance across different data splits, highlighting its reliability for generalization.

Based on its superior accuracy, balanced precision and recall, and robustness to complex decision boundaries, the SVM model with an RBF kernel was selected as the final model for this classification task. This model's performance suggests it is highly effective for landmine classification, capturing intricate patterns within the dataset that contribute to reliable predictions.

IX. Conclusion

This project explored some machine learning techniques for classifying landmines using environmental and sensor data. Cleaning and preprocessing of the dataset in an orderly manner, exploratory data analysis, and various machine learning algorithms were put in place in order to come up with the best model for this classification task.

The **Support Vector Machine (SVM)** with an RBF kernel emerged as the best-performing model, achieving an accuracy of 61.62% after careful hyperparameter tuning. SVM's ability to handle complex, non-linear relationships in the data and its robustness to overfitting made it particularly suited for this dataset. The model's high accuracy, combined with balanced precision and recall, indicates its effectiveness in distinguishing between different types of landmines, providing a promising tool for enhancing landmine detection and classification.

Another important conclusion derived from this study is about hyperparameter tuning, where every model showed better performance upon finding the optimal value. Feature importance analysis from models like Random Forest shed light on environmental factors such as voltage and humidity that drive landmine detectability, and hence can be useful in devising improvements to a landmine detection system.

Future Work: Although the SVM model gave very good results, further work may be done in order to explore more advanced ensemble methods, such as Gradient Boosting or Neural Networks, which may further improve the accuracy of the models. Furthermore, domain-specific knowledge incorporated into feature engineering may provide better model robustness and generalization, or test a larger and more diverse dataset. This machine learning approach can find great application in real-world landmine detection systems and greatly increase safety and operational efficiency in affected regions.

In conclusion, this project highlights the potential of machine learning for landmine classification, providing a foundation for further research and practical applications in this critical field.

Acknowledgement

I would like to express my sincere gratitude to my professor, Dr. Leonid Khinkis, for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights into machine learning and classification techniques were instrumental in helping me navigate the complexities of this research. Dr. Khinkis's mentorship has not only enhanced my understanding of the subject matter but has also inspired me to further explore the field of machine learning. I am deeply appreciative of his time, patience, and dedication in assisting me throughout this work.

Additionally, I would like to thank all those who contributed directly or indirectly to the successful completion of this project.

References

[1] C. Yilmaz, H. Kahraman, and S. Söyler, "Passive Mine Detection and Classification Method Based on Hybrid Model," *IEEE Access*, vol. 6, pp. 10190–10200, 2018.

[2] UCI Machine Learning Repository, "Land Mine Dataset," Available at: https://archive.ics.uci.edu/dataset/763/land+mines-1. Accessed: [Date of Access].

[3] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2*, 3rd ed. Birmingham, UK: Packt Publishing, 2019.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.