Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 4, No. 2, May 2025, pp. 124-134
e-ISSN: 2964-2922, p-ISSN: 2963-6191
DOI : https://doi.org/10.69916/jkbti.v4i2.310

MACHINE LEARNING-BASED CLASSIFICATION OF SPACE TRAVEL ELIGIBILITY USING SUPPORT VECTOR MACHINE, RANDOM FOREST, AND XGBOOST

Teguh Rizali Zahroni[1], Bahtiar Imran[2], Muhammad Tahrir[*3], Muh. Akshar[4], Zahrotul Isti'anah Marroh[5]

[1,3,4,5]Rekayasa Kayu, Politeknik Pertanian Negeri Samarinda, Indonesia
[2]Rekayasa Sistem Komputer, Fakultas Teknologi Informasi dan Komunikasi, Universitas Teknologi Mataram, Indonesia
Email: [1]teguhrizalizahroni@gmail.com, [2]bahtiarimranlombok@gmail.com, [3]mtahrir26@politanisamarinda.ac.id, [4]muhammadin.akshar@gmail.com, [5]hamzahzhie@gmail.com

(Received: April 12, 2025; Revised: April 30, 2025; Accepted: May 5, 2025)

Abstract

This study applies machine learning classification techniques to predict passenger displacement events based on corrupted data retrieved from a hypothetical interstellar spacecraft mission. Using a cleaned and preprocessed dataset containing demographic, behavioral, and exposure-related features, we compare the performance of three classification models: Random Forest, Support Vector Machine (SVM), and XGBoost. Each model is trained on 80% of the data and evaluated on the remaining 20% using training accuracy, cross-validation accuracy, and test accuracy metrics. After feature selection, SVM shows significant improvement, with test accuracy increasing from 49.97% to 72.40%, and cross-validation accuracy improving from 68.12% to 71.41%. Random Forest maintains consistent performance with test accuracy improving from 79.53% to 80.05%, and cross-validation accuracy at around 79.64%. XGBoost achieves the most stable results, with test accuracy rising from 79.76% to 79.93% and cross-validation accuracy from 79.15% to 79.31%. Feature importance analysis further enhances model interpretability, particularly in ensemble-based methods. The comparative analysis demonstrates that Random Forest and XGBoost are more effective in handling high-dimensional, partially incomplete datasets, making them suitable for complex predictive tasks in uncertain data environments.

Keywords: classification models, interdimensional displacement, machine learning, passenger classification, space anomaly prediction.

1. INTRODUCTION

With advancements in space exploration and theoretical physics, the prospect of interstellar travel has been a subject of extensive scientific inquiry[1]. By the 30th century, as envisioned by futurists and astrophysicists, humanity may have developed the capability to explore and potentially inhabit exoplanets beyond the solar system[2]. One of the conceptualized interstellar missions is the launch of *Spaceship*[3], a theoretical passenger liner designed to transport nearly 13,000 emigrants to three identified exoplanets within habitable zones.

A potential challenge in deep-space travel involves the interaction between spacecraft and unexplored astrophysical phenomena[4]. Hypothetically, while traversing near Alpha Centauri en route to 55 Cancri E, the vessel might encounter an unanticipated spacetime anomaly—akin to those postulated in general relativity and quantum gravity models—concealed within an interstellar dust cloud[5]. Such an event could theoretically lead to a high-energy perturbation affecting the fabric of spacetime, resulting in an unforeseen displacement of nearly half of the passengers into an alternate dimensional framework, if such dimensions exist as suggested by certain interpretations of string theory[6].

This study explores a machine learning-based approach to analyzing passenger data and identifying factors that may correlate with this hypothesized phenomenon. By leveraging classification models within data mining, we aim to assess the feasibility of predicting interdimensional transport events, drawing parallels to anomaly detection techniques in contemporary aerospace engineering and astrophysical research[7].

Identifying which passengers were affected by this incident poses a significant challenge due to the limited and corrupted data retrieved from the ship's damaged computer system. The complexity of interdimensional displacement necessitates a robust analytical approach, making machine learning classification models a viable solution. Data mining techniques can extract meaningful patterns from the available passenger records, enabling the prediction of which individuals were transported by the anomaly.

Previous studies have demonstrated the effectiveness of machine learning algorithms in predicting complex phenomena based on historical data. Research on predictive classification models has been widely applied in various fields, including medical diagnostics, financial risk assessment, and space mission anomaly detection. In[8], machine learning models such as Random Forest and Support Vector Machines (SVM) were successfully used to classify space mission failures with high accuracy. Similarly, in[9], Optimized boosting algorithms, such as XGBoost, were utilized to investigate the influence of cosmic radiation and solar activity on climatic patterns, yielding high-performance predictive results.

Several studies have explored predictive modeling in extreme environments. In [10], Artificial intelligence (AI)-based classification approaches were utilized in space medicine applications, though several challenges persist in ensuring their effectiveness under spaceflight conditions., with neural networks demonstrating superior accuracy compared to traditional statistical approaches. In another study [11], deep learning techniques were applied to classify gravitational wave anomalies. The effectiveness of machine learning in analyzing complex space-related datasets highlights its potential application in predicting interdimensional displacement events.

Despite the numerous classification models applied in previous research, no studies have specifically addressed the classification of passengers transported to alternate dimensions due to spacetime anomalies. Therefore, this study aims to classify affected passengers using data mining techniques, focusing on Random Forest, Support Vector Machines (SVM), and XGBoost models. These models were selected due to their strong predictive performance in handling high-dimensional data [12], their efficiency in decision-making, and their robustness in classification tasks [13].

The dataset used in this study consists of passenger records extracted from the Spaceship Titanic's damaged system, including features such as demographic details, cabin information, onboard activities, and anomalous energy exposure readings. The dataset is split into 80% training data and 20% testing data to evaluate model performance. The results of this study are expected to contribute to the development of predictive frameworks for space missions and enhance safety protocols for future interstellar travel.

2. METHODOLOGY

The methodology used in this research is illustrated in figure 1.

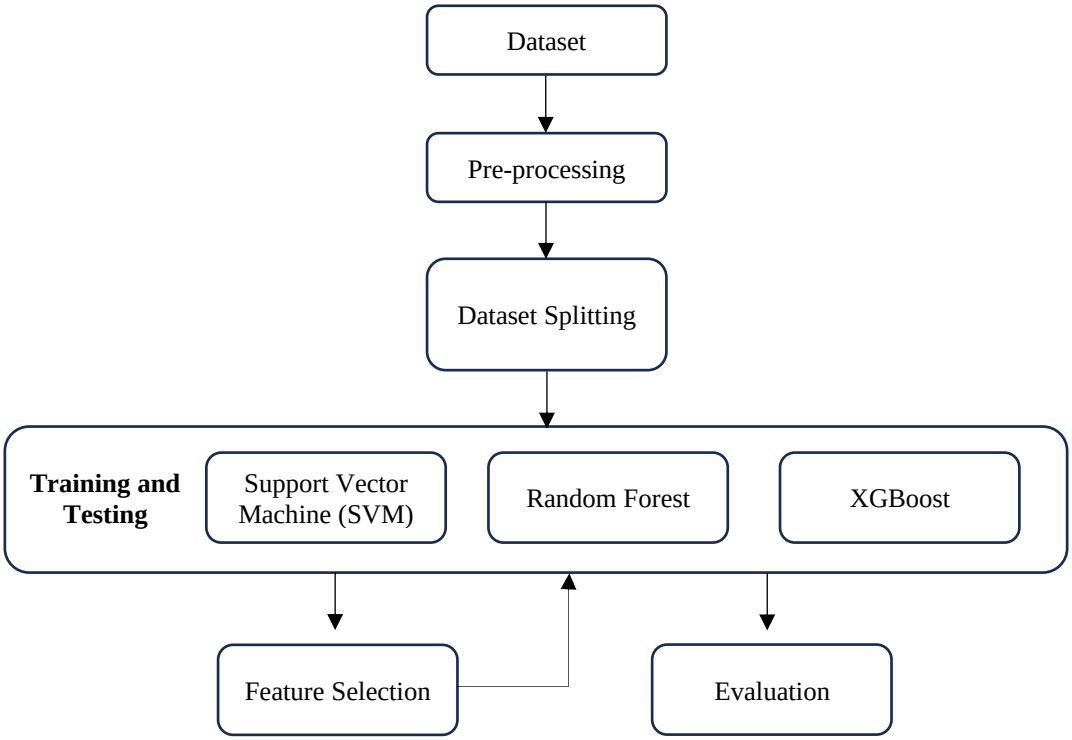

Figure 1. Research Flow

2.1. Dataset

The dataset for this study was sourced from a dataset-sharing platform, Kaggle.com, accessible via the link: https://www.kaggle.com/competitions/spaceship-titanic. The data is provided in CSV format and consists of 8,693 records with 14 attributes. An example of the dataset is presented in Figure 2, while the attributes utilized in this study are detailed in Table 1.

Table 1. The Attribute of Dataset

Number	Attribute
1	PassengerId
2	HomePlanet
3	CryoSleep
4	Cabin
5	Destination
6	Age
7	VIP
8	RoomService
9	FoodCourt
10	ShoppingMall
11	Spa
12	VRDeck
13	Name
14	Transported

	A	B	C	D	E	F	G	H	I	J
1	PassengerId,HomePlanet,CryoSleep,Cabin,Destination,Age,VIP,RoomService,FoodCourt,ShoppingMall,Spa,VRDeck,Name,Transported									
2	0001_01,Europa,False,B/0/P,TRAPPIST-1e,39.0,False,0.0,0.0,0.0,0.0,0.0,Maham Ofracculy,False									
3	0002_01,Earth,False,F/0/S,TRAPPIST-1e,24.0,False,109.0,9.0,25.0,549.0,44.0,Juanna Vines,True									
4	0003_01,Europa,False,A/0/S,TRAPPIST-1e,58.0,True,43.0,3576.0,0.0,6715.0,49.0,Altark Susent,False									
5	0003_02,Europa,False,A/0/S,TRAPPIST-1e,33.0,False,0.0,1283.0,371.0,3329.0,193.0,Solam Susent,False									
6	0004_01,Earth,False,F/1/S,TRAPPIST-1e,16.0,False,303.0,70.0,151.0,565.0,2.0,Willy Santantines,True									
7	0005_01,Earth,False,F/0/P,PSO J318.5-22,44.0,False,0.0,483.0,0.0,291.0,0.0,Sandie Hinetthews,True									
8	0006_01,Earth,False,F/2/S,TRAPPIST-1e,26.0,False,42.0,1539.0,3.0,0.0,0.0,Billex Jacostaffey,True									
9	0006_02,Earth,True,G/0/S,TRAPPIST-1e,28.0,False,0.0,0.0,0.0,0.0,,Candra Jacostaffey,True									
10	0007_01,Earth,False,F/3/S,TRAPPIST-1e,35.0,False,0.0,785.0,17.0,216.0,0.0,Andona Beston,True									
11	0008_01,Europa,True,B/1/P,55 Cancri e,14.0,False,0.0,0.0,0.0,0.0,0.0,Erraiam Flatic,True									
12	0008_02,Europa,True,B/1/P,TRAPPIST-1e,34.0,False,0.0,0.0,,0.0,0.0,Altardr Flatic,True									
13	0008_03,Europa,False,B/1/P,55 Cancri e,45.0,False,39.0,7295.0,589.0,110.0,124.0,Wezena Flatic,True									
14	0009_01,Mars,False,F/1/P,TRAPPIST-1e,32.0,False,73.0,0.0,1123.0,0.0,113.0,Berers Barne,True									
15	0010_01,Earth,False,G/1/S,TRAPPIST-1e,48.0,False,719.0,1.0,65.0,0.0,24.0,Reney Baketton,False									
16	0011_01,Earth,False,F/2/P,TRAPPIST-1e,28.0,False,8.0,974.0,12.0,2.0,7.0,Elle Bertsontry,True									
17	0012_01,Earth,False,,TRAPPIST-1e,31.0,False,32.0,0.0,876.0,0.0,0.0,Justie Pooles,False									

Figure 2. Dataset in CSV Format

2.2. Pre-processing

In the pre-processing stage, the initial steps of data handling are performed, including selecting important features, cleaning null or missing data, and removing redundant data[14]. This process is illustrated in Figure 3.

PassengerId	HomePlanet	CryoSleep	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Transported
0001_01	Europa	False	TRAPPIST-1e	39.0	False	0.0	0.0	0.0	0.0	0.0	False
0002_01	Earth	False	TRAPPIST-1e	24.0	False	109.0	9.0	25.0	549.0	44.0	True
0003_01	Europa	False	TRAPPIST-1e	58.0	True	43.0	3576.0	0.0	6715.0	49.0	False
0003_02	Europa	False	TRAPPIST-1e	33.0	False	0.0	1283.0	371.0	3329.0	193.0	False
0004_01	Earth	False	TRAPPIST-1e	16.0	False	303.0	70.0	151.0	565.0	2.0	True
...	...	...	...	...	...	...	...	...	...	...	...
9276_01	Europa	False	55 Cancri e	41.0	True	0.0	6819.0	0.0	1643.0	74.0	False
9278_01	Earth	True	PSO J318.5-22	18.0	False	0.0	0.0	0.0	0.0	0.0	False
9279_01	Earth	False	TRAPPIST-1e	26.0	False	0.0	0.0	1872.0	1.0	0.0	True
9280_01	Europa	False	55 Cancri e	32.0	False	0.0	1049.0	0.0	353.0	3235.0	False
9280_02	Europa	False	TRAPPIST-1e	44.0	False	126.0	4688.0	0.0	0.0	12.0	True

Figure 3. Dataset processed with JupyterLab

2.3. Dataset Splitting

In the dateset splitting stage, the dataset is split into two sets: the training data and the testing data, with a proportion of 80% for training and 20% for testing. The goal is to prevent bias in predictions, meaning that the model should not perform well only on the training data but also on the actual test data. Specifically, for classification

algorithms, the stratified shuffle split technique is used to ensure that both datasets maintain the distribution of the two categories. The splitting result obtained 6,954 rows for the training data and 1,739 rows for the testing data. In the target column, the predicted "Transported" data is assigned the number 1, while the non-transported data is assigned the number 0.

2.4. Training and Testing

During the training and testing phase, the process is conducted using three algorithms: Support Vector Machine (SVM), Random Forest, and XGBoost. Prior to training and testing, feature scaling is applied using the Min-Max Scaler method. Additionally, hyperparameter tuning is performed simultaneously to optimize the parameters for each model. The tuning method used in this study is RandomSearchCV[15].

Furthermore, the training process incorporates K-Fold Cross-Validation to prevent the model's performance from being influenced by luck or bias in the test data selection[16].

2.5. Feature Selection

In this stage, data quality and model performance were enhanced by identifying and selecting the most relevant features using a feature importance technique. Rather than relying on traditional methods such as Mean Loss Decrease, this study adopted the Mean Score Decrease approach, which is more versatile and applicable across different classification models.

The Mean Decrease in Accuracy (MDA) method, commonly used for evaluating feature importance in machine learning models, follows a structured procedure to assess the contribution of each feature to model performance[17]. First, the values of each feature (or column) are individually shuffled to disrupt any existing relationship with the target variable, effectively removing the feature's informational value. Second, the model is re-evaluated using the modified dataset, keeping all other parameters constant. Finally, the drop in performance—typically measured by the decrease in accuracy or other relevant metrics—is calculated. A greater decrease in accuracy indicates a more important feature. This method provides an intuitive and model-agnostic approach to understanding which features most significantly influence predictive performance.

A feature that, when randomized, leads to a significant drop in performance is considered important. Conversely, features that cause little to no decrease in performance can be considered redundant or uninformative. These unimportant features were removed from the dataset to reduce noise, simplify the model, and improve generalization.

This data-driven selection process not only helped improve accuracy but also enhanced model interpretability and reduced computational complexity during training and tuning phases.

2.6. Evaluation

The performance of the classification models was evaluated using numerical analysis methods named classification report includes four key performance metrics: Precision, Recall, F1-score, and Support[18]. Each of these metrics offers different insights into the model's performance:

a. Precision: Measures the proportion of positive identifications that were actually correct.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

A high precision score indicates that when the model predicts a passenger will be transported, it is usually correct.

b. Recall: Measures the proportion of actual positives that were correctly identified.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

High recall means that most of the transported passengers were successfully identified by the model.

c. F1-Score: The harmonic mean of precision and recall.

$$F1 - Score = 2 \times \frac{Precision \times Recal}{Precision + Recall} \quad (3)$$

This score balances precision and recall and is particularly useful when the classes are imbalanced.

d. Support: Indicates the number of actual occurrences of each class in the test dataset. While not a performance metric, it provides context for interpreting the above scores.

The combination of visual plots and numerical scores enabled a comprehensive assessment of model performance. It also helped identify any bias toward a particular class and informed further model adjustments where necessary.

3. RESULTS AND DISCUSSION

3.1. Data Testing Before Feature Selection

To visually interpret the learning process and generalization behavior of the Random Forest model, a graphical approach was employed. Figure 4 illustrates the concept of Bootstrap Aggregating (Bagging) used by the Random Forest, in which each individual decision tree is trained on a different bootstrapped subset of the training data. The output of each tree contributes to the final prediction through majority voting (for classification) or averaging (for regression). This ensemble approach reduces variance and prevents overfitting by averaging out the noise of individual trees.

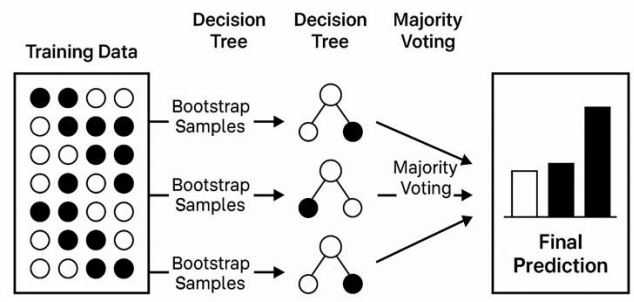

Figure 4. Bagging Illustration

The training process for the Random Forest classifier began with a hyperparameter tuning stage using the RandomizedSearchCV method combined with 5-fold cross-validation. In this process, the model was evaluated using a defined search space consisting of parameters such as the number of estimators (n_estimators), tree depth (max_depth), the fraction of features to consider at each split (max_features), and the minimum number of samples required at a leaf node (min_samples_leaf). A total of 10 candidate parameter combinations were randomly sampled from this space and evaluated across 50 training iterations.

From the tuning process, the best-performing parameter combination was found to be 133 trees (n_estimators = 133), a maximum depth of 47 (max_depth = 47), a feature usage fraction of approximately 0.477 (max_features = 0.4768), and a minimum of 11 samples required per leaf (min_samples_leaf = 11). This configuration was then used to train the model using the full training set and evaluated against the testing set.

The resulting performance metrics demonstrate a consistent and balanced outcome. The model achieved an accuracy of 0.8263 on the training dataset, while the average cross-validation accuracy across the 5 folds was 0.7967. When evaluated on the testing data, the model reached an accuracy of 0.7953. These scores suggest that the model generalizes well, as evidenced by the relatively small difference between training and testing accuracy. Furthermore, the high cross-validation score supports the robustness of the selected hyperparameters and indicates that the model is not overfitting the training data.

Figure 5. Learning Curve for Random Forest

In figure 5, the blue line represents the training accuracy, which starts below 0.78 and increases before stabilizing. The green line represents the test accuracy, which initially decreases from 0.85 before stabilizing around 0.82.

The increasing training accuracy followed by stabilization suggests the model is learning and improving as more data is used. However, the slight decline in test accuracy indicates that the model might be starting to overfit, as it performs slightly better on the training data than on the test data. The stabilization of both accuracies suggests that the model has reached an optimal level of complexity.

The training process for the Support Vector Machine (SVM) classifier was carried out using hyperparameter tuning via RandomizedSearchCV, with a 5-fold cross-validation approach. The hyperparameters optimized in this process were the regularization parameter C and the kernel coefficient gamma, both searched within a logarithmic scale ranging from 10^{-3} to 10^{3}, defined using a log-uniform distribution. This setting allows the search algorithm to explore a wide range of values, which is crucial for models like SVM that are sensitive to the choice of these parameters.

After conducting the randomized search, the best-performing combination of hyperparameters was found to be C = 0.201 and gamma = 0.027. With this configuration, the model achieved an accuracy of 0.5037 on the training data, a cross-validation mean score of 0.6812 across the five folds, and a test set accuracy of 0.4997.

The results indicate that while the SVM model reached a relatively good cross-validation score, its performance on the training and especially testing datasets was suboptimal. This significant drop in accuracy on the test set suggests that the model may not have generalized well and might have been affected by the complexity of the data or by the limitations in how the current feature set aligns with the SVM decision boundaries. Additional exploration, such as feature engineering or kernel adjustments, may be necessary to improve SVM performance in future iterations.

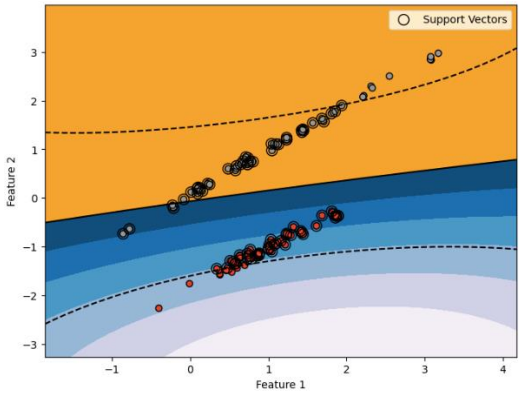

Figure 6. SVM Decision Boundary with Margin

Figure 6 illustrates the decision boundary (hyperplane) produced by the SVM classifier in a two-dimensional feature space. The solid black line represents the optimal hyperplane that separates the two classes. The dashed lines indicate the margins on either side of the hyperplane, which define the boundary of the maximum margin region. The data points shown in different colors correspond to the two classes, while the support vectors—critical points that lie on the margin boundaries—are highlighted with larger, unfilled markers. This visualization helps to demonstrate how the SVM identifies the hyperplane that maximizes the margin between the classes, ensuring good generalization to unseen data.

The XGBoost classifier was trained using hyperparameter tuning with RandomizedSearchCV and 5-fold cross-validation. The best parameters obtained were: max_depth = 5, learning_rate = 0.0266, n_estimators = 101, subsample = 0.586, colsample_bytree = 0.958, gamma = 2, reg_alpha = 1.13, and reg_lambda = 0.065.

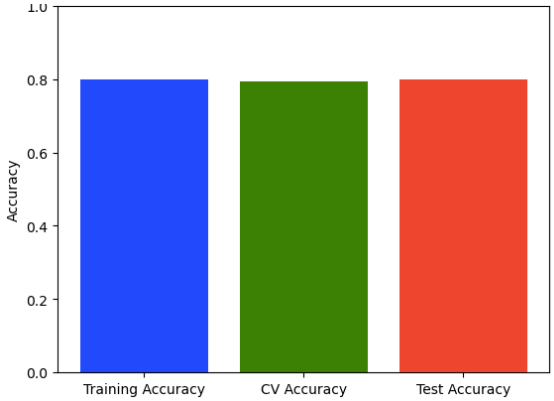

Figure 7. XGBoost Model Performance

With this configuration, which is shown in figure 7. the model achieved a training accuracy of 0.8041, a mean cross-validation score of 0.7915, and a test accuracy of 0.7976. These results indicate consistent and stable performance across datasets, suggesting that the model generalizes well and is not significantly overfitting.

3.2. Feature Selection Using Mean Score Decrease (MSD)

As part of the model improvement strategy through improvement by data, a feature selection process was conducted to identify the most significant predictors contributing to model performance. The method applied in this study is Mean Score Decrease (MSD), which was chosen for its versatility and model-agnostic nature, making it suitable for various classification algorithms used in this research, including Random Forest, SVM, and XGBoost.

MSD operates by systematically shuffling the values of each feature in the dataset while keeping all other features unchanged. The trained model is then re-evaluated on this modified dataset, and the corresponding drop in performance (e.g., accuracy or another relevant metric) is recorded. If the removal of a particular feature results in a significant decrease in model performance, the feature is deemed important. Conversely, if the perturbation of a feature does not substantially affect the model's accuracy, the feature is considered less influential or potentially redundant. The magnitude of this performance drop represents the mean score decrease and serves as the basis for ranking feature importance.

The results of the MSD analysis in this study revealed that the VIP feature had the lowest mean score decrease among all features. This indicates that shuffling the values of VIP did not significantly impact the model's accuracy, suggesting that it contributes minimally to the classification task. Based on this finding, the decision was made to remove the VIP feature from the dataset in subsequent training stages. This refinement aims to simplify the model, reduce the risk of overfitting, and enhance generalization by eliminating non-informative features. Figure 8 is a graph illustrating the mean score decrease for all analyzed features:

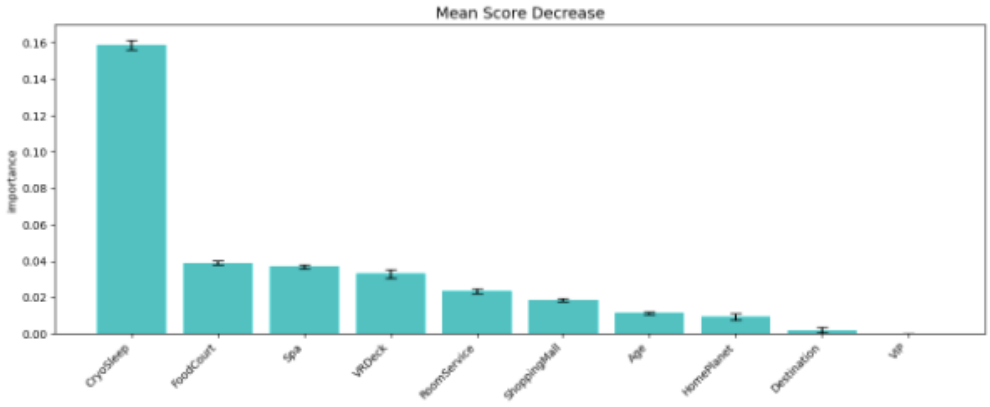

Figure 8. Mean Score Decrease Diagram

3. 3. Data Testing After Feature Selection

After removing the VIP feature as a result of feature importance analysis, the three machine learning models—Random Forest, Support Vector Machine (SVM), and XGBoost—were re-evaluated to assess the impact of this selection on model performance. The comparative results reveal varied but generally positive changes across the models.

For the Random Forest model, the selected hyperparameters after tuning were max_depth = 23, max_features ≈ 0.592, min_samples_leaf = 20, and n_estimators = 190. The training accuracy experienced a slight decline from 0.8263 to 0.8181, while cross-validation accuracy remained nearly unchanged (0.7967 before, 0.7964 after). Interestingly, test accuracy improved modestly from 0.7953 to 0.8005, suggesting better generalization and robustness when the less informative VIP feature was removed.

The SVM model showed the most substantial improvement across all metrics. With tuned hyperparameters C ≈ 0.0033 and gamma ≈ 0.0046, training accuracy increased significantly from 0.5037 to 0.7328. Cross-validation and test accuracy also rose markedly, from 0.6812 to 0.7141 and 0.4997 to 0.7240, respectively. These results imply that the presence of the VIP feature may have introduced noise or variance that the SVM model struggled to interpret. Its removal allowed the model to learn a more meaningful decision boundary.

In the case of XGBoost, the post-selection tuning yielded max_depth = 10, n_estimators = 162, and other refined parameters, including learning rate and regularization factors. Performance improved slightly but consistently, with training accuracy moving from 0.8041 to 0.8139, cross-validation from 0.7915 to 0.7931, and test accuracy from

0.7976 to 0.7993. While the increase was modest, the consistency across all metrics indicates a minor but positive effect of removing the VIP feature.

In summary, the feature selection process—guided by the mean score decrease method—helped improve model performance, especially for SVM, by eliminating a non-contributive feature. These findings reinforce the importance of careful feature evaluation in machine learning workflows to avoid overfitting and enhance generalization capabilities.

A comparative analysis between the model performance before and after feature selection can be observed in Table 2. This table highlights the key evaluation metrics—training accuracy, cross-validation accuracy, and test accuracy—for each algorithm (Random Forest, SVM, and XGBoost), clearly illustrating the impact of removing the VIP feature on model performance.

Table 2. Comparation of model Performance befor and after Feature Selection

Model	Condition	Train Accuracy	CV Accuracy	Test Accuracy
Random Forest	Before Feature Selection	0.8263	0.7967	0.7953
	After Feature Selection	0.8181	0.7964	0.8005
SVM	Before Feature Selection	0.5037	0.6812	0.4997
	After Feature Selection	0.7328	0.7141	0.7240
XGBoost	Before Feature Selection	0.8041	0.7915	0.7976
	After Feature Selection	0.8139	0.7931	0.7993

3. 4. Evaluation After Feature Selection

The Random Forest model performs consistently well, achieving 82% accuracy on the training data and 80% on the test data. Precision, recall, and F1-scores are balanced across both classes, indicating good generalization with no signs of overfitting. The classification performance can be visually observed in Figure 9.

Train report

	precision	recall	f1-score	support
False	0.83	0.80	0.81	3452
True	0.81	0.83	0.82	3502
accuracy			0.82	6954
macro avg	0.82	0.82	0.82	6954
weighted avg	0.82	0.82	0.82	6954

Test report

	precision	recall	f1-score	support
False	0.81	0.79	0.80	863
True	0.80	0.81	0.80	876
accuracy			0.80	1739
macro avg	0.80	0.80	0.80	1739
weighted avg	0.80	0.80	0.80	1739

Figure 9. Classification Report for Random Forest

The Support Vector Machine (SVM) model shows moderate performance, with 73% accuracy on the training set and 72% on the test set. The model appears slightly biased toward the "False" class, achieving higher recall in that category, while struggling more with the "True" class. This indicates a potential imbalance in class prediction. Overall, the SVM performs reasonably well, but not as balanced as the Random Forest model. The classification performance is illustrated in Figure 10.

Train report

	precision	recall	f1-score	support
False	0.69	0.83	0.76	3452
True	0.79	0.64	0.71	3502
accuracy			0.73	6954
macro avg	0.74	0.73	0.73	6954
weighted avg	0.74	0.73	0.73	6954

Test report

	precision	recall	f1-score	support
False	0.68	0.84	0.75	863
True	0.79	0.61	0.69	876
accuracy			0.72	1739
macro avg	0.74	0.72	0.72	1739
weighted avg	0.74	0.72	0.72	1739

Figure 10. Classification Report for Support Vector Machine

The XGBoost model demonstrates consistent and well-balanced performance, achieving 81% accuracy on the training set and 80% on the test set. Precision and recall values are relatively even across both classes, suggesting that the model handles both "True" and "False" categories effectively. This balance indicates good generalization and reliable classification capability. The detailed results are presented in Figure 11.

Train report

	precision	recall	f1-score	support
False	0.83	0.78	0.81	3452
True	0.80	0.84	0.82	3502
accuracy			0.81	6954
macro avg	0.82	0.81	0.81	6954
weighted avg	0.81	0.81	0.81	6954

Test report

	precision	recall	f1-score	support
False	0.81	0.78	0.79	863
True	0.79	0.82	0.80	876
accuracy			0.80	1739
macro avg	0.80	0.80	0.80	1739
weighted avg	0.80	0.80	0.80	1739

Figure 11. Classification Report for XGBoost

3. 5. Discussion

The results obtained from the classification experiments show that feature selection has a notable impact on model performance, especially in the case of the Support Vector Machine (SVM). After removing the VIP feature—identified as the least important based on the Mean Score Decrease (MSD) method—the SVM model experienced a considerable increase in accuracy, from 0.50 to 0.72 on the test data. This improvement suggests that the VIP feature was contributing noise or irrelevant information, thereby hindering the model's ability to generalize.

In contrast, both Random Forest and XGBoost models showed relatively stable performance before and after feature selection, with only marginal differences in their accuracy scores. Random Forest achieved the highest test accuracy after feature selection at 0.80, while XGBoost closely followed with 0.799. These results suggest that both ensemble models are robust and less sensitive to the inclusion or exclusion of the VIP feature, likely due to their inherent ability to handle feature importance internally through techniques such as bootstrapping and regularization.

Among the three classifiers evaluated—SVM, Random Forest, and XGBoost—the Random Forest model emerges as the best-performing model after feature selection. It not only achieved the highest test accuracy (0.8005) but also maintained a strong balance between precision and recall across both classes. Additionally, its macro and weighted F1-scores were consistently higher compared to the other models. These findings indicate that Random Forest offers the most reliable performance in this classification task, effectively handling both bias and variance, and generalizing well to unseen data.

Therefore, while SVM showed the most improvement post-feature selection, Random Forest remains the most optimal model overall based on its consistent and superior performance across all evaluation metrics. The comparison of model accuracy after feature selection is shown in Figure 12.

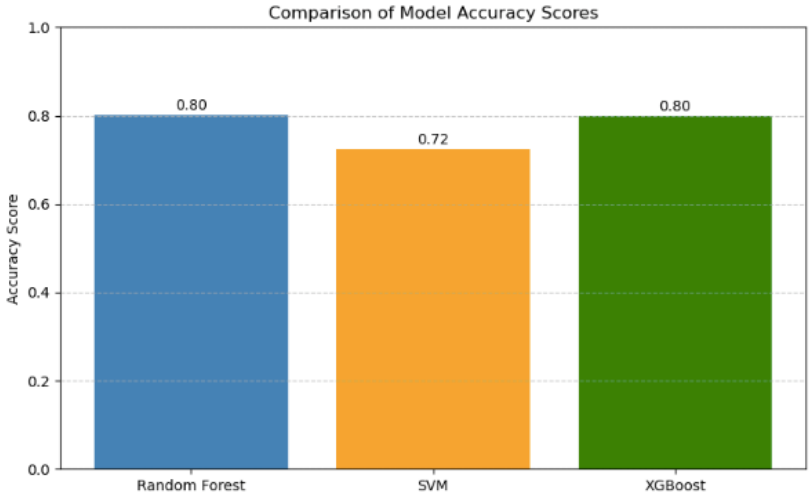

Figure 12. Comparison of Model Accuracy after Feature Seleection

4. CONCLUSION

Based on the overall analysis, it can be concluded that feature selection has a noticeable impact on model performance. The removal of the VIP feature led to improvements in classification accuracy, particularly for the SVM model, which showed the most significant gain. Among all evaluated models, the Random Forest classifier achieved the highest accuracy with a value of 0.8005 on the test dataset, making it the best-performing model in this study. The XGBoost model followed closely with an accuracy of 0.7993, while the SVM model showed the lowest accuracy of 0.7240.

These results demonstrate that careful feature selection, combined with appropriate model tuning, can significantly enhance predictive performance in classification tasks. The Random Forest model stands out as the most reliable classifier for this dataset, delivering superior accuracy compared to the other algorithms evaluated.

DAFTAR PUSTAKA

[1] M. Kaku, *Quantum Supremacy: How the Quantum Computer Revolution Will Change Everything*. Knopf Doubleday Publishing Group, 2023. [Online]. Available: https://books.google.co.id/books?id=FkKNEAAAQBAJ

[2] G. K. O'Neill, *2081: A Hopeful View of the Human Future*. in Touchstone book. Simon and Schuster, 1981. [Online]. Available: https://books.google.co.id/books?id=fyVmAAAAMAAJ

[3] A. M. Hein, M. Pak, D. Pütz, C. Bühler, and P. Reiss, "World ships - Architectures & feasibility revisited," 2012. Accessed: Apr. 12, 2025. [Online]. Available: https://www.researchgate.net/publication/236177990_World_Ships_-_Architectures_Feasibility_Revisited

[4] E. A. Brettschneider, "Encompass Encompass Honors Theses Student Scholarship Exploring the Ethics of Human Space Travel: Navigating the Exploring the Ethics of Human Space Travel: Navigating the Challenges and Implications of Missions Past, Present, and Challenges and Implicati," 2023, [Online]. Available: https://encompass.eku.edu/honors_theses/1000

[5] G. Dautcourt and M. Abdel-Megied, "Revisiting the light cone of the Gödel universe," *Class. Quantum Gravity*, vol. 23, no. 4, pp. 1269–1288, Feb. 2006, doi: 10.1088/0264-9381/23/4/013.

[6] W. Northcutt and W. Northcutt, "A Theory of Gravity Based on Dimensional Perturbations of Objects in Flat Spacetime A Theory of Gravity Based on Dimensional Perturbations of Objects in Flat Spacetime," pp. 0–119, 2025, doi: 10.20944/preprints202411.0620.v5.

[7] M. Memarzadeh, A. Akbari Asanjan, and B. Matthews, "Robust and Explainable Semi-Supervised Deep Learning Model for Anomaly Detection in Aviation," *Aerospace*, vol. 9, no. 8, p. 437, Aug. 2022, doi: 10.3390/aerospace9080437.

[8] S. Cuéllar, M. Santos, F. Alonso, E. Fabregas, and G. Farias, "Explainable anomaly detection in spacecraft telemetry," *Eng. Appl. Artif. Intell.*, vol. 133, p. 108083, Jul. 2024, doi: 10.1016/j.engappai.2024.108083.

[9] A. Polatoğlu and E. Gül, "Unveiling the impact of cosmic rays and solar activities on climate through optimized boost algorithms," *J. Atmos. Solar-Terrestrial Phys.*, vol. 265, p. 106360, Dec. 2024, doi:

10.1016/j.jastp.2024.106360.

[10] E. Waisberg *et al.*, "Challenges of Artificial Intelligence in Space Medicine," *Sp. Sci. Technol. (United States)*, vol. 2022, Jan. 2022, doi: 10.34133/2022/9852872.

[11] J. Yan, A. P. Leung, Z. Pei, D. C. Y. Hui, and S. Kim, "DeepGrav: Anomalous Gravitational-Wave Detection Through Deep Latent Features," Mar. 2025, Accessed: Apr. 12, 2025. [Online]. Available: https://arxiv.org/abs/2503.03799v2

[12] G. Quaye, "Random Forest For High-Dimensional Data," *Open Access Theses Diss.*, Aug. 2024, Accessed: Apr. 12, 2025. [Online]. Available: https://scholarworks.utep.edu/open_etd/4201

[13] A. Syahreza, N. K. Ningrum, and M. A. Syahrazy, "Perbandingan Kinerja Model Prediksi Cuaca: Random Forest, Support Vector Regression, dan XGBoost," *Edumatic J. Pendidik. Inform.*, vol. 8, no. 2, pp. 526–534, Dec. 2024, doi: 10.29408/EDUMATIC.V8I2.27640.

[14] A. A. Akimov, D. R. Valitov, and A. I. Kubryak, "DATA PREPROCESSING FOR MACHINE LEARNING," *Научное обозрение. Технические науки (Scientific Rev. Tech. Sci.*, no. №2 2022, pp. 26–31, 2022, doi: 10.17513/srts.1391.

[15] N. Alamsyah, B. Budiman, T. P. Yoga, and R. Y. R. Alamsyah, "XGBOOST HYPERPARAMETER OPTIMIZATION USING RANDOMIZEDSEARCHCV FOR ACCURATE FOREST FIRE DROUGHT CONDITION PREDICTION," *J. Pilar Nusa Mandiri*, vol. 20, no. 2, pp. 103–110, Sep. 2024, doi: 10.33480/PILAR.V20I2.5569.

[16] D. Berrar, "Cross-validation," in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, Elsevier, 2018, pp. 542–545. doi: 10.1016/B978-0-12-809633-8.20349-X.

[17] H. Han, X. Guo, and H. Yu, "Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest," *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 0, pp. 219–224, Jul. 2016, doi: 10.1109/ICSESS.2016.7883053.

[18] Dewi Widyawati and Amaliah Faradibah, "Comparison Analysis of Classification Model Performance in Lung Cancer Prediction Using Decision Tree, Naive Bayes, and Support Vector Machine," *Indones. J. Data Sci.*, vol. 4, no. 2, pp. 80–89, 2023, doi: 10.56705/ijodas.v4i2.76.