PERINTIS
Persatuan Saintis Muslim Malaysia

PERINTIS eJournal
Vol. 15, No. 1, Month 2025, pp. 1-14
e-ISSN: 2232-0725

ANALYSIS AND COMPARISON OF CLASSIFICATION ALGORITHMS FOR CREDIT APPROVAL IN ISLAMIC BANKS

Dwi Pebrianti[1]✉, Whena Wijanarko[2], Luhur Bayuaji[3], Rusdah[2], Siti Fauziah Toha[1]

[1] Faculty of Engineering, International Islamic University Malaysia, 53100, Selangor, Malaysia
[2] Faculty of Information Technology, Universitas Budi Luhur, 12260, Jakarta, Indonesia
[3] Faculty of Data Science & Information Technology, INTI International University, 71800, Nilai, Malaysia

✉ Corresponding author: dwipebrianti@iium.edu.my
Received 11 29, 2024
Revised 05 10, 2025
Accepted 05 16, 2025

ABSTRACT

In the context of Islamic banking, an efficient and quick credit analysis process is crucial to meet customer needs without compromising Sharia principles. Although Islamic banks do not charge interest, profits are obtained through contracts, requiring thorough and accurate credit analysis. This study evaluates the performance of various classification algorithms, including C4.5, Random Forest, K-Nearest Neighbors, and Naïve Bayes, in predicting credit approval decisions based on factors such as *character*, *capacity*, *capital*, *collateral*, and *economic conditions*. Using a dataset from an Islamic bank, the algorithms were assessed through precision, sensitivity, and accuracy metrics. The results highlight significant performance differences among the algorithms. Random Forest and Decision Tree demonstrated strong training accuracy but suffered from overfitting, limiting their generalization to new data. Conversely, the C4.5 algorithm achieved a balanced performance with a testing accuracy of 74.5%, making it a promising candidate for practical application in Sharia-compliant credit risk assessment. This study emphasizes the importance of selecting algorithms that address overfitting while maintaining robust predictive accuracy, contributing to improved decision-making in Islamic banking.

Keywords: Credit Score; C4.5 Algorithm; Naïve Bayes; Random Forest; k-Nearest Neighbors

1.0 INTRODUCTION

The need for efficient credit analysis in the banking sector has grown significantly in recent years. While traditional methods of credit evaluation can still be performed manually, they are often time-consuming, leading to delays in decision-making. This inefficiency not only affects customer satisfaction but also impacts on the operational effectiveness of banks. In Islamic banking, this challenge is compounded by the requirement to adhere to Sharia principles, which prohibit the charging of interest and necessitate alternative profit-sharing mechanisms. Consequently, a rigorous and accurate credit analysis process is essential to assess the feasibility of credit applications and to ensure compliance with Islamic financial practices.

Sharia-compliant credit approval relies on evaluating various factors, including *character*, *capacity*, *capital*, *collateral*, and *economic conditions*. The complexity of analyzing these factors makes it imperative to adopt advanced tools and methods for automating and enhancing the decision-making process. Among these, machine learning and data mining techniques have emerged as powerful approaches to predict creditworthiness and manage credit risk effectively.

In credit approval processes, especially within Islamic banking systems, unbalanced datasets often pose a challenge where non-approved applications significantly outnumber approved ones. This imbalance can bias classifiers toward the majority class, undermining predictive accuracy for minority instances. Several resampling techniques have been proposed to address this issue, including Random Over-Sampling [1], [2], Random Under-Sampling [2], SMOTE [2], [3], and its variants. Among these, SMOTE (Synthetic Minority Over-sampling Technique) demonstrates consistent performance improvements by generating synthetic minority class samples, thereby enhancing classifier sensitivity. Although hybrid approaches like SMOTEENN [4] offer additional noise reduction, this study adopts SMOTE due to its simplicity, wide adoption, and proven effectiveness in credit risk classification tasks.

The focus of this study is to combine SMOTE with several classifications algorithms, including C4.5., Random Forest, K-Nearest Neighbors, and Naïve Bayes in the context of Islamic banking. These algorithms have demonstrated effectiveness in various domains, such as predicting customer behavior, assessing credit risk, and optimizing operational processes[5]. By leveraging these algorithms, this research aims to address the limitations of traditional methods and provide insights into the most suitable techniques for Islamic banking.

The C4.5 algorithm, a widely used decision tree-based classifier, has proven to be versatile due to its ability to handle both numerical and categorical data, manage missing values, and generate interpretable decision rules. Similarly, Random Forest, known for its robustness and accuracy, combines multiple decision trees to improve predictive performance. However, issues such as overfitting and scalability remain critical concerns in applying these methods to real-world datasets[6], [7].

This study provides a comprehensive evaluation of these algorithms using a dataset from an Islamic bank. It aims to highlight the trade-offs between accuracy, generalizability, and computational efficiency, ultimately guiding the selection of optimal algorithms for Sharia-compliant credit analysis. The findings of this research contribute to the growing body of knowledge on machine learning applications in Islamic finance, offering practical solutions to enhance decision-making processes in this unique banking environment.

In this paper, the evaluation is focused on the credit application process based solely on the form filled out by the applicant. While the machine learning models used in this study aim to assess creditworthiness based on this form, it is important to note that, in actual banking practices, additional factors are considered by the bank. These include the applicant's CCRIS (Centralized Credit Reference Information System), CTOS (Credit Tip-Off Service), DSR (Debt Service Ratio), and LE (Loan Eligibility). Thus, the scope of this research is limited to form data, and these supplementary data sources would also play a crucial role in the final credit approval decision.

2.0 METHODOLOGY

This study employs a quantitative approach to evaluate the performance of multiple classification algorithms in predicting credit approval for Islamic banking. The research methodology encompasses data collection, preprocessing, algorithm application, and performance evaluation, as illustrated in Fig. 1.

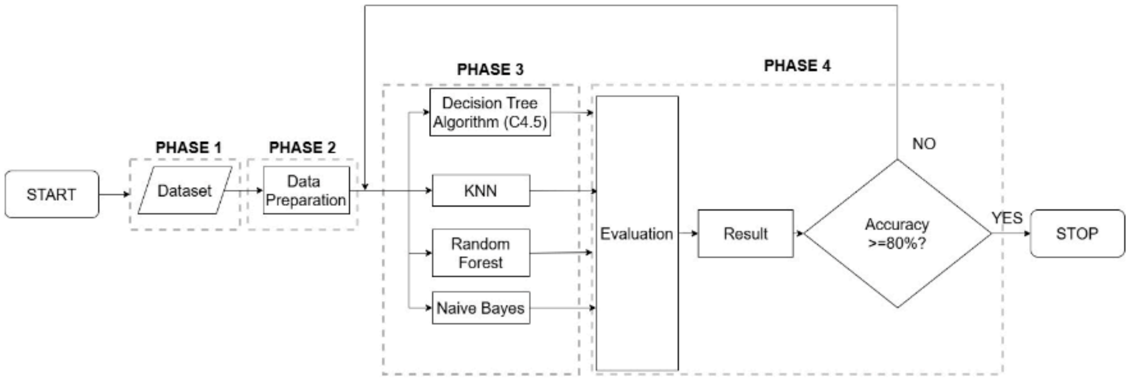

Figure 1: Data Mining and Bank Credit

2.1 Data Collection and Data Preparation

The dataset utilized in this study was obtained from an internal database of a Sharia bank in Indonesia. This dataset classifies individuals based on a set of attributes into either "*good*" or "*bad*" credit risk categories. Due to the sensitive nature of the data, certain personal identifiers such as address were omitted. The dataset consists of 1,000 multivariate records with 10 features, comprising both categorical and integer types. These features include *bank account status*, *credit history*, *loan purpose*, and *credit amount*, among others.

To ensure the data was suitable for machine learning modeling, a structured preprocessing pipeline was implemented, consisting of the following steps:

1. **Column Removal**
   Several features deemed redundant or irrelevant after transformation were removed, including 'Saving accounts', 'Checking account', 'Purpose', 'Sex', 'Housing', 'Age_cat', 'Risk', and 'Risk_good'.
2. **Age Categorization**
   The 'Age' feature was binned into labeled categories: 'Student', 'Young', 'Adult', and 'Senior', to assess the effect of age groups on credit risk.
3. **Missing Value Treatment**
   Missing entries in 'Saving accounts' and 'Checking account' were imputed with a placeholder value 'no_inf', signifying the absence of information while retaining the records.
4. **One-Hot Encoding**
   Categorical features were transformed into binary dummy variables to allow compatibility with machine learning algorithms. Features transformed include 'Purpose', 'Sex', 'Housing', 'Saving accounts', 'Checking account', and the new 'Age Category'.
5. **Logarithmic Transformation**
   The 'Credit amount' feature exhibited skewed distribution and was log-transformed to normalize its scale and reduce the impact of outliers.
6. **Handling Class Imbalance with SMOTE**
   The dataset initially exhibited imbalance, with a majority of samples labeled as "Risk_good". The Synthetic Minority Over-sampling Technique (SMOTE) was applied to synthetically generate new instances for the minority class ("Risk_bad"), improving the class balance and model robustness.
7. **Final Feature Selection**
   The final feature set included demographic, financial, employment, and credit history variables along with the dummy variables resulting from encoding. This curated set was used as input for the classification models.
8. **Data Splitting**
   The processed dataset was divided into training and testing sets in an 80:20 ratio. This allowed model training and subsequent validation using unseen data.

Table 1 shows the detailed explanation of the required data set.

Table 1: Dataset used in the study

Pre-processing Method	Information
Division of Age Categories	Categorize age into 'Student', 'Young', 'Adult', 'Senior' using certain intervals.
Data Separation	Split the dataset based on credit risk categories ('good' and 'bad').
One-Hot Encoding	Convert categorical variables to dummy variables, and fill in missing values in 'Saving accounts' and 'Checking accounts'
Variable Transformation	Convert variables such as 'Purpose', 'Sex', 'Housing', etc., to dummy variables.
Column Deletion	Delete columns that have been transformed or are no longer needed.
Logarithmic Transformation	Applying a logarithmic function to 'Credit amount' to normalize the distribution.
Data Inequality Handling (SMOTE)	Using the SMOTE method to balance the class distribution on the target variable.
Distribution of Training and Testing Data	Divide the data into a training set and a test set with certain proportions to prepare the model training process.

To address the severe class imbalance (~80:20 ratio of non-default to default loans), Synthetic Minority Over-sampling Technique (SMOTE) done by Chawla *et al.* in 2002 is used to generate additional synthetic minority examples instead of simply duplicating existing ones [8]. By augmenting the minority class, SMOTE improves the class balance to roughly 60:40 and is preferred over naive oversampling or undersampling, as it avoids overfitting on repeated samples and minimizes information loss. Prior studies on credit risk prediction report that SMOTE-based oversampling significantly enhances model performance on imbalanced loan datasets – for example, Brown and Mues found improved classification accuracy on imbalanced credit scoring data[9]. Additionally, to mitigate skewness in numeric features, we apply a log transformation, which stabilizes variance and makes their distributions more normal. This transformation reduces heterogeneity and improves interpretability in credit risk models. Finally, categorical variables are encoded using one-hot encoding (i.e., dummy variables) so that no spurious ordinal relationship is imposed and algorithms like logistic regression can utilize these features. This encoding preserves all category information and has been associated with improved accuracy and transparency in loan approval models.

From the selection process, data preprocessing will be carried out so that the dataset can be used. This research uses the C4.5 Algorithm, Random Forest, Naïve Bayes, and KNN. Evaluation is performed by observing the results of the comparison of these algorithms. The accuracy level is measured using a confusion matrix model evaluation, allowing us to determine the accuracy of these algorithms.

2.2 C4.5 Algorithm

The C4.5 algorithm is a decision tree-based machine learning method used for classification tasks, where each node in the tree represents a decision based on an attribute, and each leaf node represents a classification outcome. It selects the best attribute for splitting the data based on the gain ratio, a measure that evaluates the effectiveness of an attribute in dividing the data. The process continues recursively, with the algorithm splitting the data into subsets at each decision node and pruning the tree afterward to avoid overfitting.

In the context of loan approval, C4.5 uses attributes like Age, Income, Sex, Job, and Risk Category to predict whether a loan is approved or rejected as illustrated in Figure 2. The algorithm builds a decision tree by selecting the attribute that best splits the data at each node and continues this process until it reaches a decision. The resulting tree classifies new applicants based on their features, providing an effective tool for automated loan approval predictions. This approach ensures a systematic and interpretable decision-making process based on applicant data.

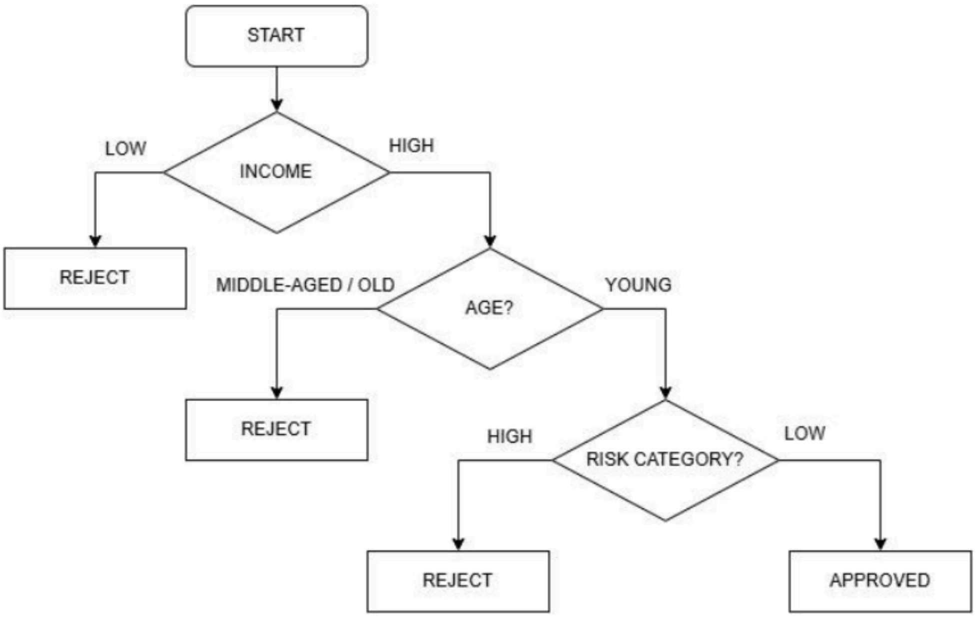

Figure 2: Illustration of C4.5 Algorithm Implemented to The Study

2.3 Naive Bayes

Naïve Bayes is a probabilistic machine learning algorithm based on Bayes' Theorem, which is used for classification tasks [10]. The key idea behind Naïve Bayes is that it assumes all features (or attributes) are independent given the class label. This assumption is called "*naïve*" because it often does not hold in real-world scenarios, but the algorithm can still perform surprisingly well despite this simplification.

Figure 3 illustrates the flow of the Naïve Bayes algorithm process, where the flow explains the process from the start to the calculation of the percentage of true positives and true negatives. The parameters used are the fields "*Name*, *Occupation*, *Age*, *Gender*, *Risk Category*," where these fields will display the percentage of true positives and true negatives within the Naïve Bayes algorithm.

2.4 K-Nearest Neighbors

The K-NN algorithm is a method that uses supervised learning. The difference between supervised learning and unsupervised learning is that supervised learning aims to discover new patterns in data by relating existing data patterns to new data. In contrast, unsupervised learning involves data that does not yet have any patterns, and its goal is to find patterns within the data. The purpose of the K-NN algorithm is to classify new objects based on their attributes and training samples.

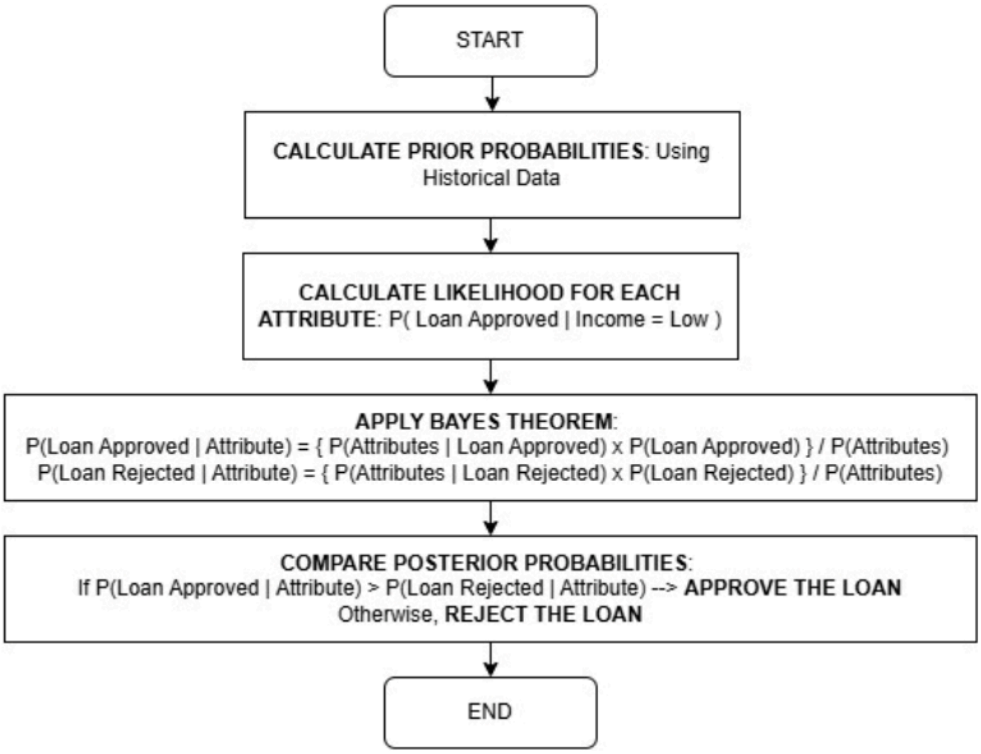

Figure 3: Illustration of Naïve Bayes

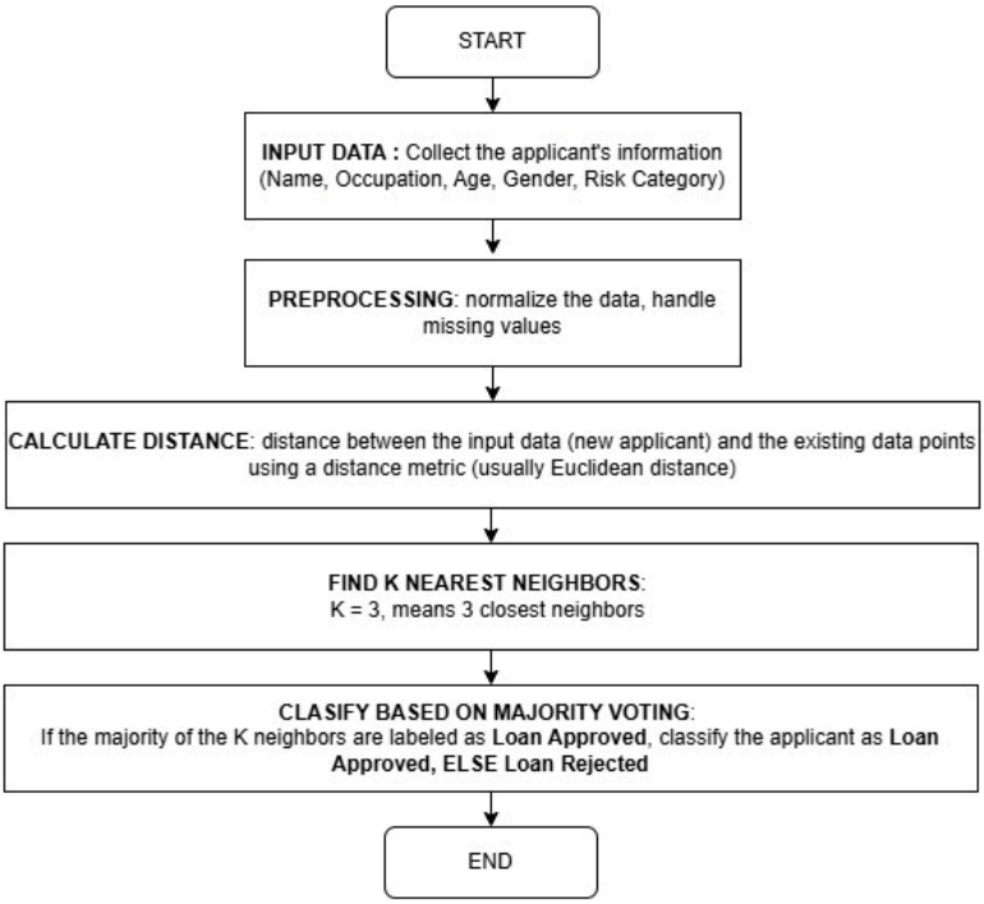

Figure 4: Illustration of k-Nearest Neighbour

Figure 4 illustrates the basic concept of K-NN implemented in the loan approval case study. The parameters used are the fields "*Name*, *Occupation*, *Age*, *Gender*, *Risk Category*," where these fields display the percentage of true positives and true negatives within the KNN algorithm.

2.5 Random Forest

Random Forest is an ensemble learning algorithm that builds multiple decision trees to improve prediction accuracy and reduce overfitting compared to a single decision tree [11]. In the context of loan approval, Random Forest uses attributes such as *Name*, *Occupation*, *Age*, *Gender*, and *Risk Category* to predict whether a loan should be **approved** or **rejected**. The algorithm works by randomly selecting subsets of features and training multiple decision trees on different samples of the training data. Each tree in the forest makes its own decision, and the final prediction is made by majority voting across all the trees. This process ensures that the model is more robust and less sensitive to noise and outliers in the data.

Figure 5 shows the illustration of Random Forest for loan approval task. The Random Forest process for loan approval begins with collecting the applicant's information, including *Name*, *Occupation*, *Age*, *Gender*, and *Risk Category*. This data is used to create multiple decision trees through *Bootstrap Sampling*, where random subsets of the training data are used to train each tree. Each decision tree independently selects a subset of features (e.g., *Age*, *Risk Category*) to make decisions at each node. This process ensures that each tree has diversity in feature selection and data, leading to more robust models that are less prone to overfitting compared to a single decision tree.

Once the trees are built, the Voting Mechanism comes into play, where each tree gives a prediction, either **Loan Approved** or **Loan Rejected**. The final loan decision is made by taking

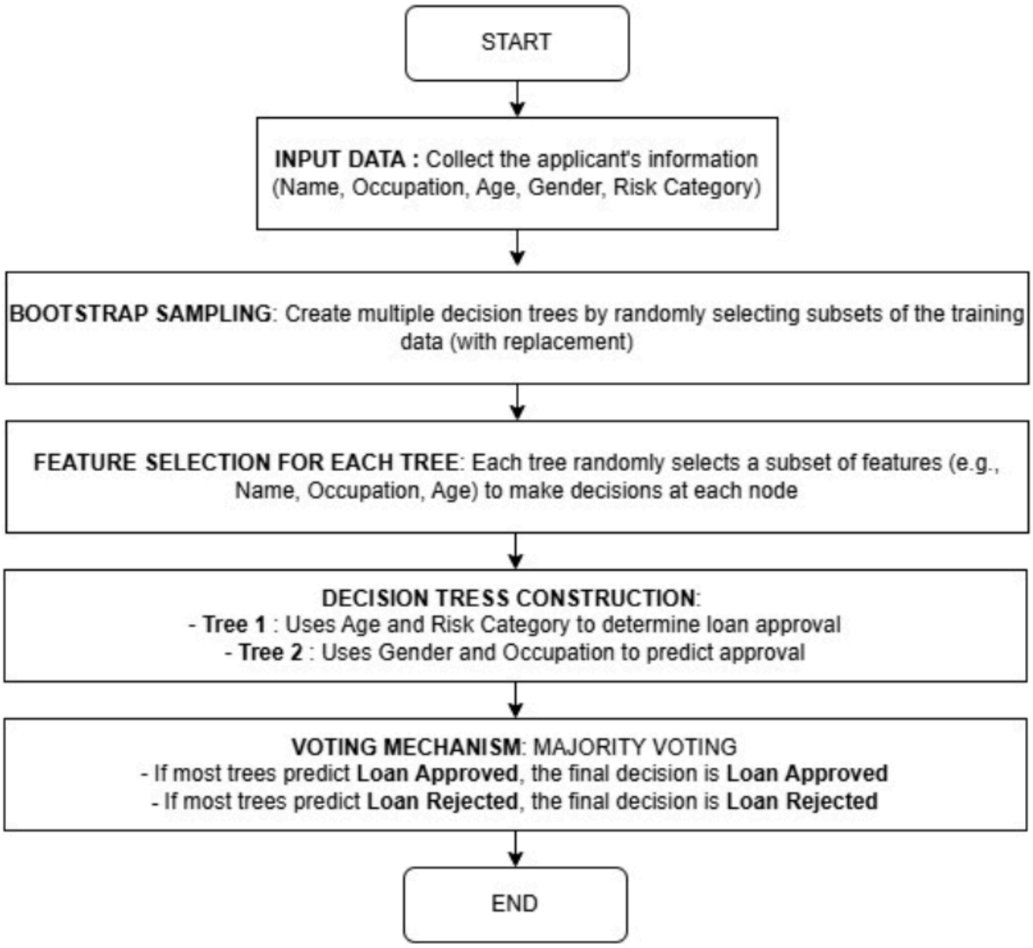

Figure 5: Illustration of Random Forest

the majority vote from all the trees in the forest. If most trees predict Loan Approved, the loan is approved; if most predict Loan Rejected, the loan is denied. This ensemble approach, combining multiple trees with varied decision-making, ensures that the model is more accurate and less biased, providing a reliable prediction based on the applicant's profile. By aggregating predictions, Random Forest delivers better generalization and is effective in handling complex, non-linear relationships in the data.

2.6 Experimental Setup

The experimental setup involved training and testing the selected algorithms on the dataset using a 70:30 split for training and testing data. The models were implemented using Python programming language and relevant machine learning libraries. Hyperparameter tuning was conducted to optimize model performance, and cross-validation was employed to ensure robustness.

3.0 RESULTS AND DISCUSSION

3.1 Data Description

The data description explains the dataset that will be used in this research. The dataset used to train the algorithm models in this study is relatively simple and relevant for use in the context of Islamic banking. This dataset provides various essential attributes for credit risk analysis, enabling Islamic banks to adapt and implement effective predictive models for assessing customer creditworthiness. Pre-processing data involves determining the dataset to be used as the research object and the data to be input into the study.

Figure 6 shows the raw data used in this study. The dataset used in this study contains various features of loan applicants, such as *Age*, *Sex*, *Job*, *Housing*, *Saving Account*, *Checking Account, Credit Amount, Duration, Purpose, and Risk (the target variable)*. The goal is to predict the Risk category (either good or bad) based on these attributes. The Risk category serves as the outcome for the loan approval decision, with "*good*" representing approved loans and "*bad*" representing rejected loans. The data includes both categorical and numerical variables, and it will be used to train different machine learning algorithms. The raw data contains some missing or inconsistent values, especially in fields like Saving Account and Checking Account, which need to be cleaned or transformed before use in a model.

Figure 7 shows the preprocessed data that has undergone several transformations to prepare it for machine learning. The categorical variables (such as Purpose and Sex) have been converted into boolean (True/False) values, which makes them suitable for classification algorithms like Naïve Bayes, K-Nearest Neighbors, and Random Forest. The missing values have been handled, and the data is now clean and structured. For instance, the Purpose column, which originally had values like 'Rumah' (House), is now represented as binary columns such as Purpose_car, Purpose_domestic appliances, and others, each indicating the presence or absence of a specific loan purpose. This transformation allows the model to interpret each loan's purpose as a feature, making it easier to classify applicants based on the different loan categories. After preprocessing, the data is now ready for input into machine learning algorithms to predict loan approval outcomes.

A	B	C	D	E	F	G	H	I	J	K
	Age	Sex	Job	Housing	Saving accc	Checking a	Credit amo	Duration	Purpose	Risk
0	67	male	2	own	NA	little	1169	6	Rumah	good
1	22	female	2	own	little	moderate	5951	48	Rumah	bad
2	49	male	1	own	little	NA	2096	12	Rumah	good
3	45	male	2	free	little	little	7882	42	Rumah	good
4	53	male	2	free	little	little	4870	24	Rumah	bad
5	35	male	1	free	NA	NA	9055	36	Rumah	good
6	53	male	2	own	quite rich	NA	2835	24	Rumah	good
7	35	male	3	rent	little	moderate	6948	36	Rumah	good
8	61	male	1	own	rich	NA	3059	12	Rumah	good
9	28	male	3	own	little	moderate	5234	30	Rumah	bad
10	25	female	2	rent	little	moderate	1295	12	Rumah	bad
11	24	female	2	rent	little	little	4308	48	Rumah	bad
12	22	female	2	own	little	moderate	1567	12	Rumah	good
13	60	male	1	own	little	little	1199	24	Rumah	bad
14	28	female	2	rent	little	little	1403	15	Rumah	good
15	32	female	1	own	moderate	little	1282	24	Rumah	bad
16	53	male	2	own	NA	NA	2424	24	Rumah	good
17	25	male	2	own	NA	little	8072	30	Rumah	good
18	44	female	3	free	little	moderate	12579	24	Rumah	bad
19	31	male	2	own	quite rich	NA	3430	24	Rumah	good
20	48	male	2	own	little	NA	2134	9	Rumah	good
21	44	male	2	rent	quite rich	little	2647	6	Rumah	good
22	48	male	1	rent	little	little	2241	10	Rumah	good
23	44	male	2	own	moderate	moderate	1804	12	Rumah	good
24	26	male	2	own	NA	NA	2069	10	Rumah	good
25	36	male	1	own	little	little	1374	6	Rumah	good
26	39	male	1	own	little	NA	426	6	Rumah	good
27	42	female	2	rent	rich	rich	409	12	Rumah	good
28	34	male	2	own	little	moderate	2415	7	Rumah	good
29	63	male	2	own	little	little	6836	60	Rumah	bad
30	36	male	2	own	rich	moderate	1913	18	Rumah	good
31	27	male	2	own	little	little	4020	24	Rumah	good
32	30	male	2	own	moderate	moderate	5866	18	Rumah	good
33	57	male	1	rent	NA	NA	1264	12	Rumah	good
34	33	female	3	own	little	rich	1474	12	Rumah	good

Figure 6: Raw Data Used in The Study

Hasil setelah dilakukan preprocessing pembersihan data

[56]:

	Age	Job	Credit amount	Duration	Purpose_car	Purpose_domestic appliances	Purpose_education	Purpose_furniture/equipment	Purpose_radio/TV	Purpose_repairs	...
0	67	2	1169	6	False	False	False	False	True	False	...
1	22	2	5951	48	False	False	False	False	True	False	...
2	49	1	2096	12	False	False	True	False	False	False	...
3	45	2	7882	42	False	False	False	True	False	False	...
4	53	2	4870	24	True	False	False	False	False	False	...
...	...	...	...	...	...	...	...	...	...	...	...
995	31	1	1736	12	False	False	False	True	False	False	...
996	40	3	3857	30	True	False	False	False	False	False	...
997	38	2	804	12	False	False	False	False	True	False	...
998	23	2	1845	45	False	False	False	False	True	False	...
999	27	2	4576	45	True	False	False	False	False	False	...

1000 rows × 25 columns

Figure 7: Pre-Processed Data

The study implements several machine learning algorithms for loan approval prediction, including C4.5, Naïve Bayes, K-Nearest Neighbors (KNN), and Random Forest. C4.5 creates a decision tree by splitting the data based on the most informative attributes, whereas Naïve Bayes calculates the probability of each class (*good* or *bad risk*) assuming independence among features. KNN classifies applicants based on the majority class of their nearest neighbors in the feature space, and Random Forest builds multiple decision trees to make a robust prediction by majority voting. Each algorithm processes the same attributes, aiming to predict the *Risk category*, but they employ different methods for classification, offering varied insights into loan approval decision-making.

The performance evaluation in this research involves matrices used to assess the performance of the algorithms employed, such as *Precision*, *Sensitivity* and *Accuracy*.

Precision can be defined as the degree of reliability of the model when it provides a "Positive" prediction. When a piece of data is classified as "Positive," precision measures how reliably the model identifies the actual label of that data as positive.

To calculate this metric, we only need the first row of the confusion matrix. Precision is the proportion of correctly labeled "Positive" predictions out of all "Positive" predictions, or

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where $TP = True\ Positive$ and F$P = False\ Positive$.

Suppose the confusion matrix for the credit application data model is given as shown in Figure 8. From the result in the figure, the precision of the model is 75%.

This is not a very high value, as it means that, on average, for every 4 customer data points categorized by the model as credit applications, 4 - 3 = 1, so 1 customer data point is approved by the bank. As implied above, precision is an appropriate metric when false positives are highly undesirable.

Sensitivity which is shown in Figure 8 can be understood as the model's reliability in correctly detecting data labeled as positive. Sensitivity is defined as the proportion of data predicted by the model as "Positive" out of all data that is actually labeled as "Positive," or

$$Sensitivity = \frac{TP}{TP + FN} \quad (2)$$

where $TP = True\ Positive$ and FN $= False\ Negative$.

Suppose the confusion matrix for a computer vision-based credit application model is provided as shown in Figure 9. The sensitivity is about 80%. This figure is not sufficient. Because this means that on average, for every 10 people who apply for credit at the bank, there are 10–8 = 2 people who FAIL to predict "Positive" by the model. Therefore, sensitivity is an appropriate metric when false negative events are very important to avoid (high risk).

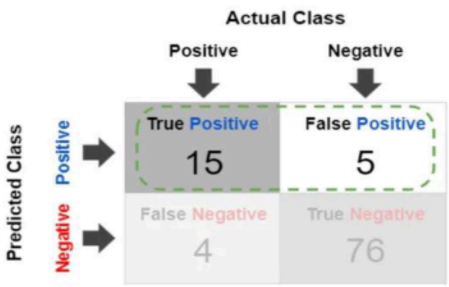

Figure 8: Submission of precision model credit data

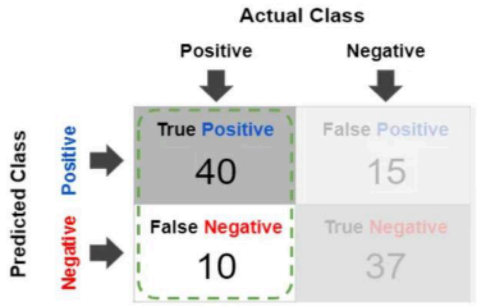

Figure 9: Submission of sensitivity model credit data

The most common or best-known accuracy metric in classification modeling as shown in Figure 10. It is the percentage of the amount of data that is predicted correctly to the total amount of data. Looking at the confusion matrix, accuracy is the ratio of the number of diagonal elements to the number of all matrix elements, or:

$$Accuracy = \frac{TP+TN}{TP\ + +FP+FN+TN} \tag{3}$$

where $TP = True\ Positive,\ TN = True\ Negative,\ FP = False\ Positive,$ and FN $= False\ Negative.$

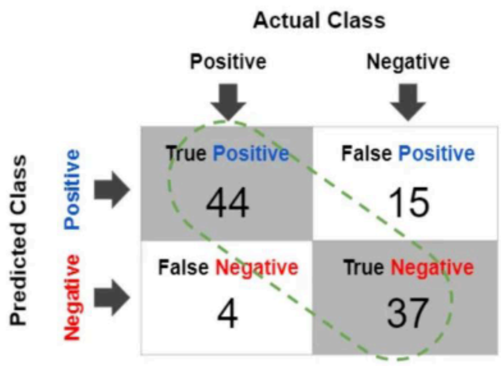

Figure 10: Submission of credit data Accuracy Model

3.2 Performance Comparison of Classification Algorithms

Accuracy in the Testing Phase in Figure 11 shows that Random Forest shows the highest accuracy (77.00%), followed by C4.5 with competitive accuracy (75.50% and 74.50%). This shows that they are relatively better at classifying both positive and negative cases. K-Nearest Neighbors, Naïve Bayes, and Decision Tree all recorded accuracies below 70%, indicating general difficulty in generalizing their learning to test data.

Figure 12 illustrates the model accuracy of various machine learning algorithms used in the study, with accuracy measured in percentage. The algorithms displayed include Random Forest, K-Nearest Neighbors (KNN), Naïve Bayes, Decision Tree, and C4.5.

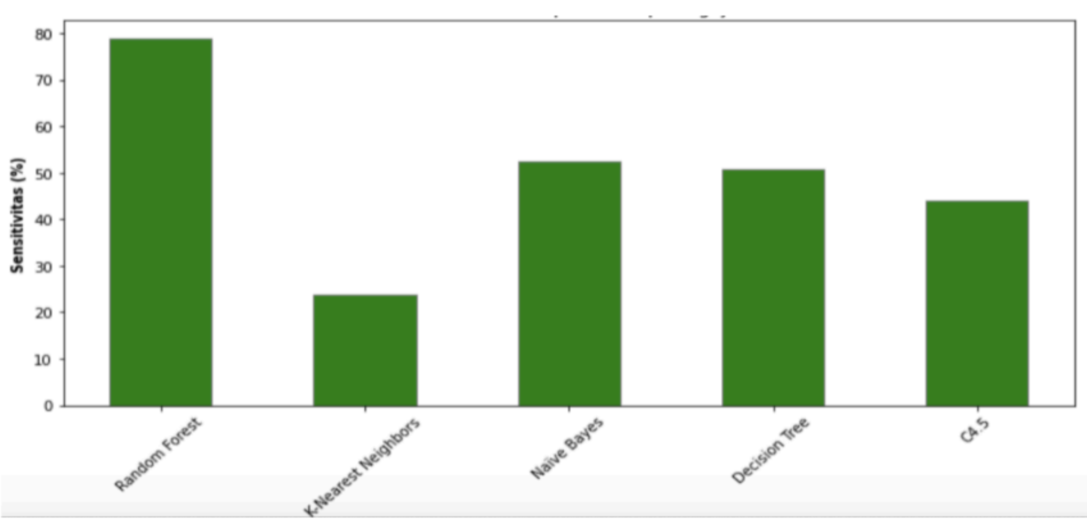

Figure 21: Comparison of Clasification Algorithms

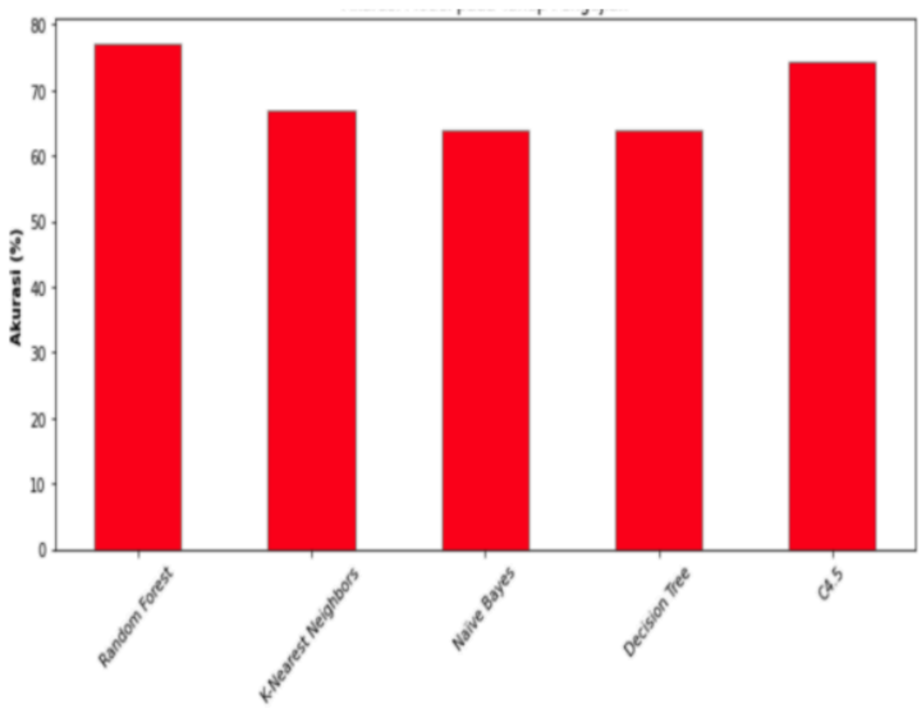

Figure 12: Model Accuracy in the Testing Phase

From the chart, it is evident that C4.5 achieved the highest accuracy, surpassing 70%. Both Random Forest and Decision Tree models also performed well, with similar accuracy levels, though slightly lower than C4.5. K-Nearest Neighbors and Naïve Bayes demonstrated the lowest accuracy, indicating that these models were less effective in classifying the credit approval data compared to the others.

This chart provides a clear comparison of how different algorithms perform in terms of accuracy, highlighting the importance of selecting the most appropriate model for credit scoring tasks.

3.3 Performance Analysis of the System in Addressing Overfitting

This study evaluates the performance of machine learning models such as Random Forest and Decision Tree, revealing significant overfitting. Overfitting occurs when a model performs excellently on training data—achieving 100% precision, sensitivity, and accuracy—but experiences a sharp decline in performance when tested on new data. In this study, Random Forest achieved 77% accuracy, and Decision Tree achieved 64%, indicating that these models became overly specific to the training data and failed to generalize effectively to unseen data. Overfitting is a common issue in predictive modeling, arising when models fit too closely to noise or irrelevant details in the training data, thus reducing their accuracy in real-world applications.

To address overfitting, several strategies can be implemented. For Decision Trees, pruning can be applied to remove parts of the tree that do not contribute to predictive power, reducing model complexity. In Random Forest, adjusting parameters such as the number of trees and the maximum depth of each tree can help optimize the balance between bias and variance, reducing overfitting. Additionally, cross-validation and using separate validation sets during training can offer a more robust evaluation of model performance, ensuring the model generalizes well. By employing these strategies, machine learning models will become more reliable and effective in real-world applications, leading to more accurate decision-making and maximizing the potential of machine learning in practical scenarios.

In contrast, the C4.5 model demonstrated a more balanced and better performance in classifying banking credit scores, with a testing accuracy of 74.50% compared to a training accuracy of 78.20%. This indicates that a more limited depth setting can lead to more stable performance and better generalization, making it a good choice for credit risk assessment in Islamic banking. This model has also not been optimized with other methods, such as hypertuning, suggesting it has the potential for even greater performance.

4.0 CONCLUSION

This research successfully designed and evaluated six machine learning algorithms to classify banking credit scores, particularly in the context of Islamic banking. The models tested include Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Naïve Bayes, and the C4.5 algorithm. The study revealed notable performance variations between the training and testing phases, specifically concerning precision, sensitivity, and accuracy. This highlighted the emerging issue of overfitting, demonstrating the necessity for models that can generalize well to new data beyond the training set.

The Random Forest and Decision Tree models showed strong performance during training but experienced significant degradation in testing accuracy, indicating considerable overfitting. In contrast, the C4.5 algorithm demonstrated a better balance between training and testing performance, achieving a testing accuracy of 74.5%. This finding suggests that models with limited depth, such as C4.5, are more effective in avoiding overfitting and may be more suitable for practical applications in credit risk assessment, particularly in Islamic banking.

On the other hand, K-Nearest Neighbors (KNN) and Naïve Bayes performed the weakest, with test accuracies of only 67.00% and 64.00%, respectively. These lower results can likely be attributed to issues such as non-homogeneous feature distribution and unmet assumptions of independence, which hindered their overall effectiveness. Therefore, this study underscores the importance of selecting and configuring models appropriately for the specific data at hand. It also emphasizes the need for strategies that address overfitting and enhance model generalization. Based on the results, the C4.5 algorithm emerges as a promising candidate for further optimization and use in Islamic banking for credit risk assessment.

ACKNOWLEDGEMENT

The authors would like to express their appreciation for the Faculty of Information Technology, Universitas Budi Luhur for the funding support to conduct the research, Department of Mechanical & Aerospace Engineering, International Islamic University Malaysia for the valuable discussion and comments for the article writing and Faculty of Data Science and Information Technology, INTI International University for the support on the programming process.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper. No financial or personal relationships that could influence the work presented in this manuscript have been identified. All authors have contributed equally to the research, and the findings presented are independent of any external influences.

REFERENCES

[1] A. Akinjole, O. Shobayo, J. Popoola, O. Okoyeigbo, and B. Ogunleye, “Ensemble-Based Machine Learning Algorithm for Loan Default Risk Prediction,” *Mathematics*, vol. 12, no. 21, Nov. 2024, doi: 10.3390/math12213423.

[2] N. Wattanakitrungroj, P. Wijitkajee, S. Jaiyen, S. Sathapornvajana, and S. Tongman, “Enhancing Supervised Model Performance in Credit Risk Classification Using Sampling Strategies and Feature Ranking,” *Big Data and Cognitive Computing*, vol. 8, no. 3, Mar. 2024, doi: 10.3390/bdcc8030028.

[3] I. Aruleba and Y. Sun, “Effective Credit Risk Prediction Using Ensemble Classifiers With Model Explanation,” *IEEE Access*, vol. 12, pp. 115015–115025, 2024, doi: 10.1109/ACCESS.2024.3445308.

[4] Y. Cheng *et al.*, “Optimized Credit Score Prediction via an Ensemble Model and SMOTEENN Integration,” in *2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 2024, pp. 355–361. doi: 10.1109/ICISCAE62304.2024.10761352.

[5] R. K. Amin, Indwiarti, and Y. Sibaroni, “Implementation of decision tree using C4.5 algorithm in decision making of loan application by debtor (Case study: Bank pasar of Yogyakarta Special Region),” in *2015 3rd International Conference on Information and Communication Technology, ICoICT 2015*, Institute of Electrical and Electronics Engineers Inc., Aug. 2015, pp. 75–80. doi: 10.1109/ICoICT.2015.7231400.

[6] A. Damayunita, R. S. Fuadi, and C. Juliane, “Comparative Analysis of Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) Algorithms for Classification of Heart Disease Patients,” *Jurnal Online Informatika*, vol. 7, no. 2, pp. 219–225, Dec. 2022, doi: 10.15575/join.v7i2.919.

[7] R. Hasan, “Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction,” *ITM Web of Conferences*, vol. 40, p. 03007, 2021, doi: 10.1051/itmconf/20214003007.

[8] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” 2002.

[9] I. Brown and C. Mues, “An experimental comparison of classification algorithms for imbalanced credit scoring data sets,” *Expert Syst Appl*, vol. 39, no. 3, pp. 3446–3453, Feb. 2012, doi: 10.1016/j.eswa.2011.09.033.

[10] Walid and Alamsyah, “Naïve Bayesian classifier algorithm and neural network time series for identification of lecturer publications in realizing internationalization of Universitas Negeri Semarang,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1742-6596/1321/3/032110.

[11] G. Louppe, “Understanding Random Forests: From Theory to Practice,” Jul. 2014, [Online]. Available: http://arxiv.org/abs/1407.7502