DOI: 10.14421/ijid.2023.3991 IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 12, No. 1, 2023, Pp. 302-313

Comparison of K-Nearest Neighbor, Support Vector Machine, Random Forest, and C 4.5 Algorithms on Indoor Positioning System

M Rizky Astari[1], Muhammad Taufiq Nuruzzaman[2,*], Bambang Sugiantoro[3]
Departement of Informatics
Universitas Islam Negeri Sunan Kalijaga
[1]rizkyasta04@gmail.com, [2]m.taufiq@uin-suka.ac.id, [3]bambang.sugiantoro@uin-suka.ac.id

Article History
Received March 3rd, 2023
Revised June 3rd, 2023
Accepted June 9th, 2023
Published June, 2023

***Abstract*— Today's most common Positioning System applied is the Global Positioning System (GPS). Positioning System is considered accurate when outdoors, but it becomes a problem when indoors making it difficult to read the GPS signal. Many academics are actively working on indoor positioning solutions to address GPS's drawbacks. Because WiFi Access Point signals are frequently employed in multiple studies, they are used as research material. This study compares the classification algorithms KNN, SVM, Random Forest, and C 4.5 to see which algorithm provides more accurate calculations. The fingerprinting method was employed in the process of collecting signal strength data in each room of the Terpadu Laboratory Building at UIN Sunan Kalijaga using 30 rooms and a total dataset of 5,977 data. The data is utilized to run experiments to determine the location using various methods. According to the experimental data, the Random Forest algorithm achieves an accuracy rate of 83%, C4.5 81%, and KNN 80%, while the SVM method achieves the lowest accuracy rate of 57%.**
Keywords*— *GPS; WIFI access point signals; classification algorithms; accurate; fingerprinting

1 INTRODUCTION

The development of technology has proven to be rampant, causing many people to utilize positioning technology in their daily lives: motorcycle taxis, bus companies, taxis, travel, and even the general public are now fond of this technology. Searching for an area or a particular address and tracking a person's geographical position is very helpful in resolving problems faced by a community.

The Global Positioning System (GPS) [1], an outdoor location technology considered accurate by the public, is the most commonly used positioning system today. GPS is precise and simple to use, as practically all gadgets regularly used by most people already incorporate GPS without additional expense. The problem is that GPS can pinpoint the location of a stationary item on Earth with an accuracy of 10 meters or more [2]. However, if the device is positioned indoors, it will surely be difficult to read by GPS because GPS signals cannot efficiently penetrate walls. The GPS receiver will fail to detect a strong enough signal to be indoors [3]. GPS, for example, will not work effectively indoors.

Because the GPS [4] signal cannot function correctly indoors, another signal that may penetrate the room, such as a WIFI signal, is required[5]. The mobile device may read its position by detecting the signal strength emitted by the access points that are typically used to share internet access with users by harnessing the signal from WIFI access points placed in the room or building[6]. While privacy problems persist, such use is beneficial in the event of a building problem, such as a fire or evacuation. Furthermore, cellular signals can be used as a location determinant or location name that shows on the screen of GSM and CDMA mobile handsets by capturing signals from BTS-BTS [7]. Another requirement is the ability to locate someone using their mobile device in a building, whether multi-storey or not, without having to phone first. Because the mobile device delivers real-time location data to the central computer, anyone can enjoy it. As a result, knowledge growth in indoor positioning search began with the notion of the Indoor Positioning System [8].

Many academics are actively working on indoor positioning solutions to address GPS's drawbacks. Wireless technologies such as ZigBee, Bluetooth, WIFI, and cell tower signals are used in this system [9]. However, because it is the most ubiquitous and does not require any additional gear, WIFI technology is most commonly employed as study material.

Many researchers conducted research and previous studies to prove the benefits of the Indoor Positioning System and WIFI Access Point [10], and to achieve good accuracy, the researchers developed the concept of Indoor Positioning System using several techniques; for example, many are found using the KNN (K-Nearest Neighbor) method, NN, and others [11]. Existing research indicates that KNN clustering gives the most data gathering and the highest accuracy for location detection problems, but there are some discrepancies in the results [12].

The author's research intends to examine numerous classification algorithms with guided learning, notably KNN, Support Vector Machine, Random Forest, and C45, in order to determine whether there are ways other than the KNN method that can deliver greater and better accuracy outcomes. The data used in this study were collected on one of the buildings on the UIN Sunan Kalijaga Yogyakarta campus using self-developed Android tools. As a result, this study is expected to add to the analysis results by comparing the level of accuracy with many algorithms and can be used as a reference in selecting the optimum algorithm for establishing an interior positioning system based on Wi-Fi access points.

2 METHOD

Figure 1 depicts the three stages of this research: dataset creation, testing, and analysis.

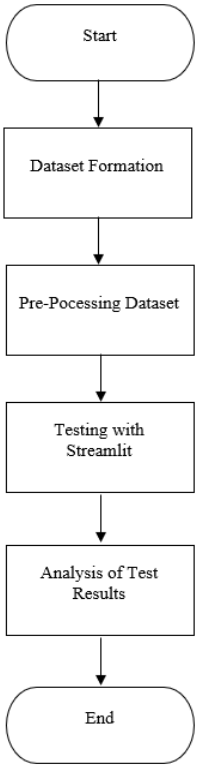

Figure 1. Research stages

2.1. Dataset Formation

Figure 2 depicts the stages of this procedure, which include Android-based application creation, data gathering, data cleaning and labelling, and dataset conversion.

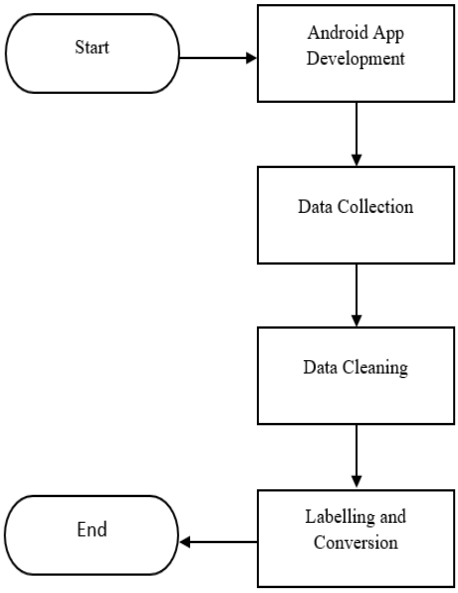

Figure 2. Steps of dataset formation

The fingerprinting approach is used to capture data in this procedure, which is carried out on an Android OS-based smartphone device. In the study, the author created Check Signal, an Android-based application tool that can scan all access points read/reachable by a smartphone. This tool displays information in the form of manually supplied room names, access point (AP) names, weather conditions, signal strength, mac Address, room latitude-longitude coordinates, date, and time, which can then be saved in the database. Each room entered is scanned one by one by the application to find RSSI (Radio Signal Strength Indication) readings that differ from one another for each room. Figure 3 depicts the display of the signal check application tools.

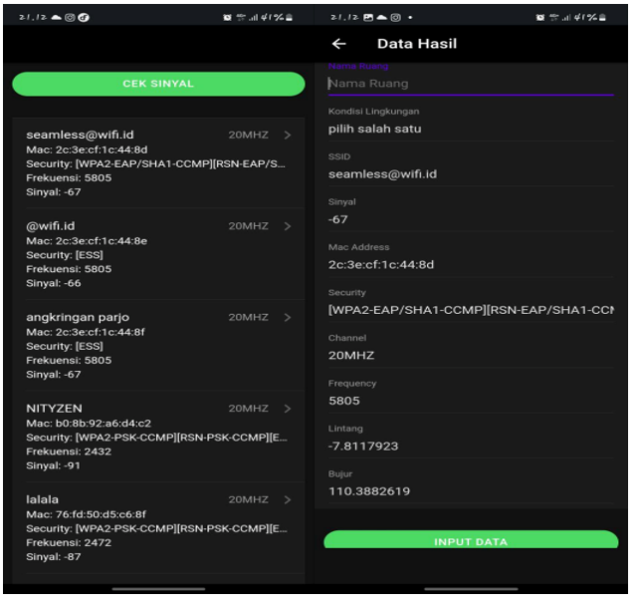

Figure 3. Display of the signal check tool used for data capture

The steps in using the "Signal Check" application in data collection are:

1. Take a position in the room where the data will be collected.
2. Run the Signal Check application. The application will automatically scan the list of access points scattered within the room.
3. Fill in the room column manually.
4. Press the save button to save the data into the database.

A sampling of WIFI access point data was carried out at the Integrated Lab Building of the Sunan Kalijaga State Islamic University, Yogyakarta, which has four floors in one building using the self-developed signal check application tool, it obtained 5,977 data as the total from 30 rooms.

The schematics of the rooms on each floor of the integrated lab building of UIN Sunan Kalijaga can be seen in Figures 4, 5, 6 and 7.

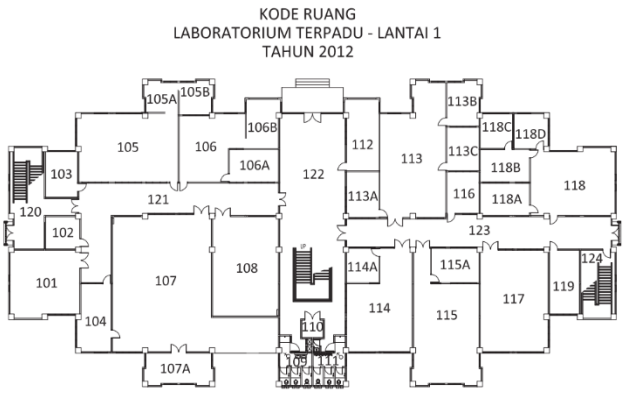

Figure 4. Schematic of the 1st floor of the integrated lab of UIN Sunan Kalijaga

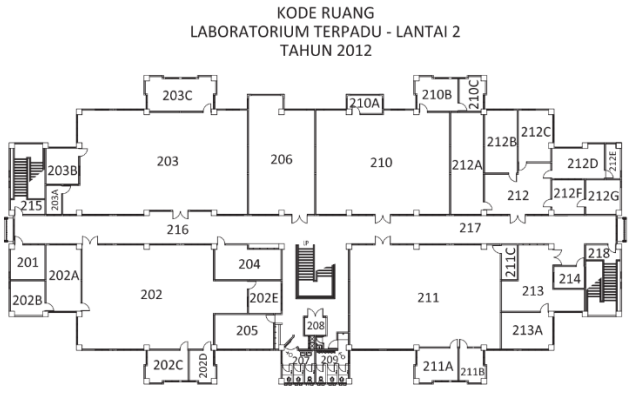

Figure 5. Schematic of the 2nd floor of the integrated lab of UIN Sunan Kalijaga

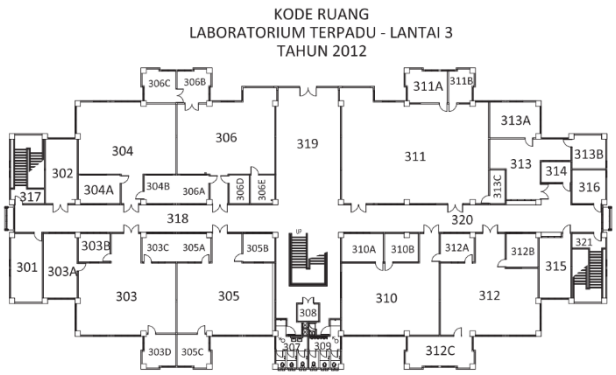

Figure 6. Schematic of the 3rd floor of the integrated lab of UIN Sunan Kalijaga

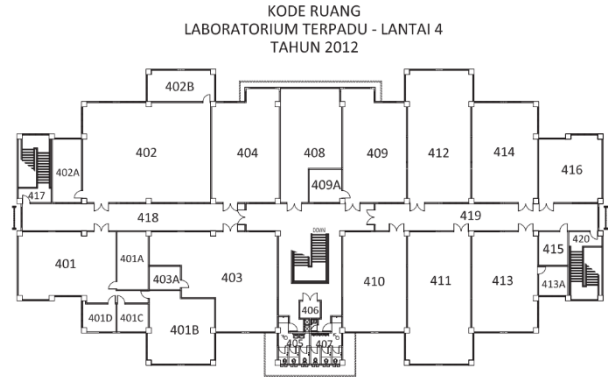

Figure 7. Schematic of the 4th floor of the integrated lab of UIN Sunan Kalijaga

Data sampling carried out in around two weeks by altering employee working hours in the integrated lab facility was divided into two; namely training data collection and testing data collection. There are two types of test data, namely the same test data as training data (same retrieval time) and random test data (retrieval time is ignored) for training data, which is taken by the restrictions of the problem, which is carried out at 10.00-14.00 WIB.

2.2. Data Pre-Processing

Data pre-processing done in this research includes label encoding and column dropping. Label encoding is used to convert word-shaped labels into numeric which refers to the process of transforming word labels into numeric. In this research, the weather column in the form of words (sunny, cloudy) is converted into a numeric label with the conversion of sunny = 1.00 and cloudy = 0.50. Meanwhile, column dropping is done to eliminate columns that are not needed during the testing process. In this case, the dropped columns are the space name and signal columns. The signal column is dropped because it is a class column.

2.3. Testing

The results of data collection are then tested to determine the accuracy value using several methods, namely KNN with parameter setting k with a range of 1-100, Support Vector Machine (SVM) with parameter setting c with a range of 0.01-10.00, Random Forest with parameter setting max_depth with a range of 1-150 and n_estimator with a range of 1-100 and C45 with default parameter settings. The algorithm is calculated using a web-based Streamlit application using the Phyton language which will be able to display the calculation results automatically from the dataset being tested. So that later a comparison can be made, which algorithm is better in accuracy based on the results obtained from the algorithm.

The comparison in this study uses five algorithms, namely KNN, SVM, Random Forest, and C45.

2.3.1. *K-Nearest Neighbor:* It is a classification algorithm with learning by example or training data based on a distance function for pairs of observations, such as Euclid distance and Manhattan distance [13]. The basic idea of KNN is straightforward. In the KNN classification paradigm from the first retrieved test sample, the similarities between the test sample and its k nearest neighbors are aggregated according to the classes of its neighbors, and the test sample is assigned to the most similar class [14]. The best choice of *k* adapts to the data; generally, larger values of *k* reduce the effect of noise on classification but create less clear boundaries between classes [15]. Various heuristic values, such as cross-validation, can determine a good *k* [16]. The particular case where the class is expected to be the class of the closest training sample (i.e., when *k*=1) is called the Nearest Neighbor (NN) algorithm [17]. One of the advantages of the KNN algorithm is that it is easy to implement and can be used to classify data on non-linear datasets. However, the KNN algorithm also has disadvantages, which are sensitive to noise values (outliers) and requires a long time to classify large datasets. In the context of Indoor Positioning System, the KNN can be utilized to predict the user's location by comparing the Wi-Fi signal from the user with the signal recorded in the database [18]. KNN requires training data consisting of a set of previously recorded data. Each of these data consists of two parts: the measured Wi-Fi signal, and the position label. In the testing phase, the KNN algorithm will search for K Nearest Neighbors of the measured Wi-Fi signal. The value of k can be predefined. Then, by taking the majority class of the nearest neighbors, the KNN algorithm will predict the user's position.

2.3.2. *Support Vector Machine:* SVM [19] is a classification method for supervised learning. On a set of training data with several *p* attributes (*p*-dimensional vectors), the SVM method tries to get a (*p*-1) dimensional hyperplane that can divide the training data according to its class [20][21]. Parameter C in SVM is a variable that controls the

trade-off between training error and model complexity. C is responsible for determining the penalty applied to classification errors in the training data. The larger the value of C, the stronger the penalty applied to classification errors, and the more complex the resulting model. Conversely, the smaller the value of C, the weaker the penalty applied to classification errors, and the simpler the resulting model [22]. Thus, if the value of C is too large, the SVM model can experience overfitting, that is, the model is too complex and too well matched to the training data, so it cannot predict well on data that has not been seen before. Conversely, if the value of C is too small, the SVM model can experience underfitting, that is, the model is too simple and does not match the training data, so it cannot predict well on data that has not been seen. One of the advantages of the SVM algorithm is that it can be used for non-linear datasets and has good generalization capabilities, making it suitable for use on complex datasets. However, the SVM algorithm also has disadvantages, namely long computation time for large and complex datasets. An illustrative picture describing the hyperplane (commonly called the decision boundary) of a set of training data can be seen in Figure 8.

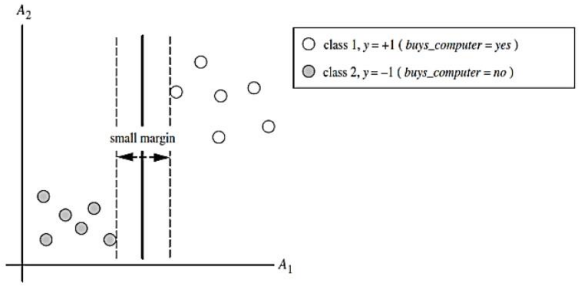

Figure 8. Example decision boundary for some training data with low distance/difference difference

2.3.3. *Random Forest:* The random forest algorithm is a development of the CART [23] method by adopting the bootstrap aggregating (bagging) method and random feature selection. In a random forest [24], several trees are grown to form a forest, then analyze the collection of trees [25]. In a group of data consisting of n observations and *p* explanatory variables, random forest is done by [26] the following steps:

- The first stage is the bootstrap stage, where a random sample of size *n* is run and the cluster data is recovered.
- Next, utilizing the bootstrap samples, the tree is built up to its maximum size (without pruning). At each node, parser selection is made through a random selection of m explanatory variables, where $m << p$. The best parser is selected from these *m* explanatory variables. Explanatory variables. This stage is a random feature selection stage.
- Next, repeat steps 1 and 2 *k* times, until a forest containing *k* trees is formed.

In the Random Forest algorithm [27], there are several parameters that can be set to control the machine learning behavior. The two main parameters in Random Forest are max depth and n estimators. Max Depth is a parameter that determines the maximum depth of each decision tree in Random Forest. The deeper the decision tree, the more complex the model and the greater the chance of overfitting. Therefore, the maximum depth should be chosen carefully so as not to produce an overfitting or overly simple model. If the max depth is set too high, the model may memorize the training data and cannot generalize well to new data [28]. Conversely, if the max depth is too low, the model will be too simple and unable to capture complex patterns in the data. N_estimator is a parameter that determines the number of decision trees in Random Forest. The more decision trees used, the more robust and stable the resulting model will be. However, using too many decision trees can also result in overfitting the model and taking longer training time. Therefore, the number of decision trees used needs to be chosen carefully to achieve a balance between accuracy and training time.

2.3.4. *C4.5:* The C4.5 [29] algorithm can be applied in classification through the formation of a decision tree from the provided training data [30]. The decision tree is a famous classification and prediction method [31], useful for extrapolating data, and detecting hidden relationships between a set of candidate input variables and target variables [32]. An example of a decision tree can be seen in Figure 4.

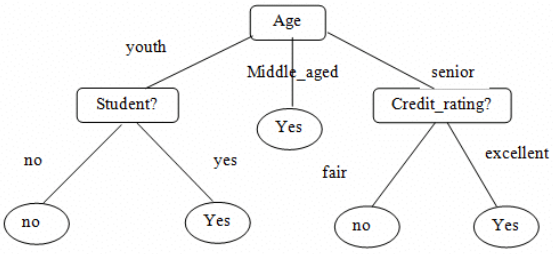

Figure 9. Example Decision Tree C45

According to [33] the following are the stages required in the formation of a decision tree:

- Checking the base case. The data read in the C4.5 algorithm is data that contains attributes and class labels. Attributes are usually numeric

or categorical variables, while class labels are the values to be predicted.

- The decision tree is built by calculating the information gain of each attribute. Information gain is the difference between entropy before and after separating data based on attributes. The attribute that has the highest information gain will be selected as the attribute used in decision making. After the attribute is selected, the data will be separated into several parts based on the value of the attribute. This process is repeated recursively until all data is classified.
- After the decision tree is built, a pruning process is performed to reduce overfitting of the model. Overfitting occurs when the model is too complex and is only suitable for training data, but cannot be generalized well to new data. Pruning is done by removing insignificant branches of the decision tree.
- After the decision tree is built and pruned, the C4.5 algorithm will convert it into a rule set. Each rule is a combination of several attributes that must be met to get the appropriate class.

2.4. Analysis

This stage examines the outcomes of testing with the KNN, SVM, Random Forest, and C45 algorithms. The findings of the analysis will be used to provide conclusions and recommendations. Using a confusion matrix, this examination determines whether weather conditions can impair the accuracy of the indoor positioning system.

3 RESULT AND DISCUSSION

3.1 Data Collection Results

The fingerprinting approach was used to collect data for two weeks, and the total overall data gathered during the collecting procedure was obtained from 7,684 datasets from 34 rooms in the integrated lab building of UIN Sunan Kalijaga. Some data was destroyed after review because there were problems while inputting room names or comparable data in one room at the same time, and there were four rooms that could not be consistent in data collecting for two weeks. After the inspection and deletion steps are completed, training data totalling 4,781 data is received from 30 rooms that are consistent when taking data, and testing data totaling 1196 data is obtained. The data is collected utilizing an Asus mobile smartphone device and the Android-based Signal Check application utility.

Only in a few laboratory rooms can data be collected consistently. Initially, the Integrated Laboratory building of UIN Sunan Kalijaga housed 34 rooms. However, only 30 rooms can collect data on a daily basis. Some rooms are sometimes inaccessible either because they are locked or they are in use. As a result, these rooms were removed from the database.

When data is collected, there are many access points at that location, including outside access points around the building, such as access points installed in the Faculty of Science and Technology building, as well as access points originating from internet cafes, boarding houses, hospitals, and hotels, so that when collecting data for access points belonging to different buildings that cannot be detected for specific rooms, the signal strength research is performed.

Table 1 shows the outcomes of data filtering and an example format of signal strength data received from data gathering.

Table 1. Example of Signal Strength Data Format

Room Name	AP1	AP2	AP3		AP23	Weather
Ruang1105	-46	-60	-58	.....	-60	Sunny
Ruang1106	-80	-55	-63	.....	-81	Sunny
Ruang1107	-85	-92	-80	.....	-85	Cloudy
.....	.....	.....	.....	.....	.....	.....
Ruang4416	-65	-69	-83	.....	-75	Sunny

Each row of data has 30 columns with 249 rows, where the first column is the room's name, while the following columns are signal strength columns and weather columns obtained from all access points on the UIN Suka campus. The signal strength is in dBm.

3.2 Pre-processing Data

In the previous explanation, the data pre-processing was carried out in two stages: label encoding and column dropping using Phyton combined with the Streamlit platform. At the label encoding stage, the weather label is changed from word form to numeric form as in Figure 10 of the original data and 11 data after preprocessing.

Figure 10 shows the original data before pre-processing the encoding label data, where the weather column is still in the form of words such as sunny and cloudy, then the weather label is encoded into numerical form, namely sunny with a value of 1.00, cloudy with a value of 0.55.

	NamaRuang	Cuaca	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	AP9	AP1
0	Ruang1105	Cerah	-75	-87	-87	-83	-83	-91	-86	-77	-79	-8
1	Ruang1105	Cerah	-43	-46	-60	-46	-60	-58	-51	-59	-66	-5
2	Ruang1105	Cerah	-69	-55	-45	-50	-69	-50	-68	-51	-54	-6
3	Ruang1105	Cerah	-85	-70	-78	-74	-83	-81	-81	-56	-62	-5
4	Ruang1105	Mendung	-80	-70	-79	-79	-43	-69	-55	-45	-50	-6
5	Ruang1105	Cerah	-87	-86	-77	-80	-86	-86	-85	-80	-80	-9
6	Ruang1105	Cerah	-46	-60	-58	-51	-60	-66	-58	-66	-83	-8
7	Ruang1106	Cerah	-55	-45	-53	-69	-53	-68	-50	-54	-63	-6
8	Ruang1106	Cerah	-84	-86	-88	-79	-83	-83	-84	-81	-57	-5
9	Ruang1106	Cerah	-75	-89	-65	-62	-62	-53	-87	-48	-48	-8

Figure. 10 Data before preprocessing

	NamaRuang	Cuaca	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	AP9	AP10
0	Ruang1105	1	-75	-87	-87	-83	-83	-91	-86	-77	-79	-86
1	Ruang1105	1	-43	-46	-60	-46	-60	-58	-51	-59	-66	-58
2	Ruang1105	1	-69	-55	-45	-50	-69	-50	-68	-51	-54	-63
3	Ruang1105	1	-85	-70	-78	-74	-83	-81	-81	-56	-62	-57
4	Ruang1105	0.5	-80	-70	-79	-79	-43	-69	-55	-45	-50	-69
5	Ruang1105	1	-87	-86	-77	-80	-86	-86	-85	-80	-80	-93
6	Ruang1105	1	-46	-60	-58	-51	-60	-66	-58	-66	-83	-89
7	Ruang1106	1	-55	-45	-53	-69	-53	-68	-50	-54	-63	-67
8	Ruang1106	1	-84	-86	-88	-79	-83	-83	-84	-81	-57	-58
9	Ruang1106	1	-75	-89	-65	-62	-62	-53	-87	-48	-48	-82

Figure 11. Data after preprocessing

Figure 11 shows the results after pre-processing the encoding label data where the weather label has changed into numeric form. Furthermore, the dropping column is carried out, namely the room name column and signal status. After dropping the space name and signal columns, the data becomes as shown in Figure 12.

	Cuaca	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	AP9	AP10	AP11	AP12
0	1	-75	-87	-87	-83	-83	-91	-86	-77	-79	-86	-87	-86
1	1	-43	-46	-60	-46	-60	-58	-51	-59	-66	-58	-66	-83
2	1	-69	-55	-45	-50	-69	-50	-68	-51	-54	-63	-67	-51
3	1	-85	-70	-78	-74	-83	-81	-81	-56	-62	-57	-44	-45
4	0.5	-80	-70	-79	-79	-43	-69	-55	-45	-50	-69	-50	-68
5	1	-87	-86	-77	-80	-86	-86	-85	-80	-80	-93	-80	-82
6	1	-46	-60	-58	-51	-60	-66	-58	-66	-83	-89	-77	-84
7	1	-55	-45	-53	-69	-53	-68	-50	-54	-63	-67	-50	-67
8	1	-84	-86	-88	-79	-83	-83	-84	-81	-57	-58	-58	-43
9	1	-75	-89	-65	-62	-62	-53	-87	-48	-48	-82	-82	-60

Figure 12. Results of dropping the column

3.3 Data Testing and Analysis

At this stage, testing is carried out using Phyton as a programming language and Streamlit as a place to display test results with several algorithms namely K Nearest neighbor, Support Vector Machine, Random Forest, and C 4.5 with the same data and using the same preprocessing techniques.

3.3.1 *K-Nearest Neighbor:* The results of data testing on the KNN algorithm obtained good results. As shown in Figure 13 which shows the confusion matrix of the KNN algorithm obtained an accuracy of 0.80 with the parameter k value used k = 1. In testing with parameter k = 3, the accuracy value is 0.64, and k = 5 is 0.62 as in Table 2. Thus, the confusion matrix shown in Figure 13 is the highest accuracy value with parameter k = 1. In addition to accuracy, to determine the performance of this algorithm, measurements are also made using precision, recall, and f1 score with parameter k = 1 which it can be seen in Table 3, from the KNN algorithm getting the same value of 0.77.

Table 2. Test Results of Several KNN Parameters

Parameters	Accuracy
K = 1	80%
K = 3	64%
K = 5	62%

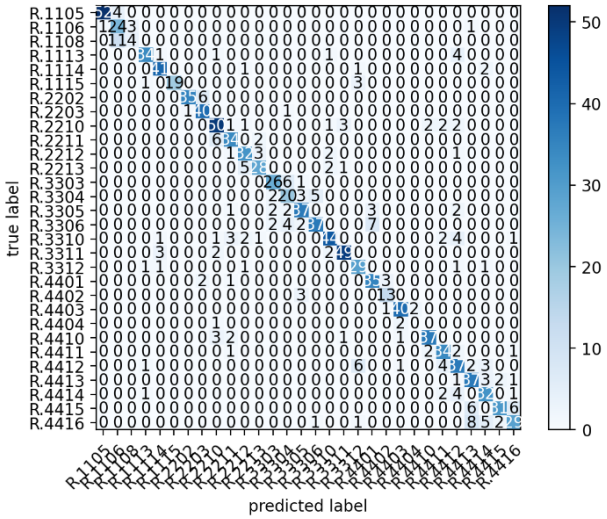

Figure 13. KNN's *c*onfusion *m*atrix

3.3.2 *Support Vector Machine:* Testing data using the SVM algorithm gives good results. Figure 14 shows the confusion matrix for the SVM algorithm. The accuracy obtained by the SVM model is 0.56 with a parameter value of C = 10. In testing with parameter C = 1.00, the accuracy value is 0.57, and C = 0.1 is 0.0,8 as in Table 4. So, the confusion matrix displayed in Figure 9 is the highest accuracy value with parameter C = 10. In addition to accuracy, measurement of the performance of this algorithm is done using precision, recall, and f1-score with the values shown in Table 5, the precision of the SVM algorithm obtained the highest value among others, which is 0.89.

Table 3. KNN's Algorithm Performance

Class	Precision	Recall	F1 Score	Amount of Data
Ruang1105	0.98	0.93	0.95	56
Ruang1106	0.62	0.83	0.71	29
Ruang1108	0.57	0.27	0.36	15
Ruang1113	0.89	0.83	0.86	41
Ruang1114	0.87	0.91	0.89	45
Ruang1115	1.00	0.83	0.90	23
Ruang2202	0.97	0.85	0.91	41
Ruang2203	0.83	0.95	0.89	42
Ruang2210	0.78	0.81	0.79	62
Ruang2211	0.77	0.81	0.79	42

Ruang2212	0.76	0.82	0.79	39
Ruang2213	0.82	0.78	0.80	36
Ruang3303	0.81	0.79	0.80	33
Ruang3304	0.61	0.67	0.63	30
Ruang3305	0.80	0.79	0.80	47
Ruang3306	0.86	0.71	0.78	52
Ruang3310	0.85	0.75	0.79	59
Ruang3311	0.91	0.88	0.89	56
Ruang3312	0.72	0.85	0.78	34
Ruang4401	0.78	0.85	0.81	41
Ruang4402	0.76	0.81	0.79	16
Ruang4403	0.91	0.93	0.92	43
Ruang4404	0.00	0.00	0.00	3
Ruang4410	0.90	0.84	0.87	44
Ruang4411	0.77	0.85	0.81	41
Ruang4412	0.64	0.69	0.66	54
Ruang4413	0.69	0.84	0.76	44
Ruang4414	0.70	0.80	0.74	40
Ruang4415	0.89	0.72	0.79	43
Ruang4416	0.74	0.63	0.68	46
Average	0.77	0.77	0.77	1196

3.3.3 Support Vector Machine: Testing data using the SVM algorithm gives good results. Figure 14 shows the confusion matrix for the SVM algorithm. The accuracy obtained by the SVM model is 0.56 with a parameter value of C = 10. In testing with parameter C = 1.00, the accuracy value is 0.57, and C = 0.1 is 0.0,8 as in Table 4. So, the confusion matrix displayed in Figure 9 is the highest accuracy value with parameter C = 10. In addition to accuracy, measurement of the performance of this algorithm is done using precision, recall, and f1-score with the values shown in Table 5, the precision of the SVM algorithm obtained the highest value among others, which is 0.89.

Table 4. Test Results of Several SVM Parameters

Parameters	Accuracy
C = 10	56%
C = 1.00	57%
C = 0.1	0.8%

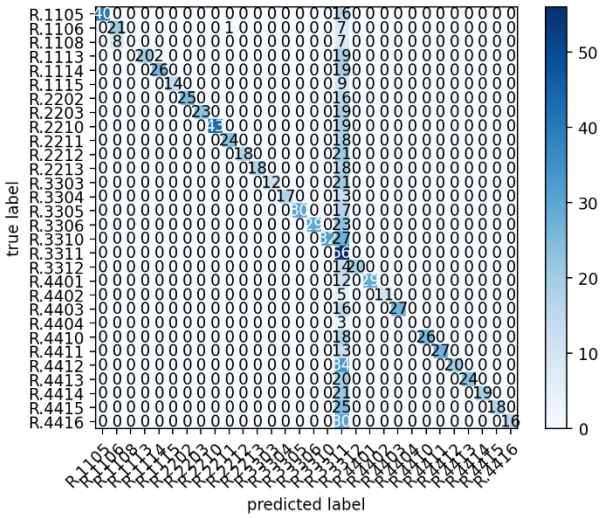

Figure 14. SVM's Confusion Matrix

Table 5. SVM's Algorithm Performance

Class	Precision	Recall	F1 Score	Amount of data
Ruang1105	1.00	0.71	0.83	56
Ruang1106	0.72	0.72	0.72	29
Ruang1108	0.00	0.00	0.00	15
Ruang1113	1.00	0.49	0.66	41
Ruang1114	0.93	0.58	0.71	45
Ruang1115	1.00	0.61	0.76	23
Ruang2202	1.00	0.61	0.76	41
Ruang2203	1.00	0.55	0.71	42
Ruang2210	1.00	0.69	0.82	62
Ruang2211	0.96	0.57	0.72	42
Ruang2212	1.00	0.46	0.63	39
Ruang2213	1.00	0.50	0.67	36
Ruang3303	1.00	0.56	0.72	52
Ruang3304	1.00	0.57	0.72	30
Ruang3305	1.00	0.64	0.78	47
Ruang3306	1.00	0.56	0.72	52
Ruang3310	1.00	0.54	0.70	59
Ruang3311	0.10	1.00	0.18	56
Ruang3312	1.00	0.59	0.74	34
Ruang4401	1.00	0.71	0.83	41
Ruang4402	1.00	0.69	0.81	16
Ruang4403	1.00	0.63	0.77	43
Ruang4404	0.00	0.00	0.00	3
Ruang4410	1.00	0.59	0.74	44
Ruang4411	1.00	0.70	0.82	40
Ruang4412	1.00	0.37	0.54	54
Ruang4413	1.00	0.55	0.71	44
Ruang4414	1.00	0.47	0.64	40
Ruang4415	1.00	0.42	0.59	43

Ruang4416	1.00	0.35	0.52	46
Average	0.89	0.54	0.64	1196

3.3.4 Random Forest: The results of data testing on the Random Forest algorithm obtained good results. As shown in Figure 15, the confusion matrix of the Random Forest algorithm obtained an accuracy of 0.83 with a parameter value of max_depth 20 and n_estimator 25. In addition to accuracy, to determine the performance of this algorithm, measurements are also made using precision, recall, and f1 score which can be seen in Table 6, from the random forest algorithm getting the same value of 0.81.

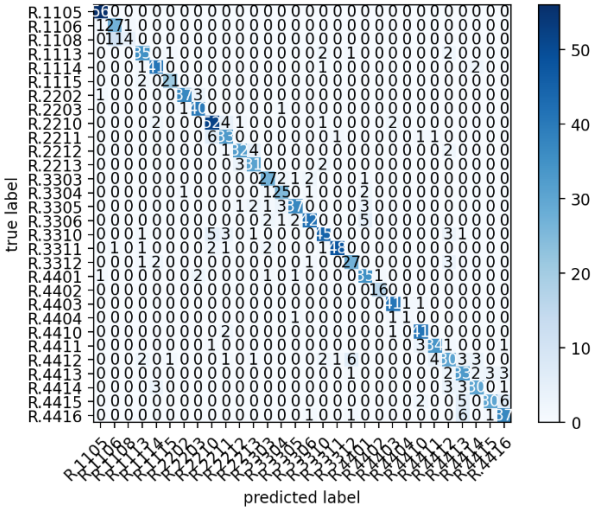

Figure 15. Random Forest's confusion matrix

Table 6. Random Forest's Algorithm Performance

Class	Precision	Recall	F1 Score	Amount of data
Ruang1105	0.93	1.00	0.97	56
Ruang1106	0.68	0.90	0.78	29
Ruang1108	0.80	0.27	0.40	15
Ruang1113	0.83	0.83	0.83	41
Ruang1114	0.89	0.91	0.90	45
Ruang1115	0.91	0.91	0.91	23
Ruang2202	0.97	0.90	0.94	41
Ruang2203	0.87	0.95	0.91	42
Ruang2210	0.81	0.84	0.83	62
Ruang2211	0.70	0.79	0.74	42
Ruang2212	0.86	0.79	0.83	39
Ruang2213	0.76	0.89	0.82	36
Ruang3303	0.81	0.88	0.84	33
Ruang3304	0.78	0.83	0.81	30
Ruang3305	0.86	0.79	0.82	47
Ruang3306	0.93	0.79	0.85	52
Ruang3310	0.85	0.76	0.80	59
Ruang3311	0.92	0.88	0.90	56
Ruang3312	0.72	0.16	0.74	34
Ruang4401	0.78	0.88	0.83	41
Ruang4402	0.94	1.00	0.97	16
Ruang4403	0.91	0.93	0.92	43
Ruang4404	0.00	0.00	0.00	3
Ruang4410	0.87	0.91	0.89	44
Ruang4411	0.88	0.88	0.88	40
Ruang4412	0.67	0.59	0.63	54
Ruang4413	0.72	0.75	0.73	44
Ruang4414	0.82	0.80	0.81	40
Ruang4415	0.89	0.77	0.82	43
Ruang4416	0.78	0.83	0.80	46
Average	0.81	0.80	0.80	1196

3.3.5 *C 4.5:* Testing data on the C 4.5 algorithm gets good results. Figure 16 shows the confusion matrix with the C 4.5 algorithm and obtained an accuracy of 0.81 with the default parameter value. To measure the performance of this algorithm in addition to accuracy, measurements are made using precision, recall, and f1 score which can be seen in Table 7, the precision of the C 4.5 algorithm gets the same value of 0.81.

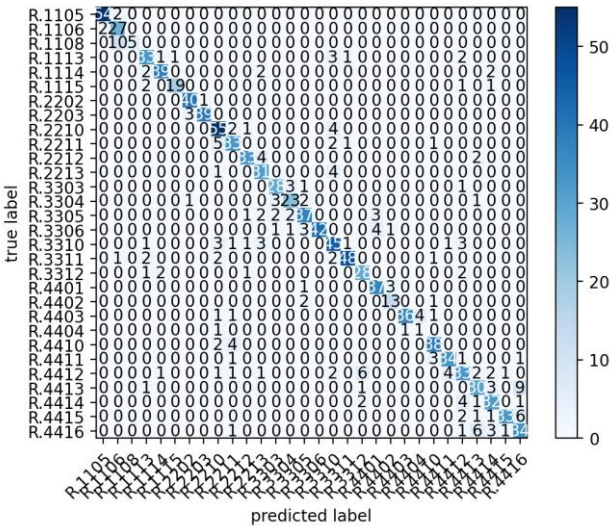

Figure 16. C 4.5's confusion matrix

Table 7. C 4.5's Algorithm Performance

Class	Precision	Recall	F1 Score	Amount of data
Ruang1105	0.96	0.96	0.96	56
Ruang1106	0.68	0.93	0.78	29
Ruang1108	1.00	0.33	0.50	15
Ruang1113	0.79	0.83	0.81	41
Ruang1114	0.91	0.87	0.89	45
Ruang1115	0.95	0.83	0.88	23
Ruang2202	0.93	0.98	0.95	41

Ruang2203	0.97	0.93	0.95	42
Ruang2210	0.76	0.87	0.81	62
Ruang2211	0.78	0.83	0.80	42
Ruang2212	0.89	0.87	0.88	39
Ruang2213	0.71	0.83	0.77	36
Ruang3303	0.75	0.82	0.78	33
Ruang3304	0.79	0.73	0.76	30
Ruang3305	0.81	0.81	0.81	47
Ruang3306	0.93	0.79	0.85	52
Ruang3310	0.73	0.75	0.79	34
Ruang3311	0.94	0.84	0.89	56
Ruang3312	0.75	0.79	0.77	34
Ruang4401	0.82	0.90	0.86	41
Ruang4402	0.81	0.81	0.81	16
Ruang4403	0.97	0.84	0.90	43
Ruang4404	0.20	0.33	0.25	3
Ruang4410	0.81	0.86	0.84	44
Ruang4411	0.90	0.88	0.89	40
Ruang4412	0.70	0.61	0.65	54
Ruang4413	0.68	0.73	0.70	44
Ruang4414	0.64	0.75	0.69	40
Ruang4415	0.97	0.77	0.86	43
Ruang4416	0.67	0.65	0.66	46
Average	0.81	0.79	0.79	1196

3.4 Analysis

At this stage, testing is carried out using Phyton as a programming language and Streamlit as a place to display test results with several algorithms namely K Nearest neighbor, Support Vector Machine, Random Forest, and C 4.5 with the same data and using the same preprocessing techniques resulting in an accuracy level such as Table 8.

Table 8. Classification Algorithm Comparison Results with Several Parameters

Algorithm	Parameters Selected	Accuracy
KNN	k = 1	80%
KNN	k = 3	64%
KNN	k = 5	62%
SVM	C = 0.1	8%
SVM	C = 1	57%
SVM	C = 10	56%
Random Forest	Max_depth = 12 N_estimator = 16	83%
C45	*default*	81%

The KNN algorithm with a parameter value of k = 1 gets an accuracy value of 80% when compared to parameter k = 3 by 64% and parameter k = 5 by 62%. The support vector machine algorithm using the parameter value C = 1.00 gets an accuracy value of 57% when compared to parameter C = 10 by 56% and parameter C = 0.1 by 8%. The random forest algorithm with a max_depth parameter of 20 and n_estimator 25 obtained an accuracy value of 83%, and the C 4.5 algorithm with default parameters obtained an accuracy value of 81%.

The Random Forest algorithm gets the highest accuracy with the same accuracy value of 83% followed by C 4.5 with a value of 81% and KNN with a value of 80% with the SVM algorithm which gets the lowest accuracy with a value of 57%. In the SVM algorithm used in this study using the parameter C = 0.1 to 10.00 so that the results obtained are not good as it is the smallest of the other algorithms, there is still a possibility for the SVM algorithm to get a better level of accuracy. This research tests the accuracy of classification accuracy in several methods using the dataset obtained at the data collection stage. The features used for classification are all columns, so there is a possibility of a cause of dimensionality, which makes the accuracy level to decrease.

4 CONCLUSION

Based on the above discussion, the data retrieval process requires further steps to check and clean the data set from errors that are usually made, namely inaccuracy when determining the name of the room in the Android tools used. The Random Forest algorithm can produce the highest accuracy rate of 83% with the parameters used max_depth 20 and n_estimator 25, followed by the C 4.5 algorithm with default parameters getting an accuracy value of 81%, KNN with the parameter k used k=1 80% highest of the other parameters, and SVM with the parameter used C=1.00 with an accuracy value of 57%. And The methods were compared in this research and the study was implemented with minimal change to the initial configuration, so it is possible to improve the accuracy level by changing the configuration and performing pre-processing on the dataset before the classification stage.

AUTHOR'S CONTRIBUTION

M Rizky Astari is the first author who conducted a literature review of previous research, data collection, research ideas, testing, analysis, and implementation, while Muhammad Taufiq Nuruzzaman as the second author provided suggestions and input on the research concept. and Bambang Sugiantoro as the third author provided input in writing the draft.

COMPETING INTERESTS

In accordance with the journal's publication ethics, M Rizky Astari, Muhammad Taufiq Nuruzzaman, and Bambang Sugiantoro (the article's authors), declare that there are no competing or conflicting interests in their work (CI).

ACKNOWLEDGMENT

The author expresses gratitude to all those who helped bring this research to fruition, including friends, family, and professors who shared their expertise and offered advice and encouragement.

REFERENCES

[1] C. J. Hegarty, "The Global Positioning System (GPS) BT - Springer Handbook of Global Navigation Satellite Systems," P. J. G. Teunissen and O. Montenbruck, Eds. Cham: Springer International Publishing, 2017, pp. 197–218. doi: 10.1007/978-3-319-42928-1_7.

[2] P. A. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning," *Trans. GIS*, vol. 13, no. SUPPL. 1, pp. 5–25, 2009, doi: 10.1111/j.1467-9671.2009.01152.x.

[3] F. H. Perdana and H. Ginardi, "Implementasi Indoor Positioning System Berbasis Smartphone dengan Penambahan Access Point untuk Studi Kasus Gedung Teknik Informatika ITS," *J. Tek. ITS*, vol. 5, no. 2, 2016, doi: 10.12962/j23373539.v5i2.17047.

[4] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-grade global positioning system (GPS) accuracy and reliability," *J. For.*, vol. 103, no. 4, pp. 169–173, 2005, doi: 10.1093/jof/103.4.169.

[5] R. Joseph and S. B. Sasi, "Indoor Positioning Using WiFi Fingerprint," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 2018, pp. 1–3. doi: 10.1109/ICCSDET.2018.8821184.

[6] A. H. Lashkari, B. Parhizkar, and M. N. A. Ngan, "WIFI-Based Indoor Positioning System," in *2010 Second International Conference on Computer and Network Technology*, Apr. 2010, pp. 76–78. doi: 10.1109/ICCNT.2010.33.

[7] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, *A Review of Indoor Localization Techniques and Wireless Technologies*, vol. 119, no. 1. Springer US, 2021. doi: 10.1007/s11277-021-08209-5.

[8] Jamaluddin, A. Tjahjo Nugroho, and W. Maulina, "Rancang Bangun Indoor Positioning System berbasis Wireless Smartphone menggunakan Teknik Global Positioning System dengan Metode Absolut (The Design of An Indoor Positioning System Prototype Using Wireless Smartphone by Modify Absolute Method of The Globa," vol. 7, no. 1, pp. 13–18, 2019.

[9] S. Subedi and J. Y. Pyun, "A survey of smartphone-based indoor positioning system using RF-based wireless technologies," *Sensors (Switzerland)*, vol. 20, no. 24, pp. 1–32, 2020, doi: 10.3390/s20247230.

[10] B. Sugiantoro and M. P. Fawzan, "Rekomendasi Access Point Network pada Fakultas di Lingkungan UIN Sunan Kalijaga Yogyakarta," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 2, no. 2, pp. 81–92, 2017, doi: 10.14421/jiska.2017.22-03.

[11] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, "A comparative study on machine learning algorithms for indoor positioning," in *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2015, pp. 1–8. doi: 10.1109/INISTA.2015.7276725.

[12] A. A. Kristianto, X. B. N. Najoan, and ..., "User Locator System Berbasis BSSID dan Alamat MAC Dalam Lingkungan Jaringan WIFI," *J. Tek. …*, vol. 12, no. 1, 2017.

[13] A. Pérez-Navarro, R. Montoliu, E. Sansano-Sansano, M. Martínez-Garcia, R. Femenía, and J. Torres-Sospedra, "Accuracy of a Single Position Estimate for kNN-Based Fingerprinting Indoor Positioning Applying Error Propagation Theory," *IEEE Sens. J.*, vol. 23, no. 16, pp. 18765–18775, 2023, doi: 10.1109/JSEN.2023.3287856.

[14] D. P. Yudha, B. I. Hasbi, and R. H. Sukarna, "Indoor Positioning System Berdasarkan Fingerprinting Received Signal Strength (RSS) Wifi Dengan Algoritma K-Nearest Neighbor (K-Nn)," *Ilk. J. Ilm.*, vol. 10, no. 3, pp. 274–283, 2018, doi: 10.33096/ilkom.v10i3.364.274-283.

[15] A. P. Permana, K. Ainiyah, and K. F. H. Holle, "Analisis Perbandingan Algoritma Decision Tree, kNN, dan Naive Bayes untuk Prediksi Kesuksesan Start-up," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 6, no. 3, pp. 178–188, 2021, doi: 10.14421/jiska.2021.6.3.178-188.

[16] K. Akromunnisa and R. Hidayat, "Klasifikasi Dokumen Tugas Akhir (Skripsi) Menggunakan K-Nearest Neighbor," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 4, no. 1, p. 69, 2019, doi: 10.14421/jiska.2019.41-07.

[17] V. Dang, M. Bendersky, and W. B. Croft, "Two-stage learning to rank for information retrieval," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7814 LNCS, pp. 423–434, 2013, doi: 10.1007/978-3-642-36973-5_36.

[18] S. Sugriyono and M. U. Siregar, "Preprocessing kNN algorithm classification using K-means and distance matrix with students' academic performance dataset," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 4, 2020, doi: 10.14710/jtsiskom.2020.13874.

[19] H. Jiawei, K. Micheline, and P. Jian, *Data mining: Data mining concepts and techniques*, 3rd ed. [1] H. Jiawei, K. Micheline, and P. Jian, Data mining: Data mining concepts and techniques. 2011.: Morgan Kaufmann is an imprint of Elsevier, 2011. doi: 10.1109/ICMIRA.2013.45.

[20] Y. Yohannes, M. R. Pribadi, and L. Chandra, "Klasifikasi Jenis Buah dan Sayuran Menggunakan SVM Dengan Fitur Saliency-HOG dan Color Moments," *Elkha*, vol. 12, no. 2, p. 125, 2020, doi: 10.26418/elkha.v12i2.42160.

[21] M. R. Maarif, "Perbandingan Naïve Bayes Classifier dan Support Vector Machine untuk Klasifikasi Judul Artikel," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 1, no. 2, pp. 90–93, 2016, doi: 10.14421/jiska.2016.12-05.

[22] C. Wei Hsu, C. Chung Chang, and C. Jen Lin, "A Practical Guide to Support Vector Classification," *Dep. Comput. Sci.*, 2016, doi: 10.1094/PHYTO-05-20-0185-R.

[23] N. K. Dewi, S. Y. Mulyadi, and U. D. Syafitri, "Penerapan Metode Random Forest Dalam Driver Analysis," *Forum Stat. Dan Komputasi*, vol. 16, no. 1, pp. 35–43, 2012.

[24] R. A. Haristu, "Penerapan Metode Random Forest untuk Prediksi Win Ratio Pemain Player Unknown Battleground," *MEANS (Media Inf. Anal. dan Sist.*, vol. 4, no. 2, pp. 120–128, 2019, doi: 10.54367/means.v4i2.545.

[25] N. I. Wibowo, T. A. Maulana, H. Muhammad, and N. A. Rakhmawati, "Perbandingan Algoritma Klasifikasi Sentimen Twitter Terhadap Insiden Kebocoran Data Tokopedia," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 6, no. 2, pp. 120–129, 2021, doi: 10.14421/jiska.2021.6.2.120-129.

[26] L. Breiman, "RANDOM FORESTS," *Stat. Dep. Univ. Calif.*, 2001, doi: 10.14569/ijacsa.2016.070603.

[27] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, 2016, doi: https://doi.org/10.1016/j.isprsjprs.2016.01.011.

[28] I. Sutoyo, "IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI DATA PESERTA DIDIK," *J. PILAR Nusa Mandiri*, vol. 14, p. 217, 2018.

[29] T. Tukino, "Penerapan Algoritma C4.5 Untuk Memprediksi Keuntungan Pada PT SMOE Indonesia," *J. Sist. Inf. Bisnis*, vol. 9, no. 1, p. 39, 2019, doi: 10.21456/vol9iss1pp39-46.

[30] S. L. Salzberg, "Book Review: C4.5: by J. Ross Quinlan. Inc., 1993.," *salzberg@cs.jhu.edu*, vol. 16, pp. 235–240, 1994.

[31] S. Ruggieri, "Efficient C4.5 [classification algorithm]," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 2, pp. 438–444, 2002, doi: 10.1109/69.991727.

[32] W. Ian H, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems, 2011. doi: https://doi.org/10.1016/C2009-0-19715-5.

[33] S. Kotsiantis, "Mössbauer study of Fe-Re alloys prepared by mechanical alloying," *Dep. Comput. Sci. Technol. Univ. Peloponnese, Greece*, vol. 31, no. 1, pp. 249–268, 2007, doi:

10.1007/s10751-016-1232-6.