Jurnal Komputer dan Informatika
Vol 20 No 1, April 2025: hlm 71 - 78

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

KLASIFIKASI KELAYAKAN KONSUMSI JAMUR MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS, DECISION TREE, SUPPORT VECTOR MACHINE DAN GRADIENT BOOSTING

Valentino Richardo Lim
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
Email: Valentino.535220186@stu.untar.ac.id

ABSTRAK

Jamur merupakan salah satu organisme yang penting atas keberlangsungan hidup manusia. Dengan adanya berbagai jenis jamur, dapat dibagi menjadi 2 yaitu jamur layak makan dan jamur beracun. Maka dengan adanya teknologi diharapkan dapat melakukan klasifikasi jamur yang layak makan dan jamur yang beracun. Metode yang digunakan merupakan eksperimental klasifikasi dari 4 algoritma klasifikasi yaitu *K-Neareset Neighbors* (KNN), *Decision Tree, Support Vector Machine* (SVM), dan juga *Gradient Boosting Classifier* (GBC). Dengan dataset yang berbentuk 9 kolom serta 54000 sampel didapatkan hasil memuaskan dengan algoritma GBC sebesar 88%. Dapat disimpulkan bahwa algoritma GBC merupakan yang terbaik.

Kata kunci: *K-nearest Neigbors, Decision Tree, Support Vector Machine, Gradient Boosting Classifier*

ABSTRACT

Mushrooms are one of the organisms that are important for human survival. With various types of mushrooms, they can be divided into 2 categories: edible mushrooms and poisonous mushrooms. Therefore, with the help of technology, it is hoped that the classification of edible and poisonous mushrooms can be done. The method used is an experimental classification of 4 classification algorithms: K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), and Gradient Boosting Classifier (GBC). With a dataset consisting of 9 columns and 54000 samples, satisfactory results were obtained with the GBC algorithm at 88%. It can be concluded that the GBC algorithm is the best.

Keywords— *K-nearest Neigbors, Decision Tree, Support Vector Machine, Gradient Boosting Classifier*

1. PENDAHULUAN

Jamur merupakan salah satu organisme di dunia yang memiliki keberagaman paling luas dan sejak jaman dahulu memegang peran yang penting dikeberlangsungan hidup manusia [1]. Peran jamur dalam ekosistem bumi dapat disimpulkan menjadi tiga bagian: (a) menjadi sebuah elemen penting dalam pencapaian dan penyebaran tumbuh-tumbuhan; (b) sebagai sebuah pengurai; (c) menjadi peran symbiosis antara jamur dan inangnya [2]. Sebanyak 200 spesies jamur sudah menjadi makanan yang fungsional di dunia, namun hanya 35 spesies yang dijadikan komersil. Mereka merupakan sumber nutrisi yang kaya terutama dengan kandungan protein

sebesar 20-35% dari berat kering dan juga mengandung 9 asam amino [3]. Namun, ada berbagai jenis jamur yang terbukti beracun dan berbahaya ketika terkonsumsi yang membuat membedakan antara jamur yang layak konsumsi dan tidak menjadi suatu hal yang rumit [4].

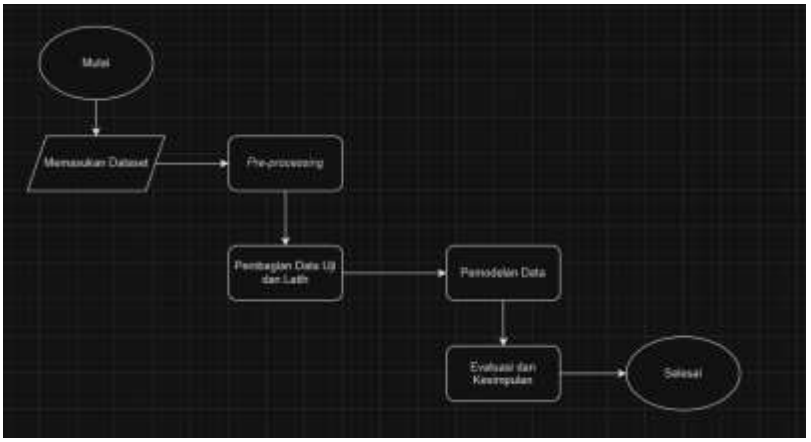
Kasus keracunan akibat jamur bervariasi berdasarkan daerah, musim, dan juga cuaca. Jamur merupakan sebuah organisme *heterotropik eukaryotik* dan memiliki kerajaan tersendiri disamping hewan dan tumbuhan yang membuatnya cukup unik dan sulit diteliti [5]. Tidak ada metode tipikal dalam mengklasifikasi jamur. Pada umumnya mereka dikategorikan layak dan tidak layak dikonsumsi. Namun secara data kita bisa melihat ketegori tersebut dari bentuk fisik [6]. Jamur yang beracun mempresentasikan kurang dari 1% dari keseluruhan dari jamur yang pernah manusia pelajari, oleh karena itu penting identifikasi yang kompeten sebelum proses konsumsi dikarenakan akbiat fatal yang ditimbulkan. Namun dengan dataset dari jamur yang layak makan, dapat dibuat sebuah gambaran mengenai ciri fisik dari jamur tersebut [7]. Klasifikasi dari jamur yang beracun dan layak makan menjadi sebuah hal yang penting. Dengan kemajuan teknologi modern, pengklasifikasian dengan cepat dan menggunakan dataset yang besar pun dapat terjadi.

Penggunaan *machine learning* (ML) menjadi sebuah ketertarikan yang bertumbuh besar dalam dekade terakhir, ketertarikan ini dipercepat oleh komputasi yang murah serta kebutuhan memori yang murah. Oleh karena itu, data yang sangat besar dapat disimpan, diproses, dan dianalisa dengan efisien [8]. Dengan adanya teknologi ini maka dapat dilakukan klasifikasi sederhana untuk membantu menentukan kelayakan konsumsi sebuah jenis jamur.

Penelitian ini memiliki tujuan agar dapat mengetahui klasifikasi kelayakan konsumsi jamur dengan memanfaatkan 4 jenis algoritma yaitu *K-Nearest Neighbors* (KNN), *Decision Tree*, *Support Vector Machine* (SVM), dan *Gradient Boosting* serta membandingkan hasil akurasi dari keempat algoritma tersebut.

2. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental dengan menggunakan 4 jenis algoritma *supervised learning* yang berbeda. Algoritma klasifikasi digunakan untuk dapat membagi dan memprediksi berdasarkan dataset yang tersedia. Pemrosesan data ini memiliki alur penelitian sebagaimana diperlihatkan pada Gambar (1).



Gambar 1 Diagram Alur Penelitian

2.1 Penggunaan Dataset

Dataset yang digunakan didapatkan oleh penulis dari sumber terbuka Kaggle. Dataset yang berjudul *Mushroom Dataset (Binary Classification)* yang diterbitkan oleh Prisha Sawhney

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

berisikan 9 kolom dan 54000 sampel dengan sebuah kolom kelas yang mengklasifikasikan kelayakan konsumsi. Pada kolom 'class' penerbit mengkodekan sebagai berikut: 1 merepresentasikan 'layak konsumsi' dan 0 mempresentasikan 'tidak layak konsumsi/beracun' ditampilkan pada Tabel (1).

Tabel 1 Sampel Dataset Kelayakan Makan Jamur

Cap Diameter	Cap Shape	Gill Attachment	Gill Color	Stem Height	Stem Width	Stem Color	Season	Class
1372	0	2	10	3.80	1545	11	1.80	1
1461	2	2	10	3.80	1557	11	1.80	1
1371	2	2	10	3.62	1566	11	0.94	1
11261	6	2	10	3.78	1566	11	0.93	0

Keterangan:
1.Cap Diamater : Diameter tutup/topi jamur, diukur dalam satuan milimeter (mm)
2.Cap Shape : Bentuk tutup/topi jamur, bel=0, kerucut=1, konveks=2, datar=3, bola=4
3.Gill Atachment : Bentuk sirip Sambungan, bebas=1, berpori=2
4.Gill Color : Warna sirip, sesuai dengan urutan warna
5.Stem Height : Tinggi stem, diukur dalam satuan sentimeter (cm)
6.Stem Width : Lebar stem, diukur dalam satuan mm
7.Stem Color : Warna stem, sesuai Dengan urutan warna
8.Season : Musim ditemukannya jamur, sesuai urutan tanggal musim 0-4
9.Class : Kelayakan konsumsi, layak=1, tidak layak=0

2.2 *Pre-Processing Dataset*

Dataset yang digunakan sebelum dilakukan pemrosesan harus diperiksa dan divalidasi nilainya karena bisa saja mengandung kesalahan ataupun data yang kosong. Beberapa objek tersebut mungkin muncul didalam variasi data yang didapatkan seperti data kosong, gangguan data, perbedaan variabel data yang intens, dll [9]. *Pre-processing* merupakan tahapan paling penting dan berpengaruh dalam algoritma *machine learning* dan memakan sebanyak 50% - 80% dari keseluruhan eksperimen. Tahapan ini menghasilkan data yang baik dan bersih dari berbagai tendensi dan juga gangguan agar dapat digunakan dan terpercaya [10].

2.3 *Pembagian Data Latih dan Uji*

Dataset yang sudah bersih akan dipisah menjadi data latih dan data uji agar dapat dimasukan kedalam model klasifikasi. Dengan permasalahan klasifikasi biner dapat ditemukan ketidak seimbangan yang diakibatkan mayoritas dan minoritas dataset. Hal tersebut dapat diselesaikan dengan berbagai algoritma yang disesuaikan dengan data latih [11]. Pada penelitian ini menggunakan *library* sklearn yaitu model selection yang secara otomatis membagi fitur dan target kelas dengan bentuk tes sesuai yang diharapkan, pada penelitian ini bernilai 0.2 yang berarti 80% data latih dan 20% data uji.

2.4 Pemodelan Data

Dataset yang sudah dibagi menjadi data latih dan data uji akan dilakukan pemodelan menggunakan berbagai *classifier* yang sudah ditentukan. Pada penelitian ini algoritma klasifikasi tersebut yaitu KNN, SVM, *Decision Tree*, dan juga GBC.

2.4.1 *K-Nearest Neighbor*

K-Nearest Neighbor atau yang biasa disingkat KNN merupakan algoritma pengklasifikasian yang dapat melakukan improvisasi dalam pembelajaran dengan menggunakan nilai *K* dan mencari tetangga terdekat dalam proses prediksi data [12]. Algoritma pengklasifikasi menggunakan kedekatan jarak tetangga yang dihitung dengan rumus *Euclidean Distance* yang dipetakan dalam tempat dimensional dan diberikan berat pada setiap data untuk menghasilkan perhitungan [13].

Jarak antara tetangga yang dipetakan dihitung menggunakan rumus *Euclidean Distance* pada persamaan (1), yang dimana *d(x,y)* merupakan jarak antar dua titik, *x* dan *y* merupakan data latih dan data uji secara berurutan, *m dan i* merupakan dimensi data variabel data secara berurutan.

$$d(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \quad (1)$$

2.4.2 *Support Vector Machine*

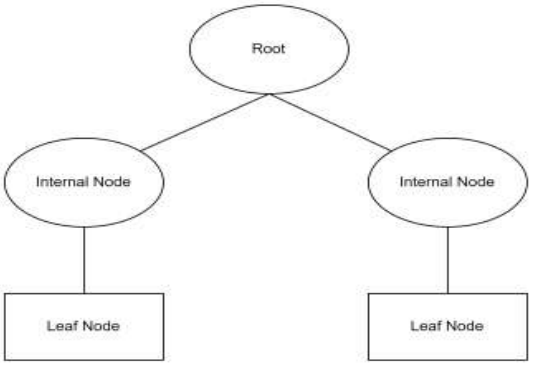
SVM merupakan algoritma yang dapat melakukan klasifikasi serta regresi. SVM dapat melakukan klasifikasi terhadap data linear dan non-linear. Pada data non-linear, digunakan kernel *Radial Basis Function* (RBF) yang memetakan data pada dimensi tinggi dan melakukan perhitungan pada data non-linear [14]. Klasifikasi SVM dengan kernel RBF menggunakan rumus pada persamaan (2), yang dimana *K(x,,x')* merupakan nilai kernel dari dua vektor, *Y* merupakan parameter control dari RBF.

$$K(x,x') = e^{-Y|x-x'|^2} \quad (2)$$

2.4.3 *Decision Tree*

Decision Tree merupakan pengklasifikasi yang membantu mengidentifikasi hubungan antara poin dataset dengan membangun struktur pohon. Cabang-cabang dari pohon tersebut digunakan untuk membagi dataset menjadi beberapa bagian yang sesuai dengan simpul keputusan [15].

Decision Tree dapat dilihat sebagai simpul-simpul pohon yang memiliki *root, internal node,* dan *leaf node.* Dari cabang merupakan awal perhitungan sampai kepada simpul dalam yang memberi perhitungan yang menghasilkan simpul daun berupa hasil keputusan. Sebagaimana ditunjukan pada Gambar (2).



Gambar 2 *Decision Tree*

Perhitungan dari setiap cabang dihitung menggunakan rumus *Gain* dengan hitungan tertinggi. Rumus *Gain* menggunakan nilai *Entropy(S)* yang merupakan nilai dari *Entropy* data set *S*, nilai *n* merupakan jumlah kelas, nilai *pi* merupakan jumlah sampel daari kelas *i* terhadap total sampel dataset *S*, *Values(A)* merupakan kemungkinan nilai pada fitur *A*, *Sv* Merupakan subset data *S* yang memiliki nilai *V= A*, dan ||*S*|| merupakan total sampel pada dataset *S*. Rumus *Entropy* dapat dilihat pada persamaan (3).

$$Gain(S,A) = Entropy(S) - \sum_{v=values(A)} \frac{||S_v||}{||S||} Entropy(S_v) \quad (3)$$

Kemudian dapat dilihan pula rumus *Entropy* pada persamaan (4).

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

$$\text{Entropy}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i \quad (4)$$

2.4.4 *Gradient Boosting*

Klasifikasi ini merupakan sebuah teknik yang bergantung pada perataan model yang digabung dengan strategi membangun ulang model secara sekuensial. Pada setiap iterasi ada sebuah hasil pelatihan yang mempelajari setiap kesalahan pada seluruh kontruksi yang dibuat sejauh ini [16].

Pada klasifikasi ini hanya akan menunjukan rumus inialisasi *Gradient Boosting* dengan dimana variabel $f_0(x)$ merupakan model awal, L merupakan fungsi kerugian yang digunakan, y_i merupakan target variabel dan γ merupakan prediksi awal model. Rumus tersebut akan diberikan pada persamaan (5).

$$f_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \quad (5)$$

2.5 *Evaluasi Model*

Evaluasi model penting dalam melihat pergorma algoritma klasifikasi [17]. Penelitian ini menggunakan *Correlation Matrix* dan *Confusion Matrix* sebagai sarana pengevaluasi model.

2.5.1 *Correlation Matrix*

Matriks ini memberikan representasi korelasi dari berbagai kolom fitur untukk melihat keterkaitan dan berat dari setiap kolom fitur [18].

2.5.2 *Confusionn Matrix*

Matriks ini merupakan tabel yang memberikan hasil dari prediksi model yang mengevaluasi klasifikasi menjadi 4 bagian yaitu, TP ketika prediksi tepat akan hasil positif, TN ketika prediksi tepat akan hasil negative, FP ketika prediksi salah akan hasil positif, dan FN ketika prediksi salah akan hasil negative [19].

Dalam gambar (3) divisualisasika menganai *Confusion Matrix* secara peletakan.

		Kelas Prediksi		
		A	B	C
Kelas Sebenarnya	A	AA	AB	AC
	B	BA	BB	BC
	C	CA	CB	CC

Gambar 3 Confusion Matrix Dengan Banyak Kelas = 3

2.5.3 *Akurasi Model*

Akurasi model dilakukan dengan rumus Akurasi yang mengambil nilai TP, TN, FP, dan FN pada pembahasan *confusion matrix*. Rumus akurasi akan diperlihatkan pada persamaan (6). Dengan variabel-variabel seperti TP ketika prediksi tepat akan hasil positif, TN ketika prediksi tepat akan hasil negative, FP ketika prediksi salah akan hasil positif, dan FN ketika prediksi salah akan hasil negative.

$$Ak\text{}\text{}a\text{}i = \frac{\text{}P + \text{}N}{\text{}P + \text{}N + \text{}P \_ \text{}N} \times 100\% \quad (6)$$

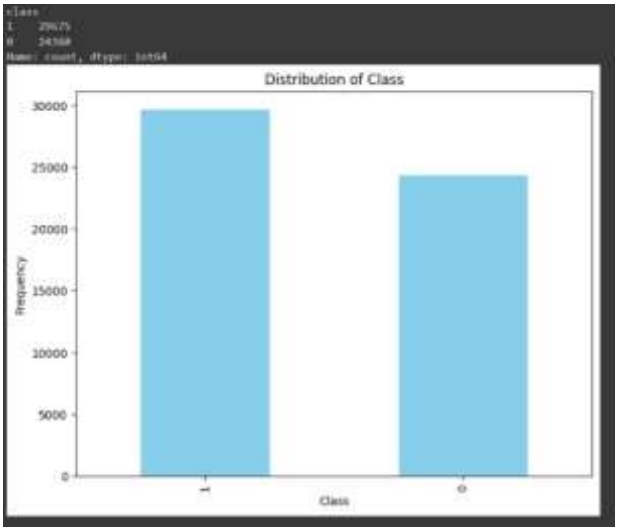
3. HASIL DAN PEMBAHASAN

Berdasarkan metode diatas maka didapatkan hasil penelitian yang akan diuraikan pada bab ini. Dengan menggunakan bahasa pemrograman python maka dihasilkan pemaparan data seperti info dari data sebagaimana dijelaskan pada gambar (4).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54035 entries, 0 to 54034
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   cap-diameter     54035 non-null  int64
 1   cap-shape        54035 non-null  int64
 2   gill-attachment  54035 non-null  int64
 3   gill-color       54035 non-null  int64
 4   stem-height      54035 non-null  float64
 5   stem-width       54035 non-null  int64
 6   stem-color       54035 non-null  int64
 7   season           54035 non-null  float64
 8   class            54035 non-null  int64
dtypes: float64(2), int64(7)
```

Gambar 4 Info Dataset yang Digunakan

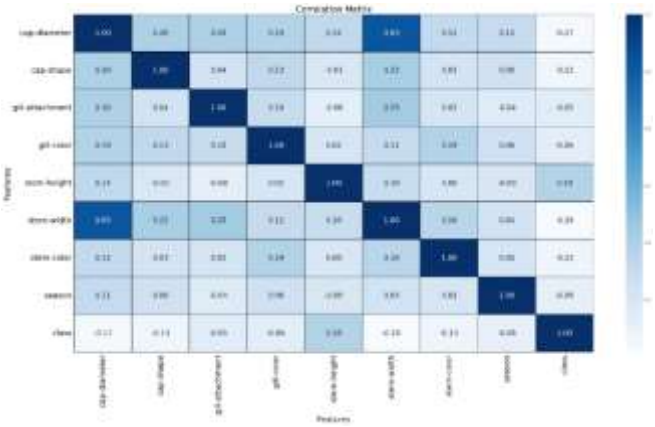
Dapat dilihat pula ketika data uraikan dan dibagi melihat seperti apa distribusi kelas menggunakan plot. Dapat dilihat plot tersebut pada gambar (5)



Gambar 5 Plot Distribusi Class dari Dataset

Jika dilihat bahwa mayoritas data merupakan nilai 1 yang berarti jenis jamur tersebut layak konsumsi. Namun tidak berbeda jauh dibanding bagian lawannya. Setelah kita melihat hal tersebut penting untuk mengetahui bagaimana keterkaitan fitur-fitur terhadap hasil dengan melihat *correlation matrix* dari data tersebut. Matriks korelasi dataset dapat dilihat pada gambar (6) yang dimana semakin biru atau nilai semakin tinggi maka semakin positif korelasi tersebut dan sebaliknya.

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)



Gambar 6 Matriks korelasi dari Dataset

Setelah melakukan hal-hal tersebut maka dilakukanlah eksperimen dengan klasifikasi-klasifikasi yang sudah ditentukan. Eksperimen dilakukan dengan mengganti *hyperparameter* maupun *tuning* dari klasifikasi yang tersedia. Kemudian digunakan pembagian data uji dan latih sebesar 0.2. Akan ditunjukan *confusion matrix* dari hasil terbaik setiap klasifikasi pada gambar (7, 8, 9, 10).

```
Confusion Matrix
array([[3815, 1059],
       [1974, 3959]])
```

Gambar 7 Confusion Matrix KNN dengan N= 4

```
Confusion Matrix
array([[3343, 1531],
       [1207, 4726]])
```

Gambar 8 Confusion Matrix Decision Tree dengan Depth = 5

```
Confusion Matrix
array([[3015, 1859],
       [1325, 4608]])
```

Gambar 9 Confusion Matrix SVM Kernel RBF dengan Nilai C=750

```
Confusion Matrix
array([[4288,  586],
       [ 706, 5227]])
```

Gambar 10 Confusion Matrix Gradient Boosting

Dari hasil-hasil diatas dilakukan perhitungan menggunakan rumus akurasi yang dapat dilihat pada persamaan (6) yang menghasilkan tabel berisi perbandingan keempat *Classifier* yang dapat dilihat pada tabel (2).

Tabel 2 hasil perbandingan akurasi dari keempat *Classifier*

Algoritma	Akurasi
KNN	72%
Decision Tree	75%
SVM	71%
GBC	88%

Berdasarkan evaluasi tersebut kita dapat melihat kinerja masing masing algoritma. Hasil eksperimen melihat kemampuan sebuah model dalam mengerjakan suatu klasifikasi dengan dataset yang diberikan, dengan hasil yang baik berada pada akurasi awal 70%-80% [20]. Jadi dapat dikatakan bahwa kinerja algoritma diatas sudah cukup baik dan GBC mendapatkkan akurasi yang sangat baik. Namun dengan kelebihan dan kekurangan tersendiri dikarenakan GBC yang cukup kompleks dan memakan waktu komputasi yang lebih dibanding algoritma lainnya.

4. KESIMPULAN

Kesimpulan dari penelitian ini adalah bahwa algoritma memiliki kekurangan dan kelebihan serta dapat juga pentingnya dalam mengatur *hyperparameter* dalam setiap dataset yang berbeda untuk mencapai tingkat akurasi yang terbaik. Dengan hasil penelitian ini didapatkan algoritma GBC mendapatkan nilai akurasi tertinggi dengan nilai 88%. Pengembangan kedepannya bisa ditingkatkan dengan eksperimen lebih lanjut dalam *fine-tuning* parameter-parameter yang ada.

DAFTAR PUSTAKA

[1] M.-I. E, G.-M. MB and A.-L. XA, "Climatic and Socioeconomic Aspects of Mushrooms: The Case of Spain," p. 1, 2019.
[2] N. Igiehon and O. Babalola, "Fungal bio-sorption potential of chromium in Norkrans liquid medium byshake flask technique," *J. Basic Microbiol,* pp. 59-73, 2019
[3] M. B. Balletini and F. A. Fiorda, "Factors affecting mushroom Pleurotus spp," *Saudi Journal of Biological Sciences,* vol. 26, no. 4, pp. 633-646, 2019.
[4] J. white, S. A and L. D. Haro, "Mushroom poisoning: A proposed new clinical classification,"*Toxicon,* vol. 157, p. 53, 2019.
[5] W. R, E. F and S. A, "Mushroom Poisoning," *Dtsch Arztebl Int,* vol. 117, p. 701, 2020.
[6] D. G and A. G. Karegowda, "Identification of Edible and Non-Edible Mushroom," *Atlantis Highlights in Computer Sciences,* vol. 4, p. 312, 2021.
[7] S. C, H. M and R. B, "COMPARATIVE STUDY ON EDIBLE AND NON EDIBLE MUSHROOM AGAINST," *Scholar : National School of Leadership,* vol. 8, 2019.
[8] D. A and Omondiagbe, "Machine Learning Classification Techniques for Breast Cancer," *IOP Conf. Series: Materials Science and Engineering 495,* pp. 1-3, 2019 .
[9] P. Mishra and A. Biancolillo, "New data preprocessing trends based on ensemble of multiple Preproccesing techniques," *Trends in Analytical Chemistry,* vol. 132, pp. 1-2, 2020.
[10] K. Maharana and S. Mondal, "A review: Data pre-processing and data augmentation techniques,"*Global Transitions Proceedings,* vol. 3, no. 1, pp. 91-92, 2022
[11] A. Rácz and D. Bajusz, "Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification," vol. 26, no. 4, 2021.
[12] S. Zhang, "KNN Classification With One-Step Computation," *IEEE Transactions on Knowledge and Data Engineering,* vol. 35, pp. 2711-2723, 2020.
[13] S. Klidbary and A. Arabameri, "A Novel Density-Based KNN in Pattern Recognition," *13th International Conference on Computer and Knowledge Engineering,* pp. 185-190, 2023.
[14] B. F and S. E, "Non-linear Multiclass SVM Classification Optimization using Large Datasets of Geometric Motif Image," *International Journal of Advanced Computer Science and Applications,* 2021.
[15] T. Thomas and A. Vijayaraghavan, "Applications of Decision Trees," *Machine Learning Approaches in Cyber Security Analytics,* pp. 157-184, 2019.
[16] M. Denuit and D. Hainaut, "Gradient Boosting with Neural Networks.," *Springer Actuarial,* p. 167, 2019
[17] M. H. K. A. T. F. N. T. a. A. A. T. Alyas, "method for thyroid disease classification using a machine learning approach," *BioMed Research International,* 2022.
[18] J. &. L. J. Graffelman, "Improved Approximation and Visualization of the Correlation Matrix," *The American Statistician,* vol. 77, p. 432, 2022.
[19] M. I. T. S. S. a. Y. A. Fikri, "Perbandingan metode naïve bayes dan support vector machine pada analisis sentimen twitter," *STIKI Informatika Jurnal,* vol. 2, p. 71, 2020.
[20] M. Patwary and X. Wang, "Sensitivity analysis on initial classifier accuracy in fuzziness based semi-supervised learning," *inf Sci,* p. 93, 2019.