International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 130-141 March 2026-Sept 2026 www.csjournals.com

Classification of Mushroom Edibility Using Machine Learning: A Comparative Study of Six Supervised Algorithms

Kajal[1], Dr. Sahil Saharan[2], Dr Anupam Bhatia[3]
[1]MCA Student, [2]Asst. Prof., [3]Associate Professor
Deptt. of Comp. Sc. and Application, Chaudhary Ranbir Singh University, Jind, Haryana, India

Abstract — The misidentification of edible and poisonous species is a continuing source of mushroom poisoning, and the manual mycological identification is prone to error and the need for expertise. The paper compares and tests six supervised machine learning (ML) algorithms: K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), XGBoost, Logistic Regression (LR), and Support Vector Machine (SVM) with the Secondary Mushroom Dataset, comprising 61,069 mushrooms with 16 morphological features after preprocessing, for the binary classification of edibility. All the models were preprocessed with a systematic pipeline that includes removal of high-missing-value features, imputation of modal features, application of IQR to cap outlier values, encoding of labels, ranking features by importance using the random forest, and splitting the data into an 80/20 train-test split. Performance was assessed with the following seven metrics: training accuracy, test accuracy, difference between training and test accuracy, precision, recall, F1 score, ROC-AUC, 5-fold cross validation score. KNN (k = 60, uniform weights) achieved the best overall performance — 97.83% test accuracy, 98.71% precision, 97.40% recall, 98.05% F1-score and ROC-AUC of 0.9983 — followed by Decision Tree (88.37%), Random Forest (77.55%), XGBoost (66.53%), Logistic Regression (64.34%) and SVM (64.30%). The most discriminative predictors are stem_width (0.134), cap_surface (0.087) and gill_attachment (0.085) which are in agreement with the mycological field-identification knowledge. The study places these results in the context of previous work on the classic UCI Mushroom Dataset, highlights important research gaps in the field of ‘scale of the mushroom dataset', ‘outlier treatment' and ‘multi-metric evaluation', and suggests that using well-preprocessed tabular morphological data, followed by an appropriate tuning of KNN, is a very efficient and low-cost approach to achieve high accuracy in mushroom-edibility prediction with no need of computationally expensive deep-learning pipelines.

Index Terms — Mushroom classification, machine learning, K-Nearest Neighbors, feature importance, food safety, supervised learning, comparative analysis.

I. INTRODUCTION

Mushrooms are one of the most nutritionally diverse and economically valuable organisms in the world and have been a part of the human diet and culture in Asia, Europe and the Americas for many years, due to their taste and rich protein content and medicinal value [1]. In recent decades, the edible-mushroom market has been growing significantly globally, amidst the increasing demand for plant-based nutrition. The fungal kingdom, however, is estimated to have between 2.2 and 3.8 million species, and only a fraction has been formally described, a large percentage of

the approximately 10,000 species that produce macroscopic fruiting bodies can cause hepatotoxic, nephrotoxic or neurotoxic effects. The genus Amanita, especially Amanita phalloides (the “death cap”), is responsible for most mushroom fatalities in the world; its amatoxins are thermostable and remain active after cooking [3]. The biggest challenge to mushroom safety is the fact that edible and poisonous varieties are so similar in appearance. The ability to distinguish such species is highly dependent on careful examination of the morphological characters; although these characters can be used for the distinction, the knowledge of trained mycologists is not generally accessible to the general public of foragers and causes annual hundreds of fatalities and thousands of non-fatal poisonings, particularly during the monsoon and autumn seasons of foraging [4]. Such problems have been successfully solved with machine learning, which can learn complex non-linear decision boundaries from the data, and with much greater capacity than rule-based or purely expert-driven systems [5] are able to. This is what makes the present comparative study relevant. Although there are large mushroom morphological databases, there is no widely used and automated method for real-time mushroom classification as to edibility using morphological traits that can be observed by non-experts. However, the majority of previous works are based on the classical UCI mushroom dataset (8,124 instances), despite its low number of instances and limited feature space, which are insufficient to allow generalisation of the model. In this paper, the binary classification problem of whether the specimen is edible or poisonous is addressed with binary classification for a Secondary Mushroom Dataset (61,069 instances) for the morphological attributes with larger and more realistic data. The specific objectives are: (i) to create a systematic pipeline to preprocess the data; (ii) to implement and test six different classifiers (all supervised) under same conditions; (iii) to determine the most discriminative morphological predictors by means of Random Forest feature importance; (iv) to interpret the results in terms of food safety and mycological relevance.

II. RELATED WORK

In UCI Mushroom Dataset, Admojo et al. [6] obtained 99% accuracy with nine features after pre-processing the data set with mode imputation, one-hot encoding, z-score and 10,807 test instances, pointing the need of scalability of the distance computation at prediction time as the most critical limitation. Arslan et al. [7] compared five ensemble techniques (Random Forest, Gradient Boosting, AdaBoost, Extra Trees and Bagging) on the same UCI dataset using the MCC and Cohen's Kappa, along with standard metrics; Extra Trees was the best (accuracy 99.17%, ROC-AUC 0.9994) and a strong positive correlation (0.83) between cap_diameter and stem_width was reported, which justifies the correlation analysis used here.

Thakur and Thakur [8] introduced an image-based five-phase framework that involved classification of the mushrooms using SVM, neural networks, Decision Tree and KNN on the basis of original images and images with background removed; KNN with real physical dimensions (cap radius, stem height, stem girth) achieved 95%, a significant improvement over the 82% achieved by KNN with virtual physical dimensions (approximated from the images). Thus, the importance of real physical dimensions over virtual physical dimensions (which are approximated from images) was emphasized. The near-perfect cross-validated accuracy rates achieved by CNN, Logistic Regression, Naïve Bayes, Decision Tree, and Random Forest in the UCI dataset (up to 100% in the case of Random Forest) demonstrate that the classic dataset is very well-separated.

Devika and Karegowda [10] performed a comparative study of deep architectures (DCNN, LeNet, AlexNet, cNet, sNet) with the traditional ML classifiers on 15,400 mushroom images and achieved 97.03% accuracy, with DCNN outperforming SVM, Naïve Bayes, Decision Tree, Random Forest and most significantly KNN (14%) on raw pixel features, whereas KNN had high accuracy on tabular morphological features, thus highlighting the importance of feature representation. Tutuncu et al. [11] tested Naïve Bayes, Decision Tree, SVM and AdaBoost on the 22-feature UCI dataset, with 10-fold cross-validation, and reported 100% classification accuracy for the latter, recommending the deployment of mobile applications as an operational extension.

TABLE I
Summary of Related Work on Mushroom Edibility Classification

Ref.	Authors	Algorithm(s)	Dataset	Best Acc.	Contribution
[6]	Admojo et al.	KNN	UCI (9 feat.)	99%	Z-norm + KNN; scalability limit
[7]	Arslan et al.	Extra Trees, RF, AB	UCI (8,124)	99.17%	Ensemble comparison; MCC/Kappa
[8]	Thakur & Thakur	KNN, SVM, NN, DT	Image data	95%	Real dims > image features
[9]	Manikanteswari& Kalyani	CNN, LR, DT, RF	UCI (8,124)	100% (RF)	CNN + ML; cross-validation
[10]	Devika &Karegowda	DCNN, SVM, KNN	15,400 images	97.03%	DCNN > ML on raw images
[11]	Tutuncu et al.	NB, DT, SVM, AB	UCI (8,124)	100% (AB)	AdaBoost perfect; mobile use

Research Gap

The dataset of mushrooms has been largely dominated by the relatively small size (8,124 instances) and exclusively categorical UCI Mushroom Dataset, and most of the comparisons are conducted at 4 or 5 classifiers at a time, rarely at a multi-metric evaluation with generalization awareness, and often using only accuracy and AUC metrics. This study aims to fill these gaps by applying six different heterogeneous algorithms (including lazy learners, tree-based methods, gradient boosting, linear models and kernel methods), running them all under exactly the same

preprocessing and evaluation conditions with explicit outlier capping using the IQR and reporting seven complementary performance measures (including the train-test gap and the 5-fold cross validation).

III. MATERIALS AND METHODS

A. Dataset Description

The Secondary Mushroom Dataset, which was downloaded from Kaggle (contributed by Prisha Sawhney), has 61,069 rows with 20 original features. A total of 16 features were used after excluding the features that had greater than 80% missing values, such as stem_root, stem_surface, veil_type, veil_color, and spore_print_color. Data is moderately imbalanced with the edible class (e: 26,888 instances, 44%) and the poisonous class (p: 34,181 instances, 56%) indicated in the binary target class. The data underlying it has been artificially generated from feature distributions obtained from the FoodOn ontology and actual mushroom characteristics of mushroom species, ensuring sufficient size and biological plausibility for conducting ML experiments.

B. Data Preprocessing

Features were dropped if missing values occurred in more than 80% of the data, categorical features were then imputed from the remaining features using the mode of the data, the remaining features that were continuous were capped at Q1 − 1.5×IQR and Q3 + 1.5×IQR, categorical features and the target features were then encoded using Scikit-learn's LabelEncoder with e = 0, p = 1 and the splits were performed as an 80/20 train-test split which maintained the class balance of 56:44 in both sets.

C. Feature Selection

The most important features were stem_width (0.134), cap_surface (0.087), gill_attachment (0.085), gill_color (0.083) and stem_color (0.081); season was the least (0.009). The Pearson correlation matrix (Fig. 2) showed a positive moderate correlation ($r = 0.82$) between cap_diameter and stem_width, which is consistent with the growth of a tree, and a moderate negative correlation ($r = -0.32$) between gill_spacing and gill_color. The majority of features had low correlation with the class label (max $|r| = 0.20$ for stem_width), which suggests that these features had non-linear class boundaries, further justified by using flexible, non-linear classifiers.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 130-141 March 2026-Sept 2026 www.csjournals.com

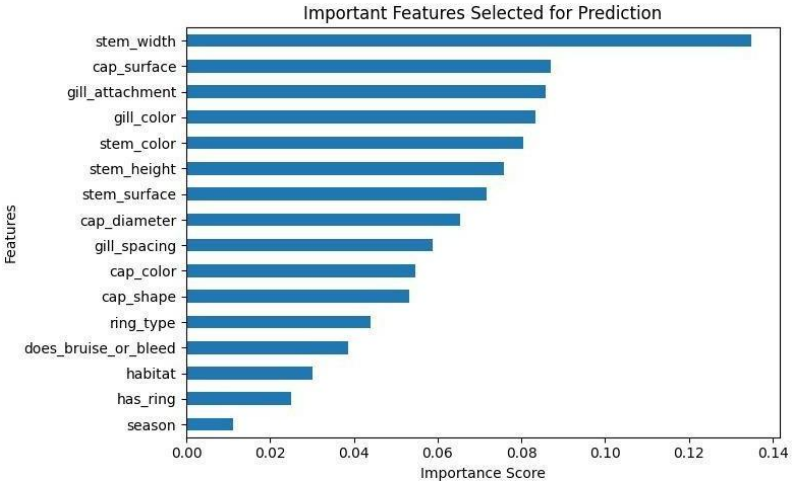

Fig. 1. Random Forest feature importance scores for all 16 retained features.

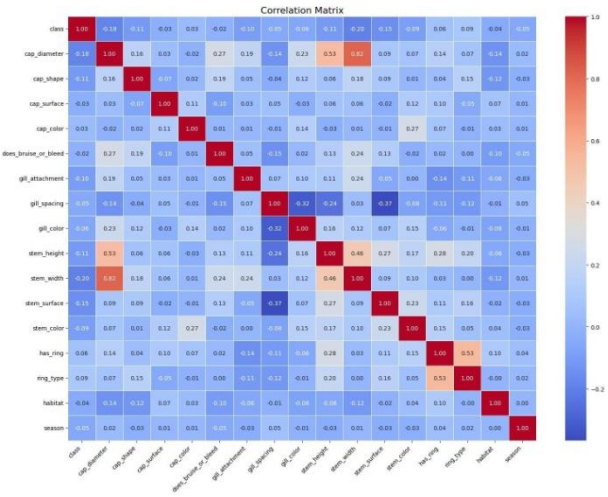

Fig. 2. Pairwise Pearson correlation matrix of all dataset features.

D. Machine Learning Algorithms

K-Nearest Neighbors (KNN) is an instance-based non-parametric classification method which classifies an instance based on the highest proportion of its k nearest training instances in feature space according to some distance function without any model construction during training. Classifier was set up with k = 60 and uniform weights. Decision Tree (DT) repeatedly divides the feature space in order to achieve minimum Gini impurity at every split; maximum depth, max_depth = 8 and minimum leaf size, min_samples_at_leaf = 10, to limit overfitting. Random Forest (RF) constructs majority voting ensembles of decision trees on subsets of a bootstrap sample with random feature selection at each split; the 50 decision trees had a maximum depth of 5 nodes, a minimum split of 20, and used the square-root of the number of features as the minimum number of features for a split.

The trees are added one by one and correct the residual errors of the previous trees using gradient descent, with L1/L2 regularization to prevent overfitting, a deliberately conservative setting (30 trees, max_depth = 2, learning rate = 0.03) was chosen to test the generalizability. The model

Logistic Regression (LR) fits the log-odds of class membership in a linear combination of features through the sigmoid function and uses the liblinear solver with C = 10. Support Vector Machine (SVM) (LinearSVC, C = 1.0) is a hyperplane which is best at separating the classes while maximizing the margin between the two classes; a linear kernel was used for the 61,069-instance dataset due to computational feasibility..

E. Tools and Technologies

All experiments were conducted in Python 3, within a Jupyter Notebook environment where scikit-learn was used for model implementation, preprocessing and model evaluation; XGBoost was used for gradient boost, Pandas and NumPy was used for data handling; Matplotlib/Seaborn was used for visualization; And the pipeline was fully reproducible.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The proposed system is a four stage pipeline architecture, as shown in Fig. 3, which includes Data Layer for raw CSV ingestion and inspection, Preprocessing Layer for feature removal, imputation, outlier capping, encoding and feature-importance ranking, Modelling Layer for training six classifiers on identical data splits, Evaluation Layer for multi-metric evaluation and comparative analysis. The process flow of the corresponding steps is depicted in Fig. 4. Each classifier was fitted using the 48,855-instance training set and the resulting predictions were then used to predict both training and test sets, with the training and test accuracy compared to give the train-test gap, followed by precision, recall, F1-score and ROC-AUC calculations on the test set predictions, and a 5-fold stratified cross-validation score was computed on the other generalization set, which was split independently. Probabilistic classifiers: predict_proba() was used to compute ROC-AUC.LinearSVC: decision_function() was used to compute ROC-AUC.

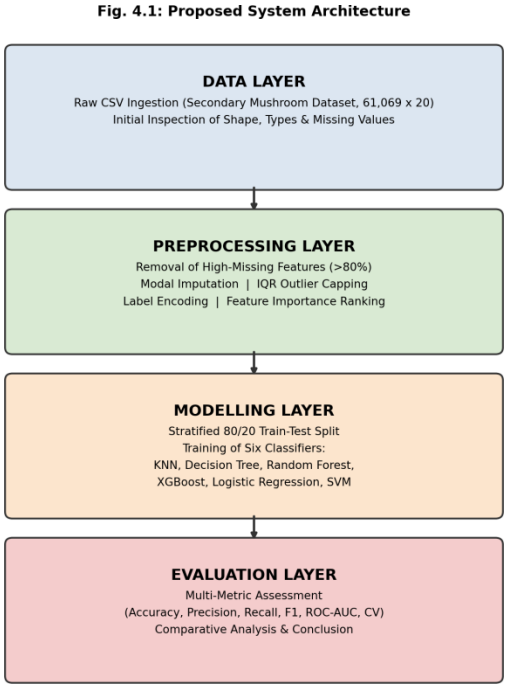

Fig. 3. Proposed four-layer system architecture.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 130-141 March 2026-Sept 2026 www.csjournals.com

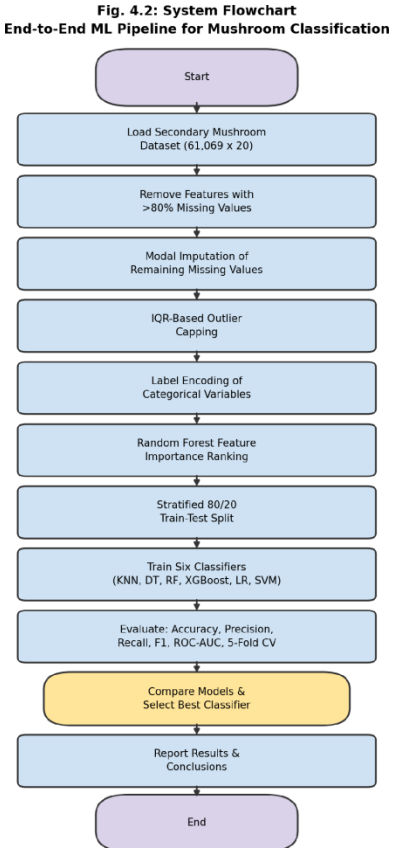

Fig. 4. End-to-end system flowchart for mushroom classification.

V. RESULTS AND DISCUSSION

A. Exploratory Data Analysis

The class distribution is displayed in Fig. 5 which had a slight imbalance of 34,181 poisonous instances and 26,888 edible instances. The distribution of stem_width by class is illustrated with box plot; stem_width is a highly discriminative feature, with a very low median size for poisonous mushrooms which is ≈ 8 mm, compared to the median size for edible mushrooms, which is ≈ 12 mm, but the interquartile ranges overlap so there is no single-feature separation and encourage the use of multi-feature ML approaches. The distribution of gill_color by class is displayed in Fig. 7, and there is a noticeable excess of white gills in both classes compared to yellow gills, which are more common among poisonous specimens. Habitat distribution (Fig. 8): both plants are found mainly in deciduous woodland, but poisonous plants are over-represented in the grasses habitat compared with edible plants.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 130-141 March 2026-Sept 2026 www.csjournals.com

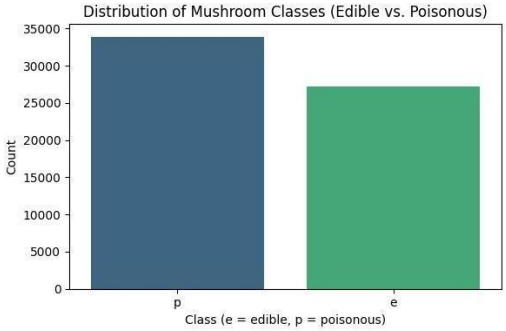

Fig. 5. Distribution of mushroom classes — poisonous (p) vs. edible (e).

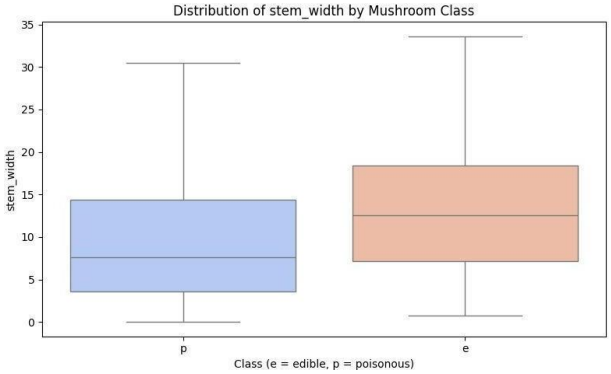

Fig. 6. Distribution of stem_width by mushroom class (box plot).

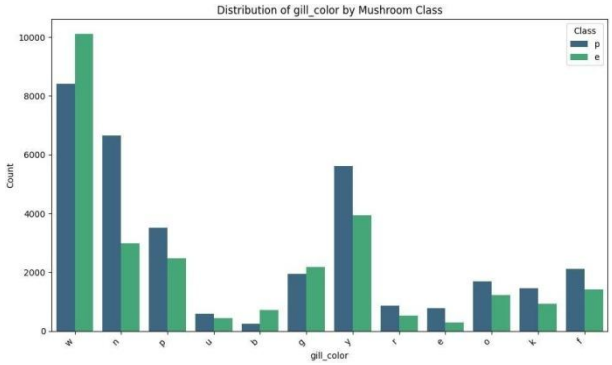

Fig. 7. Distribution of gill_color by mushroom class.

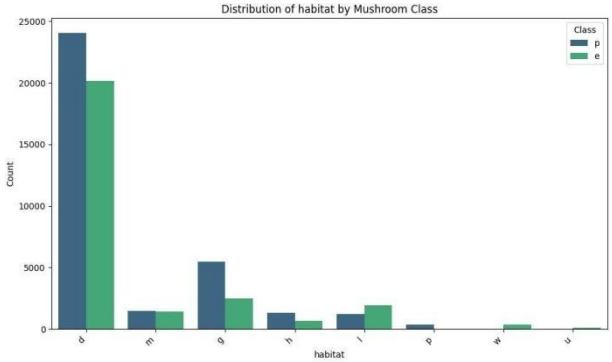

Fig. 8. Distribution of habitat by mushroom class.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 130-141 March 2026-Sept 2026 www.csjournals.com

B. Comprehensive Model Performance

All six classifiers are listed in Table II and Fig. 9, ordered by test accuracy. KNN performed best in all the metrics taken.

TABLE II
Comprehensive Model Performance Comparison

Model	Train Acc.	Test Acc.	Gap	Precision	Recall	F1	ROC-AUC	5-Fold CV
KNN	0.9791	0.9783	0.0008	0.9871	0.9740	0.9805	0.9983	0.9776
Decision Tree	0.8900	0.8837	0.0067	0.9415	0.8444	0.8905	0.9454	0.8829
Random Forest	0.7847	0.7755	0.0092	0.8346	0.7472	0.7884	0.8944	0.7980
XGBoost	0.6642	0.6653	-0.0011	0.6699	0.7931	0.7263	0.7457	0.6599
Logistic Reg.	0.6432	0.6434	-0.0002	0.6817	0.7431	0.7100	0.6968	0.6431
SVM	0.6421	0.6430	-0.0009	0.6816	0.7437	0.7100	0.6973	0.6418

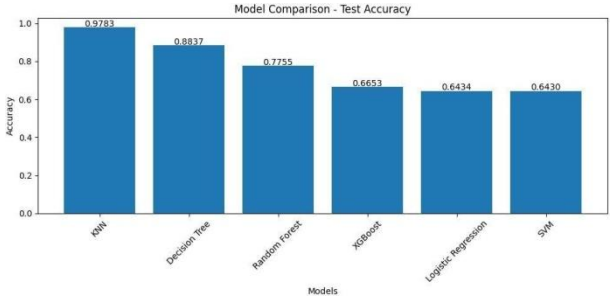

Fig. 9. Model comparison — test accuracy (sorted descending).

KNN (97.83%) significantly outperforms every other classifiers with difference of almost 9.46 percentage points from the second best model Decision Tree (88.37%) which shows that the proximity-based reasoning of KNN is more suitable for cluster structure of the dataset. A key metric in this space is the test recall, which is shown in Fig. 10. A false negative (poisonous identified as edible) has serious health impacts. Again KNN is the top performer with a 97.40% recall followed closely by XGBoost (79.31%) and Random Forest, Logistic Regression and SVM are close together and perform at around 74–75%.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 130-141 March 2026-Sept 2026 www.csjournals.com

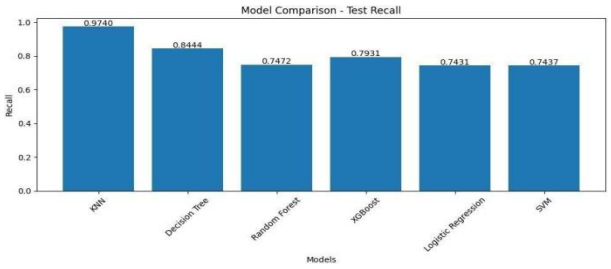

Fig. 10. Model comparison — test recall.

The ROC curve for each of the 6 classifiers are shown in Fig. 11. The KNN model has a near-perfect ROC-AUC of 0.9983 with the curve climbing towards the upper-left corner, whereas the other models are not that impressive: Decision Tree (0.9454) and Random Forest (0.8944) also perform well, while the discriminative power of Logistic Regression (0.6968), SVM (0.6973) and XGBoost (0.7457) is significantly lower. Fig. 12 shows the confusion matrix for the best performing KNN model: Of the 12,214 test instances, there were 5,287 true negatives, 6,662 true positives, 87 false positives (a conservative, low-risk error) and 178 false negatives — the critical error type for food safety, kept comparatively low.

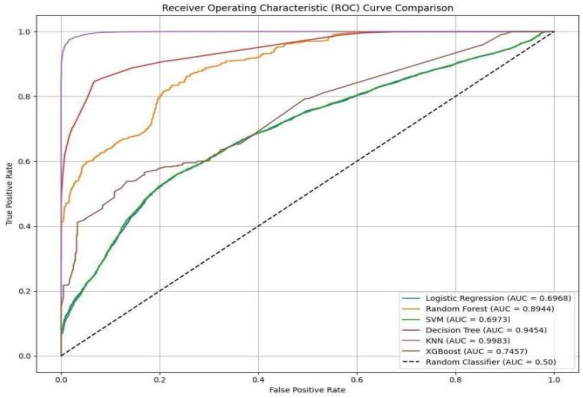

Fig. 11. ROC curve comparison for all six classifiers.

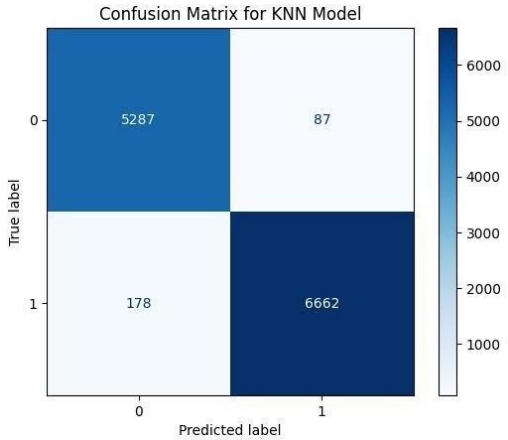

Fig. 12. Confusion matrix for the KNN model (k = 60, test set).

C. Comparative Analysis

There are significant differences in the overall performance of the models in terms of their ranking; the non-linearly separable feature space (data) of the Secondary Mushroom Dataset causes the algorithms to perform at very different levels. Near zero train-test gap scores (0.0008 for KNN and 0.0002 for Logistic Regression) are consistent with mild underfitting, which is desired when the model is deliberately set to be conservative to test generalizability; positive gap scores (0.0067 for Decision Tree and 0.0092 for Random Forest) suggest mild overfitting which can be addressed by increasing the depth of hyperparameter tuning; finally, negative gap scores for XGBoost (−0.0011) and Logistic Regression (−0.0002) suggest near ideal bias–variance balance. The poor performance of Logistic Regression (64.34%) and SVM (64.30%) is in accordance with the correlation analysis: Low pairwise correlation between the individual features and the class variable (max $|r| = 0.20$) indicates that the data is not linearly separable, which means that purely linear classifiers will not be able to capture the complex decision boundaries required.

D. Discussion of Findings

KNN's success is largely due to three factors: the normalized features are given equal weights when computing distance, there is a high instance density in the 61,069 instance dataset, which means that the k-neighbourhood around every query point is well populated; and the underlying feature space of the instances has reasonably well-separated clusters of edible and poisonous instances, and the distance properties are not explicitly assumed or modelled by the algorithm. The second and fourth most important features (gill_attachment and gill_color) are also well-known field-identification features that have been mentioned in mycological literature as being distinguishing characteristics between toxic and edible Amanita species, which is supported by the box-plot analysis (Fig. 6) comparing the distribution of these features for the two species. The KNN model is robust, with a very low miss rate (178 false negatives out of 6,840 actual poisonous test instances, or approximately 2.60%). This could be used as a screening process to reject samples likely to be poisonous first time, whilst reducing the proportion of non-poisonous samples that are rejected.

VI. CONCLUSION AND FUTURE SCOPE

In this study, a comprehensive comparative study and experimental analysis of six binary mushroom-edibility classification algorithms from supervised machine learning domain was performed on Secondary Mushroom Dataset which is a large scale dataset. All models were subject to the same preprocessing, including feature removal, modal imputation, outlier capping based on the interquartile range, label encoding, and stratified splitting. The method which performed best (with very little overfitting) was KNN ($k = 60$) with test accuracy of 97.83% and ROC-AUC of 0.9983, followed by Decision Tree, whereas linear models and XGBoost performed fairly well despite the fact that the feature space is non-linear separable. The three most important features identified from the feature importance analysis were stem_width, cap_surface and gill_attachment, and these features provide biologically meaningful information that are consistent with the established mycological knowledge. It shows that with a suitable KNN classifier and the use of well-preprocessed tabular morphological data, an easily

implemented classification of mushroom edibility can be achieved with very high accuracy, without using deep-learning techniques with their high computational requirements.

There are a few caveats: it is a synthetic dataset, performance may vary for specimens collected in the field; the hyperparameters were not extensively tuned, which may be an underestimate of the XGBoost and Random Forest models; the analysis was solely based on tabular morphological features, excluding image-based identification; and there was no explicit treatment of class imbalance through resampling. To further improve recall on the critical false-negative class, future work should consider stacking or voting ensembles of KNN and Decision Tree classifiers, or look at using a real-time mobile application with camera-based feature extraction, and finally, explore CNN-based analysis of mushroom images to complement the tabular pipeline, and extend the framework to multi-class, species-level identification..

REFERENCES

[1] H. Li et al., "Reviewing the world's edible mushroom species: A new evidence-based classification system," Comprehensive Reviews in Food Science and Food Safety, vol. 20, no. 2, pp. 1982–2014, 2021.

[2] J. White et al., "Mushroom poisoning: A proposed new clinical classification," Toxicon, vol. 157, pp. 53–65, 2019.

[3] G. Marley, Chanterelle Dreams, Amanita Nightmares: The Love, Lore, and Mystique of Mushrooms. Chelsea Green Publishing, 2010.

[4] C. Lei et al., "Mushroom poisoning surveillance analysis, Yunnan province, China, 2001–2006," OSIR Journal, vol. 1, no. 1, pp. 8–11, 2016.

[5] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," Artificial Intelligence Review, vol. 26, no. 3, pp. 159–190, 2006.

[6] F. T. Admojo, M. L. Radhitya, H. Zein, and A. Naswin, "Classification of Mushroom Edibility Using K-Nearest Neighbors: A Machine Learning Approach," Indonesian Journal of Data and Science, vol. 5, no. 3, pp. 243–250, 2024.

[7] M. Arslan, M. Azam, M. Ali, M. U. Hashmi, A. Kousar, and Z. Zafar, "A Comparative Study of Machine Learning Methods for Optimizing Mushroom Classification," Journal of Computing & Biomedical Informatics, vol. 8, no. 1, 2024.

[8] N. Thakur and H. Thakur, "Classification of Mushroom's into Edible or Poisonous Using ML," B.Tech Project Report, Jaypee University of Information Technology, Waknaghat, 2022.

[9] D. S. L. Manikanteswari and C. S. Kalyani, "Classification of Edible and Poisonous Mushrooms Using Machine Learning Algorithms," Int. J. for Multidisciplinary Research (IJFMR), vol. 6, no. 2, 2024.

[10] G. Devika and A. G. Karegowda, "Identification of Edible and Non-Edible Mushroom Through Convolution Neural Network," in Proc. 3rd Int. Conf. Integrated Intelligent Computing Communication & Security (ICIIC 2021), Atlantis Highlights in Computer Sciences, vol. 4, pp. 312–321, 2021.

[11] K. Tutuncu, I. Cinar, R. Kursun, and M. Koklu, "Edible and Poisonous Mushrooms Classification by Machine Learning Algorithms," in Proc. 2022 11th Mediterranean Conf. Embedded Computing (MECO), Budva, Montenegro, 2022, pp. 629–632.