AI-Based Intelligent Multi-Disease Prediction System Using Adaptive Symptom Analysis - A Comparative Study Of Supervised Learning Algorithms For Clinical Disease Classification

Ms. Indhumathi S[1], Livya Grace E[2],Sneka V[3],Visvanath R[4]
[1]Assist prof, Dept of Software Systems
[2, 3, 4] Dept of Software Systems
[1, 2, 3, 4] Sri Krishna Arts and Science College

Abstract*- The integration of artificial intelligence into clinical healthcare settings has gained remarkable momentum over the past decade, offering unprecedented opportunities to augment diagnostic capabilities and extend medical reach to underserved populations. Among the many promising applications of this integration, the automated identification of diseases from patient-reported symptom profiles stands out for its potential to democratize access to preliminary healthcare screening. This paper presents an empirical comparative investigation of four supervised machine learning algorithms — Random Forest, K-Nearest Neighbors, Naive Bayes, and Support Vector Machine — evaluated on the task of predicting diseases from binary symptom feature vectors. Experiments were conducted on a structured dataset containing 4,920 patient records distributed across 41 disease categories and encoded using 131 symptom attributes. A stratified 80/20 train-test partition was employed alongside 5-fold cross-validation to ensure reliable and generalizable performance estimates. Among the four algorithms evaluated, the Support Vector Machine equipped with a Radial Basis Function kernel consistently outperformed its counterparts, attaining a test accuracy of 99.19%, a weighted F1-score of 99.16%, and a cross-validation mean of 99.27% — results that clearly exceeded the 90% performance target established at the outset of this research. Beyond the algorithmic investigation, the paper describes the development of a fully functional Flask-based web application named MediAI, which embeds the trained SVM model within an adaptive symptom collection interface capable of delivering real-time differential diagnoses, confidence-calibrated predictions, and actionable clinical guidance. These findings collectively affirm that classical supervised learning, when thoughtfully applied to well-structured clinical data, can serve as a reliable foundation for accessible and scalable disease screening systems.*

Keywords: ArdSymptom-Based Disease Prediction, Supervised Machine Learning, Support Vector Machine, Multi-Class Clinical Classification, Healthcare Artificial Intelligence, Adaptive Symptom Analysis, Comparative Algorithm Evaluation

I. INTRODUCTION

Few technological developments in recent memory carry as much transformative potential for global health as the maturation of artificial intelligence and its increasing deployment within medical contexts. Where traditional diagnostic workflows depend heavily on the availability of experienced clinicians, well-equipped facilities, and informed patients, machine learning systems operate from data alone — identifying patterns across thousands of cases and translating those patterns into actionable predictions at a speed and scale no human practitioner can match. This shift is not merely a matter of convenience; in regions where healthcare infrastructure is chronically underfunded and physician shortages are severe, AI-assisted screening tools could represent a genuinely life-saving intervention.

The early detection of disease is broadly recognized as the single most effective lever for improving patient outcomes across a wide range of conditions. Illnesses such as malaria, tuberculosis, hepatitis, and type-2 diabetes respond dramatically better to treatment when identified in their early stages, yet diagnostic delays remain common even in relatively well-resourced health systems. The reasons for these delays are multifaceted — patients may not recognize the significance of their symptoms, appointments with specialists may be weeks or months away, and general practitioners often face time constraints that limit thorough symptom investigation. Taken together, these factors create a diagnostic gap that technology is uniquely positioned to help bridge.

Symptom-based classification is one of the most tractable formulations of the disease prediction problem for machine learning. A patient's reported symptoms can be represented as a structured binary vector — each dimension corresponding to one possible symptom, set to one if present and zero if absent — and the task of mapping this vector to a disease label is a well-defined supervised classification problem. The availability of structured clinical datasets encoding such symptom-disease relationships makes it feasible to train models that learn these mappings from historical data rather than relying on hand-coded medical rules. The key question, however, is not whether machine learning can be applied to this problem, but which algorithms perform most reliably and why.

Different supervised learning algorithms bring fundamentally different inductive biases to classification tasks. Some, like Naive Bayes, make strong independence assumptions that simplify computation but may sacrifice accuracy when features are correlated. Others, like Random Forest, build their predictions from ensembles of weaker learners, trading off interpretability for improved generalization. Still others, like Support Vector Machines, frame classification as a geometric optimization problem that explicitly seeks the maximum-margin decision boundary. Choosing among these approaches without empirical evidence is essentially guesswork, which motivates the comparative experimental design at the heart of this study.

This paper describes the development and evaluation of MediAI, an intelligent multi-disease prediction system that trains all four algorithms on a common dataset, evaluates them under identical conditions, and deploys the best-performing model within a Flask web application featuring an adaptive symptom collection interface. The goal is both to identify the most suitable algorithm for this specific task and to provide a reproducible methodological framework that other researchers can build upon.

II. LITERATURE REVIEW

Research into machine learning applications for clinical disease prediction has evolved considerably over the past fifteen years, progressing from narrow single-disease classifiers toward more ambitious multi-disease systems capable of distinguishing between dozens of conditions from structured patient input. Understanding this trajectory is important for situating the present work within the broader field and for appreciating the specific gaps it addresses.

Early work in this space focused predominantly on predicting individual high-prevalence conditions — heart disease, diabetes, and various cancers attracted the most attention, partly because of the availability of publicly accessible datasets and partly because of their significant public health burden. Studies employing logistic regression, decision trees, and naive Bayes on these datasets established that relatively simple statistical models could achieve competitive diagnostic performance when features were carefully selected. However, these contributions were inherently limited in scope: a model that distinguishes diabetic from non-diabetic patients offers little help to a clinician attempting to differentiate between dengue fever, malaria, and typhoid in a febrile patient presenting with overlapping symptoms.

The transition toward multi-disease prediction systems introduced new methodological challenges. When the number of target classes grows from two to forty, the complexity of the decision boundaries increases dramatically, and algorithms that perform well in binary settings do not necessarily retain that performance in highly multi-class environments. Several research groups explored Random Forest and gradient boosting approaches for multi-class symptom-based prediction, reporting that ensemble methods offered meaningful improvements over single decision trees and naive Bayes classifiers, particularly in terms of handling class overlap and managing noise in symptom reports. Accuracy figures in these studies typically ranged from 82% to 94%, depending on dataset size, disease coverage, and evaluation methodology.

Support Vector Machines were identified early in the literature as theoretically attractive for high-dimensional medical classification problems. Their ability to operate effectively in feature spaces where the number of dimensions approaches or exceeds the number of training samples — a common situation in clinical data — gave them a natural advantage over algorithms that struggle with the curse of dimensionality. Applied studies confirmed this theoretical advantage in contexts ranging from cancer subtype classification to mental health disorder detection, though direct applications to symptom-based multi-disease prediction remained relatively sparse. Where such applications were reported, SVM consistently ranked among the top performers, though few studies provided the kind of rigorous cross-validation evidence needed to support confident conclusions about generalizability.

A recurring limitation across this body of research is the inconsistency of evaluation methodology. Many published studies reported a single train-test split accuracy without cross-validation, making it difficult to assess whether reported results reflected genuine model capability or favorable data partitioning. Equally problematic was the widespread use of imbalanced datasets, in which common conditions were over-represented relative to rarer ones, artificially inflating overall accuracy while masking poor performance on minority classes. These methodological weaknesses argue for a more rigorous comparative approach — one that employs stratified sampling, cross-validation, and per-class performance metrics — which is precisely the approach adopted in the present study.

A further gap in the existing literature concerns the interface between algorithmic performance and practical usability. Most published systems report model accuracy in isolation, with little attention to how symptoms are collected from users, how results are communicated, or how the system integrates into a realistic patient interaction workflow. A model that achieves 99% accuracy in research setting but requires users to navigate an unwieldy checklist of 130 symptoms offers limited practical value. The present work attempts to address both dimensions simultaneously — rigorous algorithmic evaluation alongside a thoughtfully designed adaptive interface — in service of a system that is both technically sound and genuinely usable.

III. METHODOLOGY

A. Overall System Architecture

MediAI is organized around a three-tier architecture that separates data management, application logic, and user presentation into distinct layers. The foundational layer houses the training dataset, the serialized model artifacts produced by the training pipeline, and a metadata store encoding disease-specific clinical knowledge including symptom profiles, precautionary recommendations, and specialist referral information. The middle layer is implemented in Python using Flask and handles all server-side responsibilities: routing incoming HTTP requests, executing prediction logic, managing the SQLite analytics database, and generating PDF reports. The presentation layer delivers the user interface through HTML5 templates rendered server-side by Flask's Jinja2 templating engine and progressively enhanced with Bootstrap 5 and vanilla JavaScript for interactivity.

B. Dataset Construction and Characteristics

The dataset used throughout this study contains 4,920 patient records, each described by 131 binary symptom features and labeled with one of 41 disease categories. Class balance was deliberately enforced, with exactly 120 records per disease, ensuring that no condition receives disproportionate representation during training and that per-class performance metrics are directly comparable across diseases. The 41 diseases span a broad clinical spectrum: infectious conditions including Malaria, Dengue, Typhoid, all five hepatitis variants, Tuberculosis, Chicken Pox, and AIDS; endocrine and metabolic disorders including Diabetes, Hyperthyroidism, Hypothyroidism, and Hypoglycemia; gastrointestinal conditions including GERD, Peptic Ulcer Disease, and Gastroenteritis; neurological conditions including Migraine, Vertigo, and Paralysis; cardiovascular conditions including Hypertension and Heart Attack; respiratory illnesses including Pneumonia, Bronchial Asthma, and Common Cold; and dermatological conditions including Acne, Psoriasis, Fungal Infection, and Impetigo.

Each disease profile was constructed from validated medical knowledge, distinguishing between primary symptoms that reliably accompany the condition in virtually all patient presentations and secondary symptoms that manifest with approximately 70% probability, reflecting the natural heterogeneity of real-world clinical presentations. To further simulate clinical realism, one to two randomly selected noise symptoms were introduced into each record, representing the tendency of patients to report incidental complaints alongside their primary condition. The resulting dataset closely mirrors the structure and content of the publicly available Kaggle Disease Symptom Prediction dataset, using identical disease labels and symptom vocabulary, allowing the methodology to be validated against an externally recognized benchmark.

C. Preprocessing Pipeline

Data preprocessing proceeded through three sequential stages. In the first stage, the completeness of the dataset was verified — no missing values were present, a direct consequence of the binary encoding scheme's requirement for explicit 0 or 1 assignment to every feature. In the second stage, the string-valued target variable was encoded into integers using scikit-learn's LabelEncoder, which assigns a unique integer to each of the 41 disease names through alphabetical ordering. The fitted encoder was serialized to disk alongside the trained model so that predictions expressed as integers could be unambiguously decoded back to their corresponding disease names at inference time. In the third stage, the numerical consistency of the feature matrix was confirmed, with all 131 columns verified as integer-valued binary features requiring no further transformation.

D. Experimental Design

The full dataset was partitioned into training and test subsets using an 80/20 stratified random split, yielding 3,936 training samples and 984 test samples. Stratification on the target variable ensured that the proportion of each disease class was preserved identically in both subsets, eliminating the risk that any class would be systematically under-represented in the test set. Complementing the holdout evaluation, 5-fold stratified cross-validation was applied to each algorithm using the complete dataset. This procedure divides the data into five equal folds, trains the model five times with a different fold held out as a validation set each time, and averages the five resulting accuracy scores. The cross-validation standard deviation provides an additional measure of stability — models with low standard deviation generalize consistently across different data configurations.

E. Algorithms Under Evaluation

Random Forest constructs its predictions by aggregating the outputs of a large collection of individual decision trees, each trained on a bootstrap sample of the training data with a random subset of features considered at each split point. This dual randomization reduces inter-tree correlation and substantially lowers the variance of the ensemble relative to any single tree. The default configuration employed 100 estimators, and a hyperparameter-tuned variant was additionally evaluated using GridSearchCV, which exhaustively searched combinations of tree count (100, 200), maximum depth (unconstrained, 20 levels), and minimum samples per split (2, 5) using 3-fold internal cross-validation.

K-Nearest Neighbors operates by retaining the entire training dataset and classifying each new input according to the majority class among its k nearest neighbors, where proximity is measured using Euclidean distance in the feature space. The algorithm's non-parametric nature makes it free from distributional assumptions, though this flexibility comes at the cost of increased sensitivity to irrelevant features and higher inference-time computational requirements as the training set grows. A neighborhood size of k equals 5 was used throughout this study.

Naive Bayes, in its Gaussian formulation, applies Bayes' theorem to compute the posterior probability of each class given the observed feature values, under the assumption that features are conditionally independent given the class label and normallydistributed within each class. While both assumptions are technically violated by binary symptom data — symptoms are correlated, and binary values are not Gaussian — the algorithm's performance in practice often exceeds what its theoretical limitations might suggest, owing to the partial cancellation of errors introduced by the independence assumption across many features.

Support Vector Machine approaches classification as a constrained optimization problem, seeking the hyperplane in the feature space that maximizes the geometric margin between classes. When classes are not linearly separable in the original feature space — as is generally the case with high-dimensional symptom data — the kernel trick is employed to implicitly map the data into a higher-dimensional representation where a linear separator exists. The Radial Basis Function kernel was selected for this study, as it introduces no prior assumptions about the shape of the decision boundary and has demonstrated strong empirical performance across a wide range of classification tasks. Probability calibration was enabled through Platt scaling, allowing the model to produce confidence scores alongside class predictions.

IV. SYSTEM IMPLEMENTATION

The deployment architecture centers on a Flask application that loads the serialized SVM model, label encoder, and metadata dictionary once at startup, retaining them in memory for the duration of the application's runtime. This approach avoids the overhead of repeated disk reads during prediction requests and ensures that all incoming queries are served by identical model artifacts, maintaining prediction consistency.

The patient interface is structured around a two-stage adaptive symptom collection mechanism built on top of nine clinically motivated symptom groups, each corresponding to a major body system or disease category. In the first stage, users are presented with a curated selection of primary trigger symptoms — broadly recognizable complaints such as fever, cough, headache, joint pain, and skin rash — rendered as interactive selection chips within a searchable grid. Once initial selections are submitted, the application evaluates which symptom groups have been activated and dispatches a set of targeted follow-up questions relevant specifically to those groups via an asynchronous server request, dynamically populating a second selection panel without requiring a full page reload. This staged approach substantially reduces the cognitive load on users compared to presenting all 131 symptoms simultaneously, while ensuring the model receives a sufficiently informative feature vector for accurate prediction.

Prediction requests are processed by a dedicated function that receives the list of selected symptom names, initializes a 131-dimensional zero vector, and sets each position corresponding to a selected symptom to one. This binary vector is passed to the model's probability estimation method, which returns a distribution over all 41 disease classes. The three highest-probability classes are extracted and returned to the user as a ranked differential diagnosis. Each prediction is simultaneously recorded in the SQLite analytics database, capturing the predicted disease, confidence score, severity classification, symptom list, and timestamp for subsequent administrative review.

V. EXPERIMENTAL RESULTS AND COMPARATIVE ANALYSIS

The four algorithms — alongside the GridSearchCV -tuned Random Forest variant — were each evaluated on the same 984-sample test set, with performance measured across accuracy, weighted precision, weighted recall, and weighted F1-score. Weighted averaging was applied across all multi-class metrics to reflect the equal class distribution in the test set, ensuring that each disease contributesproportionally to the reported figures.

Algorithm	Accuracy	Precision	Recall	F1-Score	CV Mean	CV Std
Naive Bayes	97.87%	97.96%	97.87%	97.85%	98.19%	±0.38%
Random Forest	98.48%	98.48%	98.48%	98.46%	99.04%	±0.34%
KNN (k=5)	98.68%	98.73%	98.68%	98.70%	99.02%	±0.24%
RF (Tuned)	98.68%	98.74%	98.68%	98.70%	99.06%	±0.29%
SVM (RBF)	**99.19%**	**99.39%**	**99.19%**	**99.16%**	**99.27%**	**±0.12%**

The first notable observation from these results is that every evaluated configuration comfortably exceeded the 90% accuracy threshold, validating the hypothesis that symptom-disease classification is a learnable problem for a range of standard supervised learning approaches. The margin between the worst-performing algorithm (Naive Bayes at 97.87%) and the best (SVM at 99.19%) is approximately 1.3 percentage points — modest in absolute terms, but meaningful when translated to a clinical context where each percentage point represents a measurable number of misclassified patient cases.

Naive Bayes produced the lowest results across all metrics, an outcome attributable to two structural mismatches between the algorithm's assumptions and the nature of the data. The Gaussian distribution assumption is poorly suited to binary feature values, which take only two discrete states rather than a continuous range. Additionally, the conditional independence assumption fails to capture the clinically meaningful co-occurrence patterns between symptoms — for instance, the joint presence of high fever, chills, and sweating carries stronger diagnostic signal for Malaria than any single symptom alone, a relationship Naive Bayes cannot represent.

Random Forest improved meaningfully over Naive Bayes, with GridSearchCV tuning pushing it to 98.68% — equal to KNN's performance. The relatively modest improvement from tuning suggests the default configuration was already near-optimal for this dataset, and that the performance ceiling for tree-based methods on this particular feature space lies somewhat below that achievable by margin-based approaches. The cross-validation standard deviation of ±0.34% for the default Random Forest, while not alarming, indicates slightly more sensitivity to data partitioning than either KNN or SVM.

KNN's competitive performance at 98.68% reflects the tight geometric clustering of symptom vectors belonging to the same disease in the 131-dimensional feature space. Because patients with the same disease share a largely consistent set of symptoms, their binary vectors tend to be similar, making nearest-neighbor retrieval an effective classification strategy. However, KNN's cross-validation mean of 99.02% with a standard deviation of ±0.24% trails SVM on both counts, indicating marginally less consistent generalization.

The Support Vector Machine delivered the strongest performance on every evaluated metric without exception. Beyond the headline accuracy figure of 99.19%, what is particularly noteworthy is the cross-validation standard deviation of ±0.12% — the lowest of all evaluated models, indicating that SVM's performance is highly stable across different data configurations. Examining the per-class classification report reveals near-perfect precision and recall for 39 of the 41 disease classes, with the only meaningful confusion occurring between Hepatitis D and Hepatitis E — two conditions whose symptom profiles are clinically very similar, making their occasional confusion an understandable and practically acceptable limitation.

The SVM's advantage can be understood through several complementary lenses. The 131-dimensional binary feature space is, by conventional standards, high-dimensional relative to the number of training samples per class (approximately 96 samples per class in the training set). In such settings, SVM's margin maximization objective tends to produce more robust decision boundaries than algorithms that optimize for different objectives, because the support vectors nearest the boundary concentrate the model's attention on the most informative and discriminative samples rather than distributing equal weight across all training points. Furthermore, the RBF kernel's implicit mapping into infinite-dimensional space allows the SVM to capture non-linear symptom interaction effects that tree splits and Euclidean distance measures cannot fully represent.

VI. DISCUSSION

The results of this study invite reflection on both the algorithmic findings and their broader implications for the design of healthcare AI systems. On the algorithmic side, the consistent superiority of SVM across metrics and cross-validation folds supports the conclusion that margin-based classifiers are particularly well-matched to the structural properties of binary symptom data — high dimensionality, sparse activation, and tight intra-class clustering. This finding is consistent with the theoretical literature on SVM performance in high-dimensional spaces and adds empirical weight to existing evidence from related medical classification tasks.

It is worth emphasizing, however, that the relatively strong performance of all four algorithms is itself a meaningful finding. A pessimistic reading of the literature might suggest that multi-class disease prediction from symptom data is intractably difficult due to symptom overlap and the inherent ambiguity of patient-reported data. The results presented here challenge that pessimism: even the worst-performing algorithm, Naive Bayes, achieved nearly 98% accuracy under rigorous evaluation conditions, suggesting that the symptom-disease signal is sufficiently strong to support reliable classification by a range of methods.

The practical implications of selecting SVM over its alternatives extend beyond the accuracy differential. The SVM's ability to produce calibrated probability outputs through Platt scaling enables the system to communicate diagnostic uncertainty in a clinically meaningful way. Rather than presenting a single hard prediction, MediAI returns a ranked list of the three most probable diagnoses with associated confidence percentages, giving users and clinicians a richer picture of the model's reasoning. This differential diagnosis capability — presenting multiple possibilities ordered by likelihood — more closely mirrors the reasoning process of a physician than a system that commits to a single prediction.

The adaptive symptom collection interface merits separate discussion as a contribution in its own right. While the bulk of the paper has focused on algorithmic performance, the interface design choices significantly influence the quality and completeness of the symptom data the model receives, and therefore indirectly affect prediction accuracy in real-world deployment. By presenting symptom questions in clinically coherent groups and adapting follow-up questions to the patient's initial responses, the interface increases the likelihood that users report all relevant symptoms, minimizing the information loss that would occur if they were asked to self-select from an undifferentiated list of 131 options. This human-computer interaction consideration is often underemphasized in machine learning research but is critical to translating laboratory accuracy into genuine clinical utility.

VII.CONCLUSION

This paper has reported the design, experimental evaluation, and deployment of MediAI, an intelligent multi-disease prediction system built around a comparative evaluation of four supervised machine learning algorithms. Using a structured dataset of 4,920 records spanning 41 disease categories and 131 binary symptom features, Random Forest, K-Nearest Neighbors, Naive Bayes, and Support Vector Machine were each trained under identical conditions and assessed using accuracy, precision, recall, F1-score, and cross-validation statistics. The findings clearly establish SVM with an RBF kernel as the best-performing algorithm for this task, achieving 99.19% test accuracy and a cross-validation mean of 99.27% — the highest scores recorded across all metrics and all evaluated configurations.

These results have practical significance beyond the academic context. They demonstrate that a classical supervised learning algorithm, when trained on a well-structured and balanced symptom dataset, can achieve accuracy levels that are clinically meaningful for preliminary disease screening. The complete system — from data preprocessing through model training, web application development, adaptive interface design, PDF report generation, and administrative analytics — constitutes a coherent proof of concept for AI-assisted symptom-based disease screening that could, with appropriate validation and regulatory consideration, be extended toward real-world clinical deployment.

VIII. FUTURE WORK

Several directions offer promising avenues for building upon this foundation. The most impactful near-term enhancement would involve replacing the current dataset with de-identified patient records sourced from actual clinical encounters, ideally in partnership with healthcare institutions. Real-world data would introduce the full complexity of genuine patient presentations — including rare symptom combinations, co-morbidities, and reporting inconsistencies — and would allow the model's real-world accuracy to be empirically validated rather than inferred from synthetic benchmarks.

A second direction concerns the expansion of the disease vocabulary. The current system covers 41 conditions, which while broad is still a fraction of the thousands of conditions a clinician might encounter. Extending coverage to 100 or more diseases, incorporating rare and regionally specific conditions, would substantially increase the system's clinical relevance. This expansion would also create a more challenging classification problem and would provide a stronger test of the relative merits of different algorithmic approaches.

The symptom input modality presents another opportunity for meaningful improvement. Replacing the current structured selection interface with a natural language processing component capable of extracting symptom mentions from free-text patient descriptions would lower the barrier to entry for users unfamiliar with medical terminology. Transformer-based language models pretrained on biomedical text corpora offer a compelling starting point for this enhancement.

From a deployment perspective, packaging the application as a Progressive Web App or native mobile application would substantially expand its geographic reach, particularly in contexts where smartphone access is more common than desktop computing. Integration with telemedicine platforms and hospital information systems would enable the prediction outputs to flow directly into clinical workflows, increasing the likelihood that AI-generated insights translate into improved patient care.

Finally, exploring deep learning architectures — including multi-layer perceptrons, convolutional networks applied to symptom co-occurrence matrices, and graph neural networks encoding clinical ontologies — on substantially larger datasets would allow a more complete picture of the performance ceiling for this class of problems and would position the research at the frontier of clinical AI.

REFERENCES

[1] D. Singh and B. Kumar, "A Systematic Review of Machine Learning Algorithms for Disease Prediction Using Symptom-Based Datasets," *Journal of Medical Systems*, vol. 44, no. 3, pp. 1–15, Mar. 2020.
[2] S. Garg, A. Sharma, and R. Verma, "An Intelligent Disease Prediction System Using Machine Learning Algorithms," *International Journal of Engineering Research and Technology*, vol. 10, no. 5, pp. 234–241, May 2021.
[3] S. Priya and K. Mahesh, "Multi-Disease Prediction Using Ensemble Learning Approach," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 4, pp. 1981–1992, Jul.–Aug. 2022.
[4] P. Sahu and A. Mehrotra, "Healthcare Chatbot with Symptom-Based Disease Prediction Using Natural Language Processing and Machine Learning," *Expert Systems with Applications*, vol. 211, pp. 1–14, Jan. 2023.
[5] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[6] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY, USA: Springer-Verlag, 2000.
[7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.
[8] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
[9] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, 1995, pp. 1137–1143.
[10] Disease Symptom Prediction Dataset, Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset [Accessed: Jan. 2024]