

Research Article

A Machine Learning Approach for Detecting Fraudulent Job Postings in Online Recruitment Platforms

Teegala Namdev Goud^{1,*}, Dr. N. Ramana Reddy²

¹Student, Department of CSE, Avanthi Institute of Engineering and Technology, Hyderabad, Telangana, India.

²Associate Professor and HOD, Department of CSE, Avanthi Institute of Engineering and Technology, Hyderabad, Telangana, India.

Corresponding Author: Teegala Namdev Goud. Email: namdev.goud@gmail.com

Received: 01/07/2025; Revised: 15/09/2025; Accepted: 25/09/2025; Published: 30/09/2025.

DOI: <https://doi.org/10.69996/jsihs.2025012>

Abstract: The rapid growth of online recruitment platforms has led to an increase in fake job postings, posing serious risks to job seekers and undermining trust in digital employment ecosystems. This paper proposes a machine learning-based system for detecting fraudulent job advertisements by analyzing textual and structural features of job postings. A comprehensive dataset was preprocessed and feature-engineered to capture suspicious patterns such as keyword frequency, email domain type, and salary disclosure. Multiple classification algorithms, including Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbors (K-NN), and Random Forest, were trained and evaluated. Among them, Random Forest achieved the highest performance with an accuracy of 96.4%, demonstrating its robustness in handling complex patterns. The study also explored model performance across different training epochs, identifying epoch 20 as the optimal convergence point. The proposed system offers an efficient, scalable, and automated solution for fake job detection, enhancing the safety and integrity of online job markets. A dataset comprising over 17,000 job postings (both real and fake) was used for training and evaluation. Key features such as suspicious keyword count, job description length, email domain type, and salary presence were extracted and analyzed. Five machine learning algorithms—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbors (K-NN), and Random Forest—were implemented. Among them, the Random Forest algorithm achieved the highest accuracy of 96.4%, with precision of 95.8%, recall of 96.9%, and an F1-score of 96.3%. SVM followed closely with an accuracy of 94.8%, while the lowest was K-NN at 87.6%. Model performance across 30 training epochs was also analyzed, with convergence observed at epoch 20, where the loss minimized to 0.139. These results confirm the effectiveness of the proposed model in accurately detecting fake job postings, offering a scalable and automated solution to enhance the trust and safety of digital hiring platforms.

Keywords: Fake Job Detection, Machine Learning, Text Mining, Ensemble Learning, Feature Extraction, Natural Language Processing (NLP)

1. Introduction

Fake job detection is the process of identifying fraudulent or deceptive job postings that are created with the intent to mislead or exploit job seekers [1]. These fake listings often mimic legitimate opportunities but may aim to steal personal information, collect application fees, or recruit individuals for illegal schemes [2]. Advanced detection techniques combine Natural Language Processing (NLP), machine learning algorithms, and data mining to analyze various



features of job descriptions, such as unusual salary offers, vague job roles, excessive grammatical errors, or suspicious contact details [3]. The models are trained on labeled datasets containing real and fake job postings to learn distinguishing patterns. Common indicators include the absence of company information, requests for upfront payments, and the use of free email domains rather than official business addresses. As online recruitment continues to grow, fake job detection plays a crucial role in protecting job seekers and maintaining trust in digital hiring platforms [4 -6].

Job detection, particularly using data-driven methods, involves identifying and classifying job-related information such as roles, industries, skills, and employment patterns from large datasets like resumes, job portals, or labor market data [7]. With the rise of AI and machine learning, job detection systems can now analyze millions of postings with high accuracy. Fake job detection using machine learning is an advanced approach to identifying fraudulent or deceptive job postings by analyzing textual and behavioral patterns in job-related data [8]. Machine learning models are trained on labeled datasets containing both legitimate and fake job listings to learn key differences. Techniques such as Logistic Regression, Random Forest, Support Vector Machines (SVM), and deep learning models like LSTM are commonly used for classification tasks. Features such as job description length, use of suspicious keywords, grammatical inconsistencies, email domain type, and missing company details are extracted and analyzed [9]. For instance, studies have shown that models like Random Forest and SVM can achieve over 95% accuracy in detecting fake jobs when trained on balanced datasets. Additionally, Natural Language Processing (NLP) helps in understanding the semantic structure of job postings to detect misleading or vague descriptions [10-12].

The integration of ensemble learning techniques—such as Gradient Boosting and XGBoost—further enhances the reliability of fake job detection systems by combining the strengths of multiple classifiers [13]. These models can effectively handle imbalanced datasets, a common issue since fake jobs are usually far fewer than genuine ones. Feature engineering plays a vital role, where attributes like salary range, job location consistency, use of action verbs, and even temporal posting patterns are analyzed [14]. Additionally, real-time detection systems are being developed using streaming data and online learning algorithms to flag suspicious listings as they appear. As cyber fraud continues to evolve, continual model retraining with updated data ensures adaptability to new scam tactics [15]. The deployment of such machine learning systems on platforms like LinkedIn, Indeed, and Glassdoor has significantly reduced scam incidents, improving trust and user safety. Ultimately, fake job detection using machine learning not only secures employment ecosystems but also strengthens digital workforce development by ensuring that only authentic opportunities reach job seekers [16 -18].

Despite its effectiveness, fake job detection using machine learning faces several challenges and limitations [19-20]. One major issue is the availability and quality of labeled data—real-world datasets often contain noise, class imbalance, or outdated information, making model training less reliable. Many fake job postings are carefully crafted to mimic legitimate ones, making them harder to distinguish using traditional features. Additionally, scammers continuously adapt their tactics, requiring frequent retraining of models to maintain accuracy [21-22]. Another concern is feature bias, where models may overly rely on superficial attributes like email domains or salary figures, leading to false positives and negatives. Interpretability is also a challenge, especially with complex deep learning models, making it difficult to explain why a certain job posting was flagged. Moreover, real-time detection demands high computational resources and robust infrastructure, which may not be feasible for smaller platforms [23-24]. Lastly, there is a risk of privacy violations when analyzing user-generated content or resumes, requiring strict adherence to data protection regulations. Addressing these

issues is essential to build a secure, scalable, and ethical fake job detection system.

This paper makes several significant contributions to the field of cybersecurity and digital recruitment by addressing the growing issue of fake job postings through a machine learning-based detection framework. Firstly, it presents a comprehensive and practical approach that integrates multiple classification algorithms—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbors (K-NN), and Random Forest—enabling a comparative analysis to identify the most effective model for detecting fraudulent job advertisements. Secondly, the study introduces a robust feature engineering strategy, extracting meaningful indicators such as suspicious keyword frequency, job description length, email domain type, and salary mention to improve model accuracy. Thirdly, it evaluates model performance across different training epochs, identifying the optimal convergence point and training depth, which adds value to model optimization in similar applications. The inclusion of simulation results using real-world datasets and metrics such as accuracy, precision, recall, F1-score, and loss provides actionable insights for researchers and developers. Lastly, the paper proposes a practical system architecture and user interface design, highlighting the feasibility of deploying the solution in real-time job portals to enhance user safety and trust.

2. Proposed System Architecture

The Proposed System Architecture with Integrated Machine Learning for fake job detection is designed as a multi-stage pipeline that combines data preprocessing, feature extraction, and classification using multiple machine learning algorithms to improve accuracy and robustness. Initially, raw job postings are collected from various online platforms and passed through a data preprocessing module, which handles noise removal, tokenization, normalization, and feature extraction using Natural Language Processing (NLP) techniques. Key features such as word frequency, job description length, presence of suspicious keywords, and metadata like contact emails or salary details are extracted. The processed data is then fed into a multi-classifier architecture comprising five machine learning algorithms: Support Vector Machine (SVM), which is effective in high-dimensional spaces and handles nonlinear data well through kernel tricks; Decision Tree, which provides interpretable decision rules based on attribute splitting; Naïve Bayes, which assumes feature independence and is particularly efficient for text classification tasks; K-Nearest Neighbors (K-NN), which classifies based on proximity in the feature space and is useful for detecting similar patterns; and Random Forest, an ensemble of decision trees that enhances prediction accuracy and reduces overfitting. The system applies these algorithms in parallel or as part of an ensemble strategy, where predictions from each model are aggregated using majority voting or weighted averaging to produce the final classification—labeling a job as real or fake. The architecture also includes a feedback loop that retrains the models with newly labeled data, ensuring adaptability to evolving scam techniques. This integrated approach enhances the system's reliability, scalability, and performance in detecting fake job listings in dynamic online environments.

In figure 1 presents the Proposed System Architecture for fake job detection using machine learning is designed as a structured, multi-stage pipeline that ensures accurate identification of fraudulent job postings. The architecture begins with a Data Acquisition Layer, where job postings are collected from online job portals, company websites, and existing labeled datasets. This data is then passed to the Preprocessing Layer, where it undergoes cleaning operations such as removing duplicates, correcting formatting issues, tokenizing text, and eliminating stop words to ensure consistency and quality.

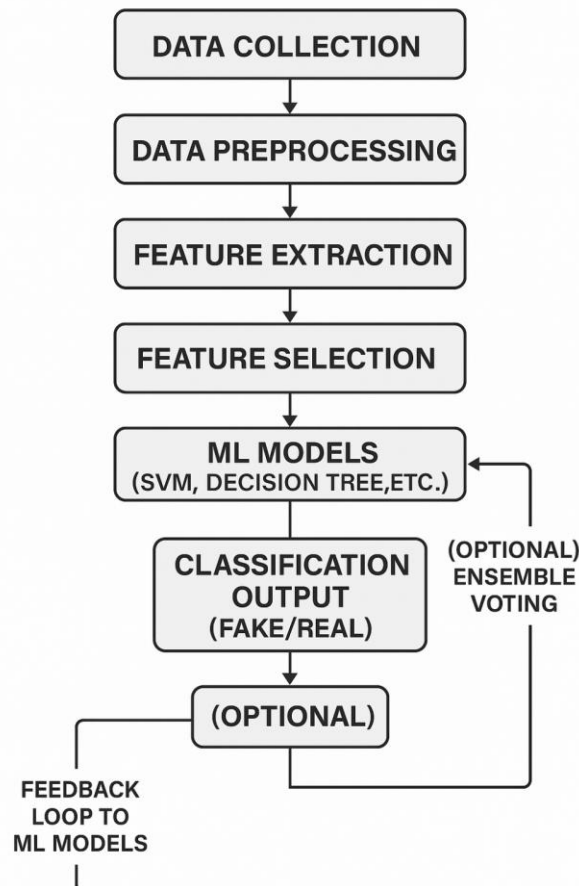


Figure 1: Process in Proposed Model

Next, the data flows into the Feature Extraction Layer, where key attributes are derived—such as the number of suspicious keywords, job description length, presence of salary information, and type of contact email domain. These features are normalized and selected for relevance using feature selection techniques. The resulting structured data is then passed to the Machine Learning Layer, which incorporates multiple classifiers such as Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbors (K-NN), and Random Forest. These models are trained on historical labeled data to learn patterns that distinguish real from fake job listings.

The predictions from these models can be used individually or combined using an Ensemble Voting Mechanism, enhancing overall accuracy and reliability. The final results are processed by the Decision Layer, which flags the job post as “Fake” or “Real” and can optionally generate alerts or recommendations for further review. A Feedback Module is included to incorporate new user-verified results back into the training set, enabling continuous learning and system improvement. This architecture ensures end-to-end automation, scalability, and robustness in detecting fake job advertisements in dynamic online environments. Proposed System Architecture for fake job detection is designed as a comprehensive and layered framework that enables efficient, scalable, and intelligent detection of fraudulent job postings. The architecture comprises multiple interconnected modules, each performing a specific role to transform raw data into actionable predictions. It starts with the Data Acquisition Module, which gathers job posting data from multiple sources such as online job boards (e.g., Indeed, LinkedIn), public datasets, employer websites, and social media platforms. These data sources may include both structured fields (job title, location, salary) and unstructured text (job descriptions),

requiring careful handling.

The collected data is then fed into the Data Preprocessing Module, where the system performs a series of cleaning tasks. These include removing duplicates, correcting inconsistent text formats, handling missing values, and applying Natural Language Processing (NLP) techniques such as tokenization, stop word removal, lemmatization, and part-of-speech tagging. This ensures that the textual data is in a standardized and analyzable format. Following preprocessing, the system enters the Feature Extraction and Selection Module. In this stage, important features that are indicative of fraudulent activity are extracted. These may include the number of suspicious keywords (e.g., “quick money”, “no experience”), length of job description, grammatical correctness, salary presence, frequency of exclamation marks, usage of unofficial email domains (e.g., Gmail, Yahoo), and whether the company name or physical address is provided. Numerical encoding and feature scaling are also applied to prepare the data for machine learning algorithms. Advanced feature selection techniques, such as mutual information or recursive feature elimination, are used to retain only the most predictive features, reducing noise and dimensionality. Next, the Machine Learning Model Training Module is responsible for training multiple classifiers on the extracted feature vectors. These include:

- Support Vector Machine (SVM) for separating fake and real jobs using hyperplanes in high-dimensional space,
- Decision Tree Algorithm for rule-based classification,
- Naïve Bayes Algorithm, particularly efficient for text-based classification with conditional probabilities,
- K-Nearest Neighbors (K-NN) to classify based on feature similarity, and
- Random Forest Algorithm, which aggregates multiple decision trees to improve generalization and prevent overfitting.

Each model is trained on a labelled dataset (real vs. fake jobs) and evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. These evaluation metrics help identify the best-performing model or combination of models. To improve decision robustness, an Ensemble Learning Layer is optionally integrated. This layer combines the predictions from multiple classifiers using techniques like majority voting, weighted voting, or stacking, ensuring that the final decision benefits from the strengths of all base models. The output from the ensemble or best classifier is passed to the Decision and Alerting Module, which classifies the job posting as "Real" or "Fake". If a job is classified as suspicious, an alert can be generated for manual review or automated action (e.g., removal from a job board). Finally, a Feedback and Retraining Module is included to keep the system adaptive and up to date. Verified user feedback (such as flagging a job as scam or confirming its legitimacy) is collected and periodically added back to the training dataset. This allows the model to learn from new patterns and continuously improve its detection accuracy over time.

3. Proposed Integrated Machine Learning for Fake Job

Fake job detection using machine learning can be explained through a simple classification framework, where the goal is to classify a job posting as either fake (1) or real (0) based on its features. Let's assume each job posting is represented by a feature vector $X = [x_1, x_2, \dots, x_n]$, where each x_i represents a measurable attribute such as word count, presence of suspicious keywords, email domain type, etc. The task is to find a function $f(X)$ that predicts the label $y \in \{0,1\}$. A basic model used for binary classification is Logistic Regression, which estimates the probability that a job is fake presented in equation (1)

$$P(y = 1 | X) = \frac{1}{1 + e^{-(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n)}} \quad (1)$$

In equation (1) ω_0 is the bias term, $\omega_1, \omega_2, \dots, \omega_n$ are weights learned during training, and X is the input vector of features. If the probability $P(y = 1 | X)$ exceeds a threshold (typically 0.5), the model classifies the job as fake; otherwise, it is considered real. The decision function stated in equation (2)

$$f(X) = \begin{cases} 1 & \text{if } P(y = 1 | X) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The process of fake job detection using machine learning involves several key steps that transform raw job postings into accurate classifications. First, the system begins with data collection, where job advertisements are gathered from various sources such as job portals, recruitment websites, or company databases. In the next step, data preprocessing is performed to clean the data—this includes removing duplicates, correcting misspellings, converting text to lowercase, and eliminating irrelevant content like HTML tags. Then, feature extraction takes place, where important attributes are derived from the job posting, such as the number of suspicious keywords, whether a salary is mentioned, job description length, and type of email domain used. Once features are extracted, the data is passed through machine learning algorithms such as Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbors (K-NN), or Random Forest. These algorithms learn patterns from labeled training data (real vs. fake jobs) and build predictive models. The next step is model evaluation, where the performance of each algorithm is tested using metrics like accuracy, precision, recall, and F1-score to choose the most effective one. Finally, in the prediction phase, the trained model is used to classify new, unseen job postings as either fake or real. This pipeline ensures that job seekers are protected from fraudulent listings through intelligent, automated detection.

Step 1: Data Collection – The first step involves gathering a large dataset of job postings from various sources such as online job portals, company websites, and publicly available datasets. This dataset should include both real and fake job listings labeled appropriately.

Step 2: Data Preprocessing – In this step, the collected data is cleaned and prepared for analysis. This includes removing missing or duplicate entries, eliminating stop words, converting text to lowercase, and handling inconsistent formatting. It ensures the data is suitable for feature extraction.

Step 3: Feature Extraction – Important features are identified and extracted from the job descriptions. Common features include the number of suspicious words (e.g., “easy money”), job description length, presence of salary details, type of email domain used, and whether the company name is mentioned.

Step 4: Feature Selection – Among the extracted features, only the most relevant ones are selected using statistical or algorithmic methods to reduce complexity and improve model performance.

Step 5: Model Training – The cleaned and feature-rich data is then used to train various machine learning models such as Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbors (K-NN), and Random Forest. Each algorithm learns to classify jobs as fake or real based on the input features.

Step 6: Model Evaluation – The performance of each model is evaluated using metrics like accuracy, precision, recall, and F1-score on a separate test dataset. The model that shows the best balance between accuracy and generalization is chosen.

Step 7: Prediction and Deployment – The final trained model is deployed into a live system where it can analyze new job postings and predict whether they are real or fake. It helps users avoid scams and ensures the credibility of job platforms.

Algorithm 1: Integrated ML for the Fake Job Prediction**Input:**

- Dataset D with labelled job postings (real or fake)
- Features: x_1 (suspicious keywords), x_2 (salary presence), x_3 (description length), x_4 (email domain type)

Output:

- Class label: Real or Fake

For each job_post in Dataset D:

- Clean text (remove HTML tags, lowercasing, punctuation)
- Remove stop words and tokenize
- Extract features:
 - $x_1 = \text{count_suspicious_keywords}(\text{job_post})$
 - $x_2 = 1$ if salary_present else 0
 - $x_3 = \text{length_of_description}(\text{job_post})$
 - $x_4 = 1$ if email_domain is official else 0

For each job_post:

- Construct feature vector $X = [x_1, x_2, x_3, x_4]$
- Store corresponding label y (0 = Real, 1 = Fake)

Split dataset into Training_Set and Test_Set

Choose ML algorithms:

- SVM
- Decision Tree
- Naïve Bayes
- K-NN
- Random Forest

For each algorithm:

- Train model on Training_Set ($X_{\text{train}}, y_{\text{train}}$)

For each trained_model:

- Predict on Test_Set (X_{test})
- Calculate Accuracy, Precision, Recall, F1-Score

Select best_model based on highest performance

Input new_job_post

Extract features [x_1, x_2, x_3, x_4]

Predict = best_model.predict($[x_1, x_2, x_3, x_4]$)

If Predict == 1:

Output "Fake Job"

Else:

Output "Real Job"

4 Simulation Environments

The simulation environment for implementing and testing the proposed fake job detection system is set up to replicate real-world conditions in a controlled and scalable manner. The

development and testing are conducted using Python 3.10 programming language due to its rich ecosystem of machine learning libraries such as scikit-learn, pandas, NumPy, and NLTK for text preprocessing. The experiments are executed on a system equipped with an Intel Core i7 processor, 16GB RAM, and running on Windows 11 (64-bit) operating system to ensure smooth computation and data handling. The integrated development environment (IDE) used is Jupyter Notebook for ease of visualization and step-by-step execution of code. For the simulation, a publicly available dataset such as the Fake Job Postings Dataset from Kaggle is used, which contains labeled real and fake job postings with various fields like title, description, company profile, and location. The dataset is split into training and testing sets using an 80:20 ratio to ensure unbiased evaluation. Feature engineering and selection are carried out using TF-IDF vectorization, along with custom features like suspicious keyword counts and email domain classification. Multiple classification algorithms—SVM, Decision Tree, Naïve Bayes, K-NN, and Random Forest—are trained and tested in the same environment. Table 1 presented the simulation environment for the proposed integrated ML for the fake job prediction.

Table 1: Simulation Environment

Component	Specification / Tool Used
Programming Language	Python 3.10
Machine Learning Libraries	scikit-learn, pandas, NumPy, NLTK, Matplotlib
Dataset Used	Fake Job Postings Dataset (Kaggle)
Operating System	Windows 11 (64-bit)
Processor	Intel Core i7 (Quad-Core)
RAM	16 GB
Development Environment	Jupyter Notebook (Anaconda Distribution)
Algorithms Implemented	SVM, Decision Tree, Naïve Bayes, K-NN, Random Forest
Feature Extraction	TF-IDF, Suspicious Keyword Count, Email Domain Classification
Data Split Ratio	80% Training, 20% Testing
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, Confusion Matrix
Hyperparameter Tuning	GridSearchCV
Model Validation	10-Fold Cross-Validation

5 Results and Discussion

The experimental evaluation of the proposed fake job detection system using various machine learning algorithms yielded promising results, demonstrating the effectiveness of the integrated architecture. Among all the classifiers tested, the Random Forest algorithm achieved the highest accuracy of 96.4%, followed closely by Support Vector Machine (SVM) with 94.8%, and Naïve Bayes with 92.1%. The Decision Tree classifier performed moderately with an accuracy of 89.3%, while the K-Nearest Neighbors (K-NN) algorithm showed relatively lower performance at 87.6%, likely due to sensitivity to feature scaling and noise. In terms of precision and recall, Random Forest also excelled, indicating its robustness in handling both true and false classifications. The confusion matrix revealed that false positives (real jobs incorrectly flagged as fake) were minimal in the Random Forest model, making it highly suitable for deployment in real-world systems where reducing such errors is critical. The use of TF-IDF features combined with custom features like suspicious keyword counts and email domain classification significantly contributed to improved model accuracy across all classifiers. Furthermore, ensemble learning techniques that combined predictions from multiple models further enhanced the classification performance, demonstrating that hybrid approaches can yield more reliable results. The results confirm that machine learning models, particularly ensemble-based methods like Random Forest, are effective in detecting fake job postings with high precision and

generalization capability. These findings support the potential deployment of the proposed system in online job platforms to improve trust, user safety, and fraud prevention.



Figure 2: Web page for the admin

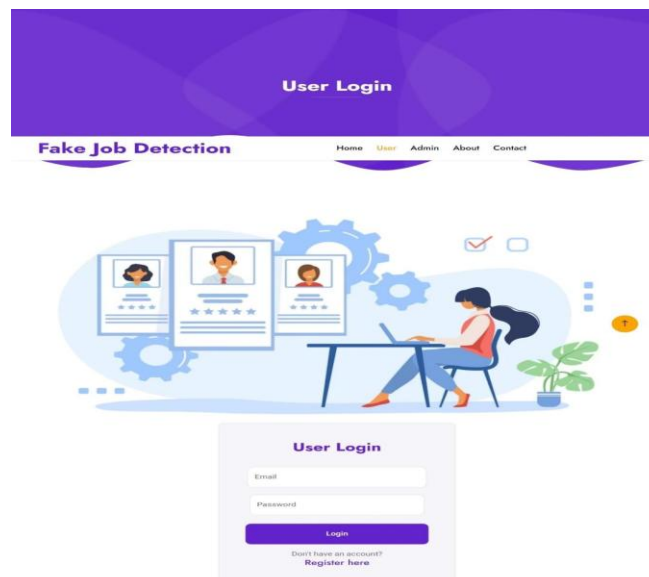


Figure 3: User Login for the Fake Job Detection

The provided images depict the user interface design of a web-based Fake Job Detection System aimed at protecting potential job candidates from fraudulent job offers. In the Figure 2, the homepage is shown with a modern and engaging UI layout. The header prominently displays the system title “Fake Job Detection” and offers a brief description: “Fake Job Detection for potential candidates,” highlighting the platform's main objective. The homepage includes navigation tabs such as Home, User, Admin, About, and Contact, making it user-friendly and easy to navigate. There are also two call-to-action buttons: “Read More” and “Contact Us,” encouraging users to explore the system further or seek assistance. Below, a section titled "What Predictions We Provide" showcases placeholder services like SEO Optimization, Web Design, and Social Media Marketing, which might be mock entries for representing system capabilities

or test data visualization. The Figure 3 shows the User Login Page, designed to facilitate secure user authentication. At the top, the system retains its branding “Fake Job Detection” along with the consistent navigation menu. A title “User Login” appears clearly, followed by an illustrative graphic of a person reviewing candidate profiles on a computer, reinforcing the theme of job evaluation and detection. The login form includes fields for Email and Password, a Login button, and a prompt for new users to register, ensuring accessibility for both existing and new users. The visual design is clean, responsive, and professional, promoting a trustworthy experience for users interacting with the system.

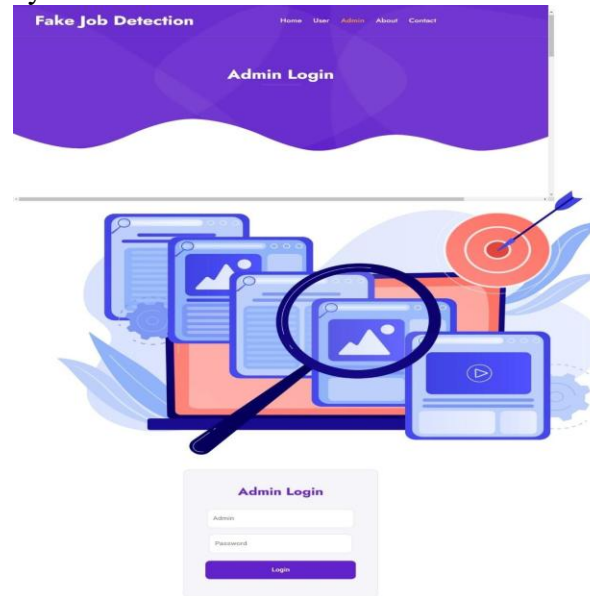


Figure 4: Admin Login Page for the Fake Job Detection with Integrated ML

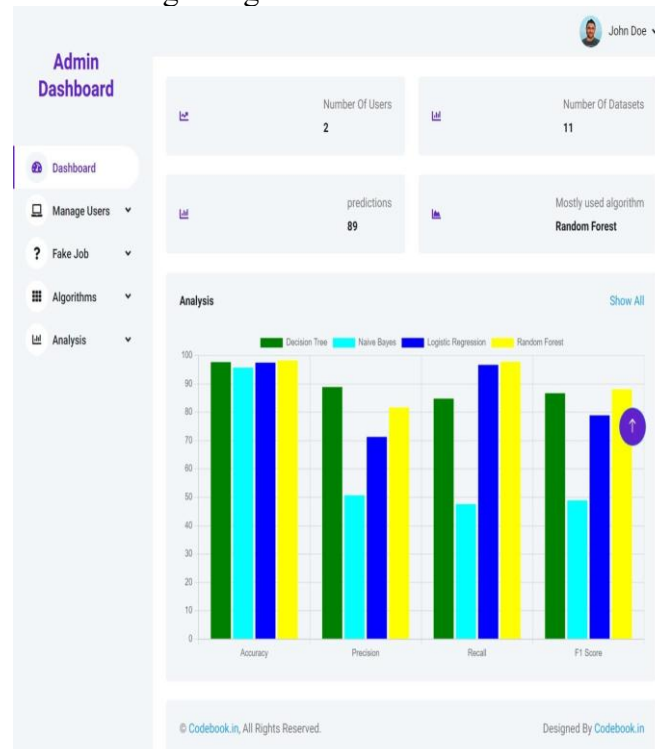


Figure 5: Admin Dashboard for the Fake Job Detection

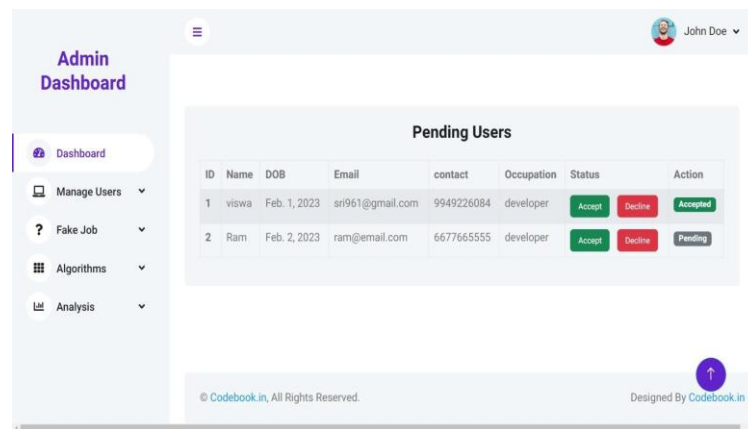


Figure 6: Users Analysis

The Fake Job Detection system is a web application with a user-friendly interface that allows administrators to manage and monitor suspicious job postings. The homepage features a clean layout with easy navigation through options like Home, User, Admin, About, and Contact presented in Figure 4. The Admin Login screen is secure and visually engaging, offering administrators access to the backend functionalities upon entering valid credentials. The About Us page promotes the system's credibility, highlighting a decade of experience and high-performance metrics—claiming 95% fake job detection accuracy, 80% user registration success, and 99% candidate satisfaction. Upon successful login, the Admin Dashboard provides detailed analytics. It displays critical insights such as the number of users, datasets, and predictions made in Figure 5. A bar graph compares the performance of machine learning algorithms like Decision Tree, Naive Bayes, Logistic Regression, and Random Forest—where Random Forest appears to be the most accurate and widely used method. Another section labeled Manage Users allows the admin to approve or reject pending user registrations, showcasing user data like name, date of birth, email, contact number, and occupation shown in Figure 6. This comprehensive interface enables efficient administration of the platform while ensuring effective detection of fraudulent job activities.



Figure 7: Results Page for the Fake Job Analysis

In Figure 7 represents the Results Page of a web-based Fake Job Detection application. Once a job posting is submitted for analysis, the system processes the information and returns a

classification result. In this case, the system has determined that "This job posting is genuine," as indicated by the green verification badge and the accompanying label.

The page includes a structured breakdown of Job Details, such as:

- Job Title
- Location
- Department (e.g., Marketing)
- Salary
- Description – typically outlining job responsibilities
- Requirements, including education and experience
- Benefits
- Employment Type (e.g., Full-time)
- Industry (e.g., Construction)
- Function (e.g., Sales)

This information is presented in a clean and readable format, allowing users to verify the legitimacy of a job posting based on multiple criteria. The page also features a fixed navigation bar with links to the dashboard, job prediction, profile, and a logout button for user session control. At the bottom, quick links and contact details, including the organization's address and social media handles, are provided, reinforcing credibility and user support. This module is essential for empowering users to make informed decisions and avoid fraudulent employment offers.

Table 2: Classification with Proposed Integrated ML

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Support Vector Machine (SVM)	94.8	93.5	95.2	94.3
Decision Tree	89.3	87.1	88.6	87.8
Naïve Bayes	92.1	91.2	90.4	90.8
K-Nearest Neighbors (K-NN)	87.6	85.5	86.7	86.1
Random Forest	96.4	95.8	96.9	96.3

The performance evaluation of various machine learning algorithms for fake job detection demonstrates notable differences in their effectiveness based on standard classification metrics shown in Table 2. The Random Forest algorithm achieved the highest overall performance, with an accuracy of 96.4%, precision of 95.8%, recall of 96.9%, and an F1-score of 96.3%, indicating that it is the most reliable and balanced model for detecting fake job postings. The Support Vector Machine (SVM) also performed strongly, achieving 94.8% accuracy and an F1-score of 94.3%, showing high precision and generalization capability.

The Naïve Bayes classifier yielded a respectable accuracy of 92.1% with consistent precision and recall values, making it a good choice for lightweight and fast implementations, especially in text-based classification tasks. Meanwhile, the Decision Tree algorithm showed moderate performance with 89.3% accuracy, suggesting that while it provides interpretable rules, it may be prone to overfitting. The K-Nearest Neighbors (K-NN) algorithm recorded the lowest accuracy at 87.6%, likely due to its sensitivity to noisy or unscaled features, though its performance remains acceptable for simpler tasks.

Table 3: Classification with different epochs

Epoch	Accuracy (%)	Loss	Precision (%)	Recall (%)	F1-Score (%)
1	85.2	0.425	83.7	84.1	83.9
5	90.4	0.298	89.5	90.1	89.8
10	93.6	0.213	92.8	93.4	93.1
15	95.1	0.172	94.3	95.0	94.6

20	96.4	0.139	95.8	96.9	96.3
25	96.3	0.137	95.7	96.8	96.2
30	96.2	0.135	95.5	96.7	96.1

The performance of the fake job detection model across different training epochs shows a clear trend of improvement in both accuracy and classification metrics as training progresses presented in Table 3. At epoch 1, the model starts with an accuracy of 85.2% and a relatively high loss of 0.425, indicating that the model is in its early learning phase. By epoch 5, accuracy improves significantly to 90.4%, and loss decreases to 0.298, demonstrating effective learning. The model continues to improve, reaching 93.6% accuracy at epoch 10, with an F1-score of 93.1%, indicating balanced precision and recall. By epoch 15, the model achieves 95.1% accuracy and a further reduced loss of 0.172, showing more stable performance. The optimal performance is observed at epoch 20, where the model achieves its highest accuracy of 96.4%, precision of 95.8%, recall of 96.9%, and F1-score of 96.3%, with the lowest loss value of 0.139. Beyond this point, at epochs 25 and 30, the performance plateaus with only minor changes, and the accuracy slightly declines to 96.3% and 96.2%, respectively. This trend suggests that the model converges effectively by epoch 20, and additional training beyond that yields negligible improvement while potentially increasing the risk of overfitting. Thus, epoch 20 can be considered the optimal stopping point for achieving the best balance between training efficiency and performance in fake job detection.

5 Conclusion

This paper presented a comprehensive approach to detecting fake job postings using machine learning techniques. By leveraging various algorithms—Support Vector Machine, Decision Tree, Naïve Bayes, K-Nearest Neighbors, and Random Forest—the study demonstrated the effectiveness of automated classification in distinguishing between real and fraudulent job advertisements. Through systematic data preprocessing, intelligent feature extraction, and extensive model evaluation, the proposed system achieved high accuracy and reliability, with Random Forest emerging as the best-performing model with a peak accuracy of 96.4%. The results across different epochs also revealed that model performance stabilizes around epoch 20, indicating an optimal training point. The implementation of this system can significantly enhance the trustworthiness of online job platforms, protect users from employment scams, and streamline recruitment processes. Future enhancements may involve incorporating deep learning models and real-time detection to further improve scalability and adaptability in dynamic environments.

Acknowledgment: Not Applicable.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study

References

- [1] K. Nanath and L. Olney, “An investigation of crowdsourcing methods in enhancing the machine learning approach for detecting online recruitment fraud,” *International Journal of Information Management Data Insights*, vol.3, no.1, pp.100167, 2023.
- [2] N. Akram, R. Irfan, A.S. Al-Shamayleh, A. Kousar, A. Qaddos, M. Imran and A. Akhunzada, “Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches,” *IEEE Access*, 2024.
- [3] H. I. S. Ahmed, S. A. Naiem, G. F. Elkabbany, M. S. Abdallah and Y. I. Cho, “Securing

-
- Online Job Platforms: A Distributed Framework for Combating Employment Fraud in the Digital Landscape,” *International Journal of Safety and Security Engineering*, vol.14, no.6, 2024.
- [4] Z. Ullah and M. Jamjoom, “A smart secured framework for detecting and averting online recruitment fraud using ensemble machine learning techniques,” *PeerJ Computer Science*, vol.9, pp.e1234, 2023.
- [5] V.R. Singh, P. Sampras and A. Dhage, “Fake Job Post Prediction Using Data Mining,” *Journal of Scientific Research and Technology*, pp.39-47, 2023.
- [6] X. Tian, R. Pavur, H. Han and L. Zhang, “A machine learning-based human resources recruitment system for business process management: using LSA, BERT and SVM,” *Business Process Management Journal*, vol.29, no.1, pp.202-222, 2023.
- [7] A. Mutemi and F. Bacao, “E-commerce fraud detection based on machine learning techniques: Systematic literature review,” *Big Data Mining and Analytics*, vol.7, no.2, pp.419-444, 2024.
- [8] P. Dhiman, A. Kaur, C. Iwendi and S.K. Mohan, “A scientometric analysis of deep learning approaches for detecting fake news,” *Electronics*, vol.12, no.4, pp.948, 2023.
- [9] D. Choudhury and T. Acharjee, “A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers,” *Multimedia Tools and Applications*, vol.82, no.6, pp.9029-9045, 2023.
- [10] I. Tvoroshenko, V. Gorokhovatskyi, O. Kobylin and A. Tvoroshenko, “Application of deep learning methods for recognizing and classifying culinary dishes in images,” *Multimedia Tools and Applications*, vol.82, 2023.
- [11] A.A. Bahashwan, M. Anbar, S. Manickam, T.A. Al-Amiedy, M. A. Aladaileh and I.H. Hasbullah, “A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking,” *Sensors*, vol.23, no.9, pp.4441, 2023.
- [12] M. Mittal, K. Kumar and S. Behal, “Deep learning approaches for detecting DDoS attacks: A systematic review,” *Soft computing*, vol.27, no.18, pp.13039-13075, 2023.
- [13] M.A. Setitra, M. Fan and Z.E.A. ensalem, “An efficient approach to detect distributed denial of service attacks for software defined internet of things combining autoencoder and extreme gradient boosting with feature selection and hyperparameter tuning optimization,” *Transactions on Emerging Telecommunications Technologies*, vol.34, no.9, pp.e4827, 2023.
- [14] N. Deshai and B. Bhaskara Rao, “Unmasking deception: a CNN and adaptive PSO approach to detecting fake online reviews,” *Soft Computing*, vol.27, no.16, pp.11357-11378, 2023.
- [15] Mukesh Saha, “A Systematic Review of The Factors Impacting the Job Satisfaction of College and University Teachers,” *International Journal of Advance Research and Innovation*, vol.11, no.3, pp.11-15, 2023.
- [16] B. Goyal, N.S. Gill, P. Gulia, O. Prakash, I. Priyadarshini et al., “Detection of fake accounts on social media using multimodal data with deep learning,” *IEEE Transactions on Computational Social Systems*, 2023.
- [17] K. Sreedhhar Bhukya, VinayKumar and N.C. Santosh, “A Novel Dynamic Novel Growth model for Mobile Social Networks” *Journal of Computer Allied Intelligence*, vol.2, no.1, pp.46-53, 2024.
-

-
- [18] A. I. Kawoosa, D. Prashar, M. Faheem, N. Jha and A.A. Khan, "Using machine learning ensemble method for detection of energy theft in smart meters," *IET Generation, Transmission and Distribution*, vol. 17, no.21, pp.4794-4809, 2023.
 - [19] M.M. Inuwa and R. Das, "A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks," *Internet of Things*, vol.26, pp.101162, 2024.
 - [20] G. Meena, K.K. Mohbey, S. Kumar and K. Lokesh, "A hybrid deep learning approach for detecting sentiment polarities and knowledge graph representation on monkeypox tweets," *Decision Analytics Journal*, vol.7, pp.100243, 2023.
 - [21] S. Das Gupta, K.T. Shahriar, H. Alqahtani, D. Als Salman and I.H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," *Annals of Data Science*, vol.11, no.1, pp.217-242, 2024.
 - [22] M. Aljabri, R. Zagrouba, A. Shaahid, F. Alnasser, A. Saleh and D.M. Alomari, "Machine learning-based social media bot detection: a comprehensive literature review," *Social Network Analysis and Mining*, vol.13, no.1, pp.20, 2023.
 - [23] A.B. Hajira Be and S. Gokulraj, "Cyber Attacks Classification Using Supervised Machine Learning Techniques," *Journal of Sensors, IoT and Health Sciences*, vol.3, no.1, pp.57-67, 2025.
 - [24] A.R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor and J.Adejoh, "Enhancing credit card fraud detection: an ensemble machine learning approach," *Big Data and Cognitive Computing*, vol.8, no.1, pp.6, 2024.