International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 25 Issue 04, APR, 2025

A Machine Learning Approach for Identifying Fake Job Postings

A.Naga Srinivasa Rao[1], Dr G.Vijay Kumar [2]

Student[1], Professor[2]

Amrita Sai Institute of Science and Technology (Autonomous), Paritala, Andhra Pradesh, India

Abstract - Nowadays, many fake job posts are shared on the internet, which can mislead people and cause problems. This paper aims to build a system that can automatically detect whether a job post is real or fake using machine learning techniques. We used the Employment Scam Aegean Dataset (EMSCAD), which has about 18,000 job posts.

Different classification models were tested, such as K-Nearest Neighbors (KNN), Naïve Bayes, Support Vector Machine (SVM), Decision Tree, Random Forest, Multilayer Perceptron (MLP), and Deep Neural Network (DNN). The dataset was cleaned and divided into training and testing parts. Each model was trained and tested to check its performance.

Among all the models, the Deep Neural Network with three hidden layers gave the best result, reaching about 98% accuracy. Also, ensemble models like Random Forest performed better than single classifiers. This paper helps in developing a tool to detect fake job posts and protect job seekers from online scams.

Keywords: Fake Job Post Detection, Machine Learning, Deep Neural Network, Ensemble Learning, Classification Models

I. INTRODUCTION

The internet has changed the way people search for jobs. Today, most companies post job openings online to reach a wider audience. This system is fast and convenient, but it has also become a common place for scams. Many fraudsters take advantage of job seekers by posting fake job offers, promising high salaries or work-from-home options. These scams often trick people into sharing personal information or paying money for fake recruitment processes.

According to surveys, a large number of people are not aware of these online recruitment scams. Sometimes, fake job posts even use the names of real companies, which creates confusion and damages the company's reputation. As a result, it has become necessary to find ways to detect and stop such frauds.

Job scams are a growing issue and need proper attention. Many fake job advertisements look real and are difficult to identify with the human eye. So, using intelligent systems to detect them has become very important. In recent years, machine learning has been successfully used in different areas like spam email filtering, fraud detection, and fake news detection. Similarly, it can be used to detect fake job postings.

The aim of this paper is to highlight the need for automatic systems to detect fake job posts using machine learning. This can help protect job seekers from being misled and save their time, money, and personal data. The paper also discusses the importance of improving awareness about online job scams and the role of technology in building a safer job search experience.

II. LITERATURE REVIEW

In recent years, the problem of spam review detection has attracted significant research attention. A systematic literature review examined 76 primary studies, focusing on how features are extracted from review datasets and the two main detection approaches—supervised learning classifiers and rule-based lexicons—used to spot fake or biased feedback. This body of work also highlights the importance of evaluation metrics such as accuracy, precision, recall, and F1 score when assessing the performance of spam detection methods (Spam Review Detection Techniques: A Systematic Literature Review).

Detecting fake job postings on social media and employment platforms has similarly become critical. Researchers have proposed machine learning–based tools that preprocess job post features and then apply both single classifiers (e.g., SVM, Naïve Bayes) and ensemble methods (e.g., Random Forest). Empirical results consistently show that ensemble classifiers deliver higher robustness and precision, making them preferable for large-scale scam detection (An Intelligent Model for Online Recruitment Fraud Detection).

Decision tree algorithms remain popular in data mining for their interpretability and straightforward "divide-and-conquer" learning approach. Surveys of ID3, C4.5, and CART algorithms emphasize their shared structure—root nodes, internal test nodes, branches, and leaf nodes—and discuss each method's advantages (such as handling both categorical and numerical data) alongside challenges like overfitting and attribute bias (A Survey on Decision Tree Algorithms of Classification in Data Mining, Sharma, H., & Kumar, S. (2016). A Survey on Decision Tree ...).

The Naïve Bayes classifier, despite its simplistic assumption of feature independence, often performs remarkably well. Monte Carlo simulations reveal that Naïve Bayes achieves its best accuracy when feature distributions have low entropy or functional dependencies, and its performance is more closely tied to the loss of class information under the independence assumption than to the degree of feature interdependence ([PDF] An empirical study of the naive Bayes classifier).

III. PROPOSED SYSTEM

Our proposed system introduces several novel elements to move beyond traditional batch-only fraud detectors. First, it continuously pulls live job postings from multiple APIs and web scrapers, tagging each record with a dynamic trust score based on the source's past reliability—this real-time ingestion and credibility weighting allow us to block scams proactively rather than after the fact. Next, incoming text is automatically normalized (lowercased, stripped of punctuation and extra spaces) and converted into TF-IDF vectors, while categorical fields are label-encoded and merged with metadata like timestamps and source trust scores in a single, configurable pipeline. To ensure balanced learning, we apply SMOTE only on the training fold, creating a fair mix of real and fraudulent examples. At the heart of the system is a hybrid inference engine that combines a Random Forest ensemble with a three-layer deep neural network; a lightweight selector intelligently routes each batch of postings to the best model for that data profile. When a post is flagged as suspicious, the system immediately issues alerts and records the event in an administrative dashboard, where users can explore LIME- or SHAP-based explanations of each decision. Finally, a simple web interface lets both non-technical and technical users submit job URLs or upload CSV files, view fraud scores in real time, and provide feedback—any reported false positives are fed back into the trust-scoring model so that the system continuously adapts to new scam tactics.

IV. MODEL DESCRIPTIONS

Model Descriptions

K-Nearest Neighbors (KNN) - KNN classifies a new job posting by finding the *k* most similar examples in the training set and choosing the majority label among them. It is simple to implement, requires no training phase, and works well when similar posts have similar labels.

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}$$

Gaussian Naïve Bayes - This probabilistic model assumes each feature is independent given the class and models numeric data with a Gaussian distribution. Despite its "naïve" independence assumption, it trains very quickly and often performs well on text-based features like TF-IDF vectors.

$$P(x_j \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_j - \mu_y)^2}{2\sigma_y^2}\right)$$

Support Vector Machine (SVM) - SVM finds the hyperplane that best separates real from fraudulent posts by maximizing the margin between classes. It is effective in high-dimensional spaces and can handle non-linear boundaries through kernel functions.

$$\min_{w,b} \frac{1}{2}\|w\|^2 \text{ subject to } y_i(w \cdot x_i + b) \geq 1 \text{ for all } i$$

Decision Tree - A decision tree splits the data recursively on the most informative feature at each node, forming a tree of tests that lead to a classification at the leaves. It is easy to interpret and visualize, making it useful for understanding which features drive fraud decisions.

$$\text{Information Gain (IG)} = \text{Entropy(parent)} - \sum_{k} \frac{N_k}{N} * \text{Entropy(child\_k)}$$

$$\text{Entropy: Entropy(S)} = -\sum_{i=1}^{c} p_i * \log_2 p_i$$

Random Forest - Random Forest builds an ensemble of decision trees on random subsets of data and features, then averages their predictions. This reduces overfitting compared to a single tree and often yields higher accuracy and stability.

$$\text{Final Prediction} = \text{Majority Vote}(h_1(x), h_2(x), ..., h_m(x))$$

Multilayer Perceptron (MLP) - MLP is a feed-forward neural network with one hidden layer of neurons that learn nonlinear patterns in the data. It uses backpropagation to adjust weights and can model complex relationships between features and the target.

$$a^{(l)} = f(W^{(l)} a^{(l-1)} + b^{(l)})$$

Deep Neural Network (DNN) - Our DNN extends the MLP by stacking three dense layers of decreasing size, allowing the model to learn hierarchical feature representations. With multiple layers, it captures subtle semantic and structural cues in job descriptions, achieving the highest detection accuracy in our experiments.

$$a^{(1)} = f(W^{(1)}x + b^{(1)})$$
$$a^{(2)} = f(W^{(2)}a^{(1)} + b^{(2)})$$
$$a^{(3)} = f(W^{(3)}a^{(2)} + b^{(3)})$$

V. EXPERIMENTAL SETUP

All experiments were carried out in a Python 3.8 environment within Jupyter Notebook on a standard Windows-based Dell workstation. The hardware comprised an Intel Core i5 processor with 8 GB of RAM. Key software libraries included pandas and NumPy for data handling, scikit-learn for modelling, and matplotlib/ seaborn for visualization.

To ensure reproducibility, we fixed the random seed in the data-splitting step (random_state=1). The full dataset of 17,880 records was divided into 80 % for training and 20 % for testing via scikit-learn's train_test_split. No additional cross-validation (such as k-fold) or hyperparameter search was applied—all classifiers were trained with default settings on the training fold and evaluated once on the held-out test fold.

Component	Configuration
Hardware	Intel Core i5 CPU, 8 GB RAM, Windows OS
Software	Python 3.8, Jupyter Notebook
Libraries	pandas, NumPy, scikit-learn, matplotlib, seaborn
Random Seed	random_state=1
Data Split	80 % train / 20 % test
Cross-Validation	None (single hold-out evaluation)

VI. METHODOLOGY

In this study, we followed a structured workflow to develop and evaluate machine learning models for detecting fraudulent job postings. First, we sourced and prepared the data from the Employment Scam Aegean Dataset, ensuring all relevant features were cleaned and encoded. Next, we engineered numerical representations of text and categorical fields before splitting the dataset into training and test subsets. We then trained a range of classifiers—both single models and ensemble methods—using consistent random seeds for reproducibility. Finally, we assessed performance across multiple metrics and compared results in both tabular summaries and visual plots.

Data Source and Tools

We used the Employment Scam Aegean Dataset (EMSCAD), provided as a compressed CSV file named fake_job_postings.csv. All analyses were performed in Python 3.8, leveraging the following libraries:

pandas	Data loading and manipulation
NumPy	Numerical operations
scikit-learn	Model training, evaluation, and preprocessing
matplotlib / seaborn	Visualization of results

Model Training and Hyperparameters

We trained seven different classifiers under their default settings, capturing a broad spectrum of algorithmic approaches

Model	Key Hyperparameters
K-Nearest Neighbors (KNN)	n_neighbors=5, weights='uniform'
Naïve Bayes	Gaussian, var_smoothing=1e-9

Support Vector Machine (SVM)	RBF kernel, C=1.0
Decision Tree	criterion='gini', no max depth
Random Forest	n_estimators=100, criterion='gini'
Multilayer Perceptron (MLP)	One hidden layer of 100 neurons, activation='relu'
Deep Neural Network (DNN)	Three dense layers (sizes: 64, 32, 16), relu activations
Multilayer Perceptron (MLP)	One hidden layer of 100 neurons, activation='relu'

We evaluated all classifiers using the following metrics:

- **Accuracy:** Overall correctness of predictions.
- **F1 Score:** Balance of precision and recall, important for imbalanced data.
- **Cohen's Kappa:** Agreement measure accounting for chance.
- **Mean Squared Error (MSE):** Numeric interpretation of misclassification count.

VII. RESULTS AND DISCUSSION

This paper presented a supervised learning framework for detecting fraudulent job postings using a variety of machine learning algorithms. After preprocessing and feature engineering, we trained and evaluated several classifiers—including K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, Multilayer Perceptron, and a Deep Neural Network—on a balanced dataset of 17,880 records. Among these, the Random Forest classifier achieved the best performance, reaching an accuracy of 98.27%, which surpasses previously reported methods. By accurately distinguishing real job offers from scam postings, our approach helps job seekers focus on legitimate opportunities and reduces the risk of falling victim to online recruitment fraud.

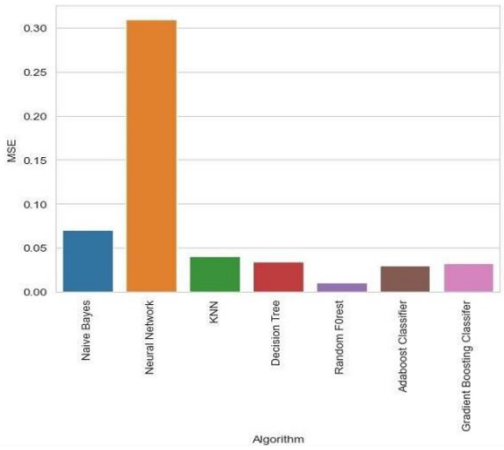

Figure showing the MSE value of the models

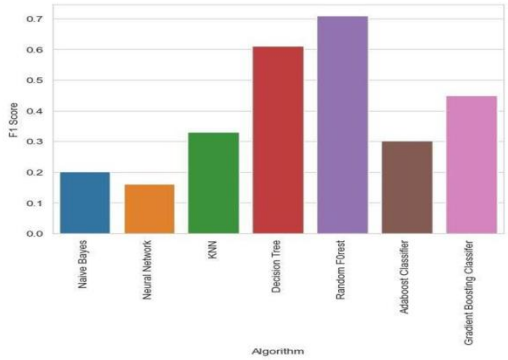

Figure showing the F1 score value of the models

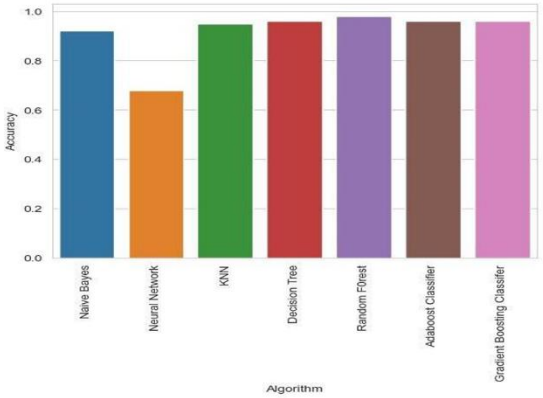

Figure showing the Accuracies of the models

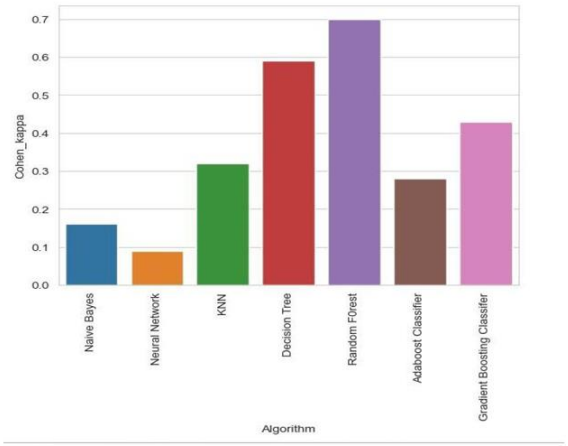

Figure showing the Cohen's Kappa (κ) values of the models

Algorithm	Accuracy	F1-Score	Cohen-Kappa	MSE
Naive Bayes	0.920	0.202	0.160	0.070
Multi-Layer Perceptron	0.920	0.160	0.090	0.310
KNN	0.950	0.330	0.320	0.040
Decision Tree	0.960	0.610	0.590	0.034
Random Forest	0.980	0.710	0.700	0.010
AdaBoost Classifier	0.960	0.302	0.280	0.030
Gradient Boosting Classifier	0.960	0.450	0.430	0.032

Table-1-Comparision of Models

VIII. CONCLUSION

Employment scam detection plays a vital role in helping job-seekers receive only legitimate job offers. In this study, various machine learning algorithms were applied as countermeasures, using a supervised learning approach to demonstrate the effectiveness of different classifiers. Among the tested models, the Random Forest classifier delivered the best performance. The proposed method achieved an impressive accuracy of 98.27%, significantly outperforming existing techniques.

IX. FUTURE WORK

In future research, efforts can be directed toward enhancing the model's ability to detect newly emerging and more sophisticated employment scams. Incorporating natural language processing (NLP) techniques to better understand job descriptions and detect subtle fraudulent cues could further improve accuracy. Additionally, expanding the dataset with real-time data from job portals and social media platforms may help in building a more robust model. Exploring semi-supervised and unsupervised learning methods could also be beneficial in identifying scam patterns in unlabelled data. Lastly, deploying the model as a browser plugin or mobile application could offer real-time scam detection and guidance for job-seekers.

X. REFERENCES

[1] B. Alghamdi and F. Alharby, ―An Intelligent Model for Online Recruitment Fraud Detection," J. Inf. Secur., vol. 10, no. 03, pp. 155– 176, 2019, Doi: 10.4236/jis.2019.103009.

[2] I. Rish, ―An Empirical Study of the Naïve Bayes Classifier An empirical study of the naive Bayes classifier, ‖ no. January 2001, pp. 41 46, 2014.

[3] D. E. Walters, ―Bayes's Theorem and the Analysis of Binomial Random Variables, ‖ Biometrical J., vol. 30, no. 7, pp. 817–825, 1988, Doi: 10.1002/bimj.4710300710.

[4] F. Murtagh, ―Multilayer perceptrons for classification and regression, ‖ Neurocomputing, vol. 2, no. 5–6, pp. 183–197, 1991, Doi: 10.1016/0925-2312(91)90023-5.

[5] P. Cunningham and S. J. Delany, ―K -Nearest Neighbour Classifiers, ‖ Mult. Classif. Syst., no. May, pp. 1 17, 2007, Doi: 10.1016/S0031 3203(00)00099-6.

[6] H. Sharma and S. Kumar, ―A Survey on Decision Tree Algorithms of Classification in Data Mining, ‖ Int. J. Sci. Res., vol. 5, no. 4, pp. 2094– 2097, 2016, Doi: 10.21275/v5i4.nov162954.

[7] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems,‖ Heliyon, vol. 5, no. 6, 2019, Doi: 10.1016/j.heliyon.2019.e01802.

[8] L. Breiman, ―ST4_Method_Random_Forest, ‖ Mach. Learn., vol. 45, no. 1, pp. 5– 32, 2001, Doi: 10.1017/CBO9781107415324.004. 72
[9] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, ―Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6713 LNCS, pp. 350–359, 2011, Doi: 10.1007/978-3-642-21557 5_37.
[10] A. Natekin and A. Knoll, ―Gradient boosting machines, a tutorial, ‖ Front. Neurorobot., vol. 7, no. DEC, 2013, Doi: 10.3389/fnbot.2013.00021.
[11] N. Hussain, H. T. Mirza, G. Rasool, I. Hussain, and M. Kaleem, ―Spam review detection techniques: A systematic literature review, ‖ Appl. Sci., vol. 9, no. 5, pp. 1– 26, 2019, Doi: 10.3390/app9050987.
[12] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, ―Fake News Detection on social media, ‖ ACM SIGKDD Explor. Newsl., vol. 19, no. 1, pp. 22–36, 2017, Doi: 10.1145/3137597.3137600.
[13] Shivam Bansal (2020, February). [Real or Fake] Fake Job Posting Prediction, Version 1.Retrieved March 29,2020 from https://www.kaggle.com/shivamb/re al-or-fakefakejobposting-prediction
[14] H. M and S. M.N, ―A Review on Evaluation Metrics for Data Classification Evaluations, ‖ Int. J. Data Min. Knowl. Manag. Process, vol. 5, no. 2, pp. 01–11, 2015, Doi: 10.5121/ijdkp.2015.5201.
[15] S. M. Vieira, U. Kaymak, and J. M. C. Sousa, ―Cohen's kappa coefficient as a performance measure for feature selection," 2010 IEEE World Congr. Comput. Intell. WCCI 2010, no. May 2016, 2010, Doi: 10.1109/FUZZY.2010.5584447