# Fake Job Recruitment Detection Using Machine Learning Approach

Swapnil[#1], Pranav Parth [#2], Rishu Raj [#3], Ritik Saini [#4]

[#1]*Student, Sapthagiri College of Engineering, Bengaluru, Karnataka-560057*

*Abstract* — **Employment scams are on the rise. According to CNBC, the number of employment scams doubled in 2018 as compared to 2017. The current market situation has led to high unemployment. Economic stress and the impact of the coronavirus have significantly reduced job availability and the loss of jobs for many individuals. A case like this presents an appropriate opportunity for scammers. Many people are falling prey to these scammers using the desperation that is caused by an unprecedented incident. Most scammer do this to get personal information from the person they are scamming. Personal information can contain address, bank account details, social security number etc. I am a university student, and I have received several such scam emails. The scammers provide users with a very lucrative job opportunity and later ask for money in return. Or they require investment from the job seeker with the promise of a job. This is a dangerous problem that can be addressed through Machine Learning techniques and Natural Language Processing (NLP).**

## I. INTRODUCTION

Employment fraud is one of the most serious problems recently raised in the field of fake recruiting. Nowadays, many companies prefer to post vacancies online for easy and timely access to job seekers. However, this intent could be a sort of scam by scammers, as scammers provide employment to job seekers by robbing them of money. Fraudulent classified ads are unreliable and can be placed against reputable companies. This fraudulent job detection is focused on getting automated tools to identify fake jobs and report to people not to apply for such jobs. To this end, a machine learning approach is applied that uses multiple classification algorithms to detect fake posts. In this case, the classification tool separates fake job ads from a larger set of jobs and alerts the user. Supervised learning algorithms are first considered as a classification method to address the problem of detecting fraudulent job listings. The classifier considers the training data and assigns the input variables to the target class. Let's take a brief look at the classifier used in this paper to identify fake classified ads from the rest. This classifier-based forecast can be broadly categorized into a single classifier-based forecast and an ensemble classifier-based forecast.

### A. Single Classifier based Prediction-

Classifiers are trained for predicting the unknown test cases. The following classifiers are used while detecting fake job posts-

#### a) Naive Bayes Classifier-

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these

features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

• P(h): the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h.

• P(D): the probability of the data (regardless of the hypothesis). This is known as the prior probability.

• P(h|D): the probability of hypothesis h given the data D. This is known as posterior probability.

• P(D|h): the probability of data d given that the hypothesis h was true. This is known as posterior probability.

**b)SGD Classfier-**

**Stochastic Gradient Descent (SGD)** is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features.

The class SGD Classifier implements a plain stochastic gradient descent learning routine which supports different loss functions and penalties for classification. Below is the decision boundary of a SGD Classifier trained with the hinge loss, equivalent to a linear SVM.

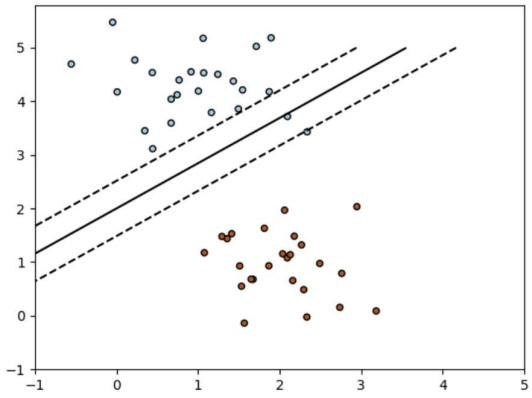


Figure1: SGD Classifier Working

**c) Natural Language Processing-**

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment.

## II. Methods and Methodology

This project uses data provided from Kaggle. This data contains features that define a job posting. These job postings are categorized as either real or fake. Fake job postings are a very small fraction of this dataset. That is as excepted. We do not expect a lot of fake jobs postings. This project follows five stages. The five stages adopted for this project are –

1. Problem Definition (Project Overview, Project statement and Metrics)
2. Data Collection
3. Data cleaning, exploring and pre-processing
4. Modeling
5. Evaluating

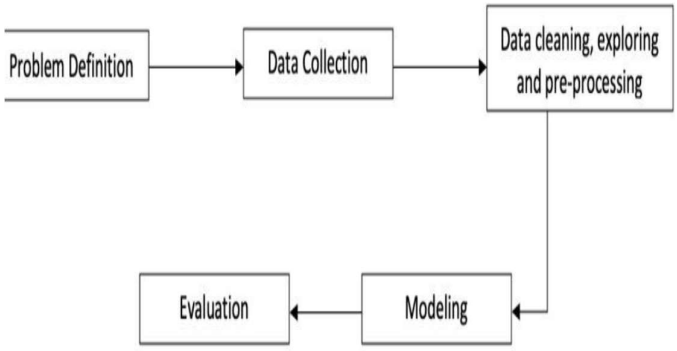


Figure2: Methodology/ WorkFlow

*A.* Algorithms And Techniques

Based on the initial analysis, it is evident that both text and numeric data is to be used for final modeling. Before data modeling a final dataset is determined. This project will use a dataset with these features for the final analysis:

1. telecommuting
2. fraudulent
3. ratio: fake to real job ratio based on location
4. text: combination of title, location, company_profile, description, requirements, benefits, required_experience, required_education, industry and function
5. character_count: Count of words in the textual data Word count histogram

Further pre-processing is required before textual data is used for any data modeling.

The algorithms and techniques used in project are:

1. Natural Language Processing
2. Naïve Bayes Algorithm
3. SGD Classifier

Naïve bayes and SGD Classifier are compared on accuracy and F1-scores and a final model is chosen. Naïve Bayes is the baseline model, and it is used because it can compute the conditional probabilities of occurrence of two events based on the probabilities of occurrence of each individual event, encoding those probabilities is extremely useful.

A comparative model, SGD Classifier is used since it implements a plain stochastic gradient descent learning routine which supports different loss functions and penalties for classification. This classifier will need high penalties when classified incorrectly. These models are used on both the text and numeric data separately and the final results are combined.

*B.* Methodology

The following steps are taken for text processing:

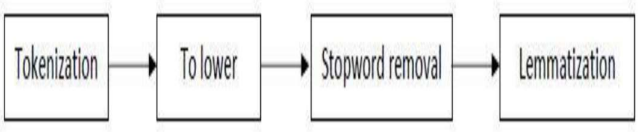


Figure3: Pre-Processing of data set

• Tokenization: The textual data is split into smaller units. In this case the data is split into words.
• To Lower: The split words are converted to lowercase
• Stopword removal: Stopwords are words that do not add much meaning to sentences. For example: the, a, an, he, have etc. These words are removed.
• Lemmatization: The process of lemmatization groups in which inflected forms of words are used together.

*C.* Implementation

A diagrammatic representation of the implementation of this project is given below. The dataset is split into text,numeric and y-variable. The text dataset is converted into a term-frequency matrix for further analysis. Then using sci-kit learn, the datasets are split into test and train datasets. The baseline model Naïve bayes and another model SGD is trained on the using the train set which is 70% of the dataset. The final outcome of the models based on two test sets – numeric and text are combined such that if both models say that a particular data point is not fraudulent only then a job posting is fraudulent. This is done to reduce the bias of Machine Learning algorithms towards majority classes. The trained model is used on the test set to evaluate model performance. The Accuracy and F1-score of the two models – Naïve bayes and SGD are compared and the final model for our analysis is selected.

*D.* Refinement

The independent variables have been tweaked in various capacities to improve the results of model. This has been done by adding and removing features. Also, different penalties are used to evaluate the final model. However, the difference in outcomes were very insignificant.

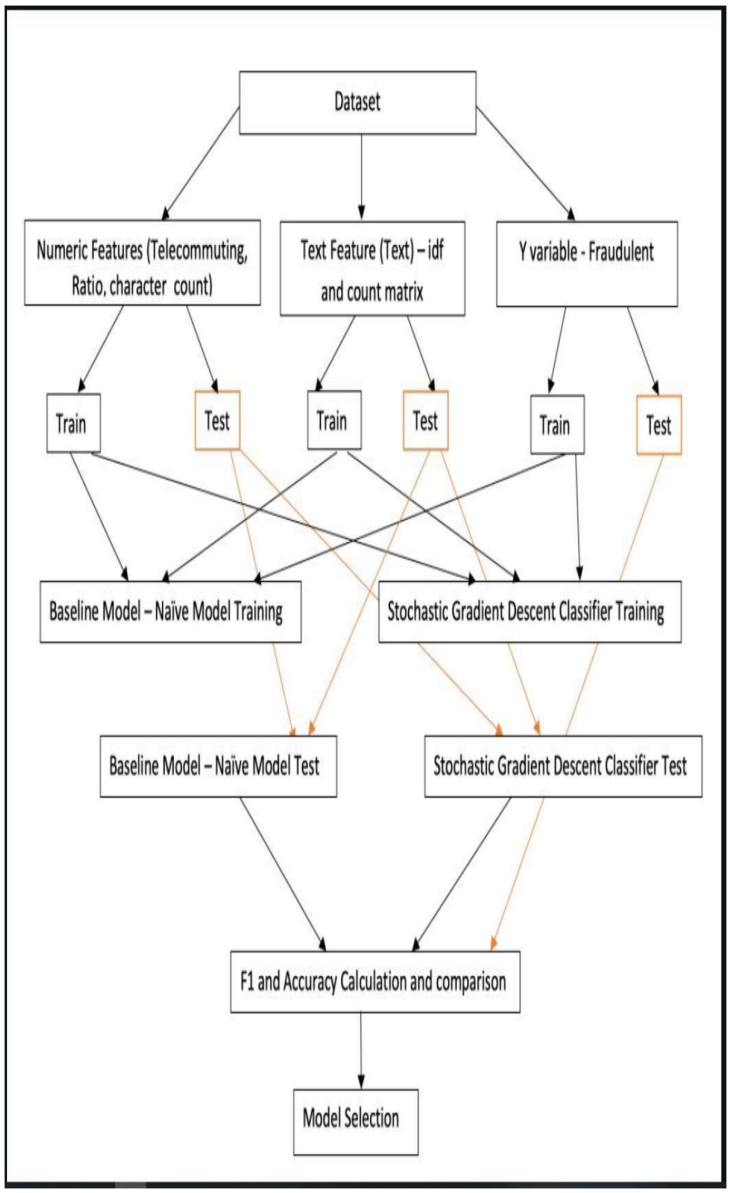


Figure4: Implementation of Algorithm

## III. Result and Discussions

- This paper surveyed different methods for the prediction of real and fake jobs using machine learning models.

- This survey paper determines that all the methods and algorithms listed have shown to be effective in the prediction of Online Recruitment Fraud. The Fake Job Recruitment has shown to be excellent in terms of accuracy.

- Various factors have been considered for the prediction process, which are deemed to be the best features for accurate predictions.

## IV. Conclusion

All the above mentioned classifiers are trained and tested for detecting fake job posts over a given dataset that contains both fake and legitimate posts. This paper is a comparative study of different algorithms that have been used by various methods for the effective prediction of ORF such as Machine learning algorithms and approaches i.e., Naïve Bayes, SGD Classifier. The approaches and algorithms used are seen to be effective in the prediction of the presence of fake jobs, shown to be an imperative factor in this process.

## References


[1] B. Alghamdi and F. Alharby, ―An Intelligent Model for no. 03, pp. 155–176, 2019, doi: 10.4236/jis.2019.103009.

Online Recruitment Fraud Detection," J. Inf. Secur., vol. 10,

[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.

[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.