EasyChair Preprint

№ 8717

Detection of Online Employment Scam Through Fake Jobs Using Random Forest Classifier

D. Madhavi, M.Sri Manisha Reddy and M. Ramya

August 28, 2022

DETECTION OF ONLINE EMPLOYMENT SCAM THROUGH FAKE JOBS USING RANDOM FOREST CLASSIFIER

Dr. D. Madhavi, Associate Professor (dasarimadhavi3@gmail.com**) ,Sridevi Women's Engineering College, Computer Science Engineering.**

M.Sri Manisha Reddy (manishareddy2626@gmail.com**)**, **M. Ramya(**ramyasri774625@gmail.com**),**

G. Sanjana (sanjana.gangam@gmail.com**)**

ABSTRACT

To avoid fraudulent post for job on the internet, an automated tool using machine learning based classification techniques. Different classifiers are used for checking fraudulent post on the web and the results of those classifiers are compared for identifying the best employment scam detection model. It helps in detecting fake job posts form an enormous number of posts. Two major types of classifiers, such as single classifier and ensemble classifiers are considered for fraudulent job post detection. However, experimental results indicate that ensemble classifiers are the best classification to detect scams over the single classifiers.

Keywords: Fake Job, Online Recruitment, Machine Learning, Ensemble Approach.

1.INTRODUCTION

Employment scam is one of the serious issues in recent times addressed in the domain of Online Recruitment Frauds (ORF). In recent days, many companies prefer to post their vacancies online so that these can be accessed easily and timely by the job-seekers. However, this intention may be one type of scam by the fraud people because they offer employment to job-seekers in terms of taking money from them. Fraudulent job advertisements can be posted against a reputed company for violating their credibility. These fraudulent job post detection draws a good attention for obtaining an automated tool for identifying fake jobs and reporting them to people for avoiding application for such jobs. For this purpose, machine learning approach is applied which employs several classification algorithms for recognizing fake posts. In this case, a classification tool isolates fake job posts from a larger set of job advertisements and alerts the user.

1.1. Ensemble Approach based Classifiers-

Ensemble approach facilitates several machine learning algorithms to perform together to obtain higher accuracy of the entire system. Random forest (RF) [8] exploits the concept of ensemble learning approach and regression technique applicable for classification-based problems. This classifier assimilates several tree-like classifiers which are applied on various sub-samples of the dataset and each tree casts its vote to the most appropriate class for the input.

2.PURPOSE

There are a lot of job advertisements on the internet, even on the reputed job advertising sites, which never seem fake. But after the selection, the so-called recruiters start asking for the money and the bank details. Many of the candidates fall in their trap and lose a lot of money and the current job sometimes. So, it is better to identify whether a job advertisement posted on the site is real or fake. Identifying it manually is very difficult and almost impossible. We can apply Machine Learning to train a model for fake job classification. It can be trained on the previous real and fake job advertisements and it can identify a fake job accurately.

3.LITERATURE SURVEY

Internet is one of the important inventions and a large number of persons are its users. These persons use this for different purposes. There are different social media platforms that are accessible to these users. Any user can make a post or spread the news through the online platforms. These platforms do not verify the users or their posts. So, some of the users try to spread fake news through these platforms. This news can be propaganda against an individual, society, organization or political party. A human being is unable to detect all these fake news. So, there is a need for machine learning classifiers that can detect this fake news automatically. Use of machine learning classifiers for detecting fake news is described in this systematic literature review.

3.1. Existing system:

In traditional system, so many implementations have already done in field of recognizing fake or real categories.

- **Review spam detection:**

People often post their reviews online forum regarding the products they purchase. It may guide other purchaser while choosing their products. In this context, spammers can manipulate reviews for gaining profit and hence it is required to develop techniques that detects these spam reviews. This can be implemented by extracting features from the reviews by extracting features using Natural Language Processing (NLP).

- **Email spam detection**

Unwanted bulk mails, belong to the category of spam emails, often arrive to user mailbox. This may lead to unavoidable storage crisis as well as bandwidth consumption. To eradicate this problem, Gmail, Yahoo mail and Outlook service providers incorporate spam filters using Neural Networks. While addressing the problem of email spam detection, content-based filtering, case based filtering, heuristic based filtering, memory or instance based filtering, adaptive spam filtering approaches are taken into consideration.

- **Fake news detection**

Fake news in social media characterizes malicious user accounts, echo chamber effects. The fundamental study of fake news detection relies on three perspectives- how fake news is written, how fake news spreads, how a user is related to fake news. Features related to news content and social context are extracted and a machine learning model are imposed to recognize fake news .

Disadvantages:

- There were no such systems for fake job prediction
- Less Accurate to be dump into real-time application.

3.2. Proposed system:

The target of this study is to detect whether a job post is fraudulent or not. Identifying and eliminating these fake job advertisements will help the jobseekers to concentrate on legitimate job posts only. a couple of classifiers are employed such as Naive Bayes Classifier, Decision Tree Classifier, K-nearest Neighbour Classifier, and Random Tree Classifier for classifying job post as fake. In this context, a dataset from Kaggle is employed that provides information regarding a job that may or may not be suspicious. The dataset has the schema as at first, the classifiers are trained using the 80% of the

entire dataset and later 20% of the entire dataset is used for the prediction purpose. The performance measure metrics such as Accuracy, are used for evaluating the prediction for each

Advantages:

- Accuracy is improvised
- Less Time computing

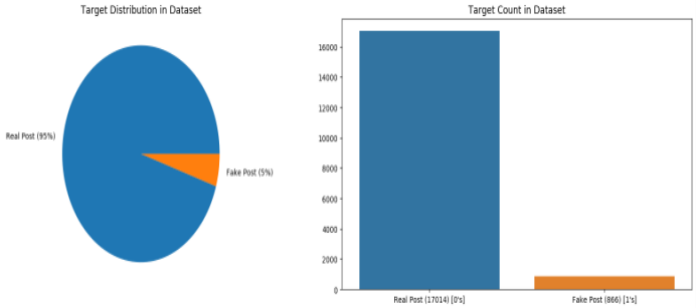
3.3. Data Collection:

We have trained and tested the dataset obtained from Kaggle as well as from the University of Aegon in order to detect the fake jobs. The data we have collected consists of 17 columns and nearly 18000 rows of textual data as well as numerical data for training and testing the machine learning and deep learning algorithms. These columns are related to the headings and the data present in various job posts being posted in online job hiring sites such as intern Shala, Naukri, etc. These data give a complete image of how the job is being posted online. Figure 1 shows the sample data being collected.

```
---  ------               --------------  -----
 0   job_id               17880 non-null  int64
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64
 10  has_company_logo     17880 non-null  int64
 11  has_questions        17880 non-null  int64
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64
dtypes: int64(5), object(13)
memory usage: 2.5+ MB
```

3.4. Data Analysis:

Once the data is collected, data analysis has to be done in order to have an insight about the data we are dealing with. The python modules and libraries such as pandas, NumPy, matplotlib and seaborn helps us to get a visual insight of the distribution of the data and provides a basic insight about the real jobs and fake jobs. From the analysis phase, we get an image of how unclean our data is and hence it requires data cleaning. Figure 2 shows the data analysis phase. Figure 2(a) shows the number of fake jobs and real jobs in the dataset.



3.5. Data Cleaning and Pre-processing:

After the data analysis is performed on the obtained data, we get that there are a lot of null values and textual data which needs to be cleaned. Hence, we first have a look at all the null values present in each column and remove the columns which have a large number of null values. After this we check for stop words. Stop words are all the unnecessary words which do not contribute for the detection of fake jobs. Figure 3 shows the cleaned data. The graph shows the distribution of unigrams and bigrams in the cleaned dataset. The graph to the left show unigrams and the graph to the right show the bigrams.

Once the stop words are removed from the data, all the textual data are combined together into a single column so that it is in a form suitable for the application of machine learning as well as deep learning algorithms.

```
# Deal with missing values and drop unneccessary columns

# Location missing values will be assigned none
data['location'] = data.location.fillna('none')

# department missing values will be assigned not specified
data['department'] = data.department.fillna('not specified')

# drop salary range, benefits, telecommuting, has_questions (not compulsory) in the cont
data.drop(['salary_range', 'benefits','telecommuting','has_questions'],
          axis=1, inplace=True)

# Company profile missing values will be assigned none
data['company_profile'] = data.company_profile.fillna('none')

# Company profile missing values will be assigned not specified
data['requirements'] = data.requirements.fillna('not specified')

# employment_type missing values will be assigned not specified
data['employment_type'] = data.employment_type.fillna('not specified')

# required_experience missing values will be assigned not specified
data['required_experience'] = data.required_experience.fillna('not specified')

# required_education missing values will be assigned not specified
data['required_education'] = data.required_education.fillna('not specified')

# industry missing values will be assigned not specified
data['industry'] = data.industry.fillna('not specified')

# function missing values will be assigned not specified
data['function'] = data.function.fillna('not specified')
```

4. System Architecture:

The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The solution architecture has features, properties, and characteristics which satisfy, as far as possible, the problem or opportunity expressed by a set of system requirements (traceable to mission/business and stakeholder requirements) and life cycle concepts (e.g., operational, support) and which are implementable through technologies (e.g. Mechanics, electronics, hydraulics, software, services, procedures, human activity).

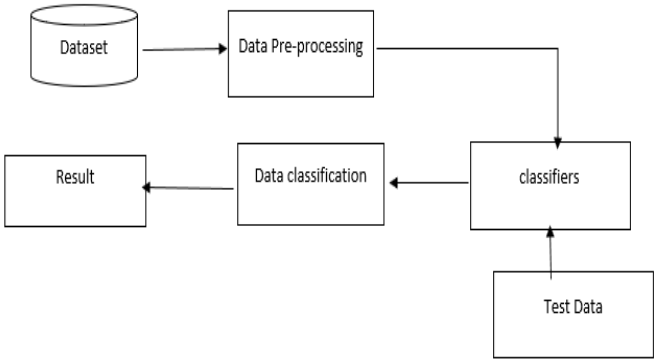


Fig : System Architecture

5.Data Flow Diagram:

A data flow diagram (or DPD for short) shows how processes flow through a system. It also gives you information about things such as the inputs and outputs (where things come from, which route they go through, and where they end up), and the process itself. This includes data stores and the various subprocesses the data moves through. Unlike a flow chart, there are no decision points. And unlike a network diagram, there are no loops. work out where there are issues and inefficiencies.

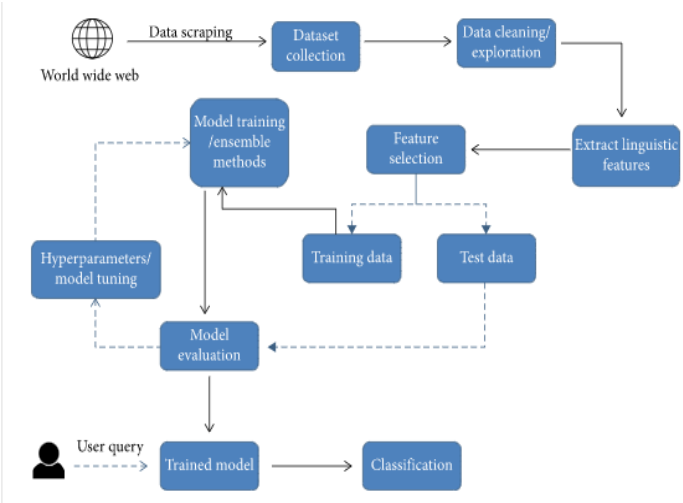


fig: Data Flow Diagram

6. RESULTS

```
accuracy with Logistic Regression: 0.9216069489685125 %
accuracy with Random Forest: 0.9939196525515743 %
accuracy with Support Vector Machine: 0.9218241042345277 %
accuracy with Decision Tree: 0.9819761129207383 %
accuracy with K-Nearest Neighbors : 0.9394136807817589 %
accuracy with Naive Bayes: 0.9259500542888165 %
```

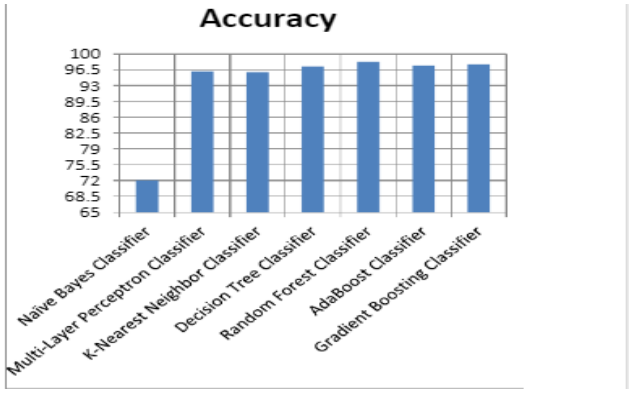
In [ ]:
```
test_vector = np.reshape(np.asarray([17614,5362,1393,1669,11417,1,7,13,75,37]),(1,10))
p = int(clf.predict(test_vector)[0])

if p==0:
    print('Job profile is Real')
else:
    print('Job profile is fake')
```

```
Job profile is Real
```

In [ ]:

PERFORMANCE COMPARISON CHART FOR ENSEMBLE CLASSIFIER BASED PREDICTION

Performance Measure Metric	Random Forest Classifier	AdaBoost Classifier	Gradient Boosting Classifier
Accuracy	98.27%	97.46%	97.65%
F1-Score	0.97	0.98	0.98
Cohen-Kappa Score	0.74	0.63	0.65
MSE	0.02	0.03	0.03



7.ALGORITHMS:

Algorithm Fit:

In this step train features and labels are fit to algorithm and model is saved to system which is used for prediction. In this step details are fed as input in the form of csv of various profiles and prediction is performed.

Steps to implement Random Forest Classifier in Python Algorithms

Step 1 - Import the Libraries We will start by importing the necessary libraries required to implement the Algorithm in Python. We will import the numpy libraries for scientific calculation.

Step 2 - Fetch the Data We will fetch the data from csv file using 'pandas_datareader'. We store this in a data frame 'df'.

Step 3- Split the Dataset we will split the dataset into training dataset and test dataset. We will use 70% of our data to train and the rest 30% to test. To do this, we will create a split parameter which will divide the data frame in a 70-30 ratio.

Step 4 - Instantiate Random Forest Classifier Model After splitting the dataset into training and test dataset, we will instantiate Random Forest Classifier fit the train data by using 'fit' function. Then we will store as model.

8.CONCLUSION

Employment scam detection will guide jobseekers to get only legitimate offers from companies. For tackling employment scam detection, several machine learning algorithms are proposed as countermeasures in this paper. Supervised mechanism is used to exemplify the use of several classifiers for employment scam detection. Experimental results indicate that Random Forest classifier outperforms over its peer classification tool. Employment scam detection will guide jobseekers to get only legitimate offers from companies. For tackling employment scam detection, several machine learning algorithms are proposed as countermeasures in this paper. Supervised mechanism is used to exemplify the use of several classifiers for employment scam detection. Experimental results indicate that Random Forest classifier outperforms over its peer classification tool. The proposed approach achieved accuracy 98.27% which is much higher than the existing methods.

9. FUTURE SCOPE:

Based on these insights, we now know that it is possible to find out which job postings are fake and which are not. But in these unprecedented times, where hundreds of individuals are being laid off every day, job seekers are desperate. The scammers are using this desperation to put out more and more fake job advertisements. Hence, we need more of these algorithms and tools being used on job search websites like LinkedIn, Glassdoor and Indeed so that these fake postings are filtered out, and the job seekers only see the genuine ones. huge amount of data take increase the prediction values improve the accuracy level

10. 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, 2020, 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 classificationand 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, 2017, 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. 2094–2097,2019,doi: 10.21275/v5i4.nov162954