Fake Job Prediction using Sequential Network

Devsmit Ranparia
Department of Mechanical Engineering
Indian Institute of Tecnology, Ropar
Ropar, India
2017meb1202@iitrpr.ac.in

Shaily Kumari
Department of Mechanical Engineering
Indian Institute of Tecnology, Ropar
Ropar, India
2018meb1263@iitrpr.ac.in

Dr, Ashish Sahani
Centre of Biomedical Engineering
Indian Institute of Tecnology, Ropar
Ropar, India
ashish.sahani@iitrpr.ac.in

2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS) | 978-1-7281-8524-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICIIS51140.2020.9342738

***Abstract*— With increased number of data and privacy breaches day-by-day it becomes extremely difficult for one to stay safe online. Number of victims of fake job posting is increasing drastically day by day. The companies and fraudsters lure the job-seekers by various methods, majority coming from digital job-providing web sites. We target to minimize the number of such frauds by using Machine Learning to predict the chances of a job being fake so that the candidate can stay alert and take informed decisions, if required. The model will use NLP to analyze the sentiments and pattern in the job posting. The model will be trained as a Sequential Neural Network and using very popular GloVe algorithm. To understand the accuracy in real world, we will use trained model to predict jobs posted on LinkedIn. Then we worked on improving the model through various methods to make it robust and realistic.**

Keywords—GloVe, Neural Networks, Fake Job post, NLP

I. INTRODUCTION

Evident from the daily news, we can observe number of cyber-crimes and digital breach occurring more frequently and causing havoc at an alarming rate. This has made job-seekers really vulnerable. Sum up it with the desperation of lending best, lucrative job offer and one can easily be the victim of fake job-posting with number of cases in thousands and economic lost in millions annually (for US), according to the reports by 'hashed out' [1]. Such cases can even be noticed in and around us. The companies lure the worthy talents to scam them for money, keep their records and use it as leverage for current employers to underpay them or cause favour to specific class. They have serious consequences on the mind of youth victims.

The problem is global and massive, yet possess a feasible and relatively simple solution. This task can be efficiently solved by using Machine Learning to predict the chances of a job being fake so that the candidate can stay alert and take informed decisions, if required.

II. CURRENT SCENARIO

Major job-providing platforms like Linked, Indeed Jobs, Glassdoor, etc. have been working on resolving this issue and have succeeded to a considerable level. They employ scrutinized authentication from the job posting companies along with detailed fact-checks and inspection. Yet, small, local platforms like monster.com, naukri.com, shine.com have remarkable number of such fraud postings which usually takes some time before they are brought down. Considerable amount of cases are observed when the posting and brochure are shared from person to person via social networking sites.

Although, the job-seekers have an option to check out and get reviews from internet, few fraudsters are able to even fill those gaps. On the contrary, some gets victimised due to their negligence. In all these situations, if the job seeker has an option to proof-check about the posting it can lead to drastic drop in number of victims.

With abundance of data available on internet and increasing capability of machine intelligence this task can be solved by using Machine Learning to predict the chances of a job being fake so that the candidate can stay alert and take informed decisions, if required. The model will be built from existing data published by trusted organizations and model can be trained which can be easily deployed to a local system via an app or web-extension.

III. DATA SOURCE

We have decided to use the dataset published by EMSCAD (EMployment SCam Aegean Dataset) which contains real life job ads posted by workable [2]. EMSCAD contains 17,014 legitimate and 866 fraudulent job ads. Data contains various information like job ID, job title, name of the organisation, location, company profile, employment type, job description, job requirements, benefits, required education, type, whether job posting is fraudulent or not, etc. This dataset contains both categorical and description format which is pre-processed to make it useful in training the model.

To give more relevancy and real life market exposure we extracted data from major online job portals, LinkedIn. This was done by web scrapping using Beautiful Soup library. The data was obtained in .json format which was then converted to .csv format. The data base had 138 job posts with details like job title, name of the organisation, location, employment type, job description, job function, type of the organisation, whether job posting. This data was manually analysed to identify if the posting is fake or not. This updated database was used to test model and evaluate the results which are more realistic.

IV. EXPLORATORY DATA ANALYSIS

We will perform Exploratory Data Analysis (EDA) [3] to achieve following tasks:-

- Minimize the computation
- Remove noise from the data
- Visualize the data for getting direction
- Improve accuracy of the model

A. Correlation Matrix

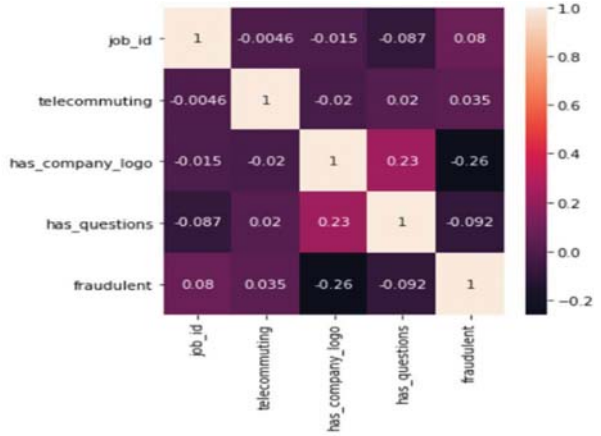

Fig. 1 Correlation matrix for few datafields

From correlation matrix (Fig. 1) It was clear that fraudulent has minimum correlation with respect to the data of job_id, telecommunicating, company_logo, has_questions, and it was acceptable to drop this data in building the model to get accurate model.

B. Visualizing the amount of fraud job posts

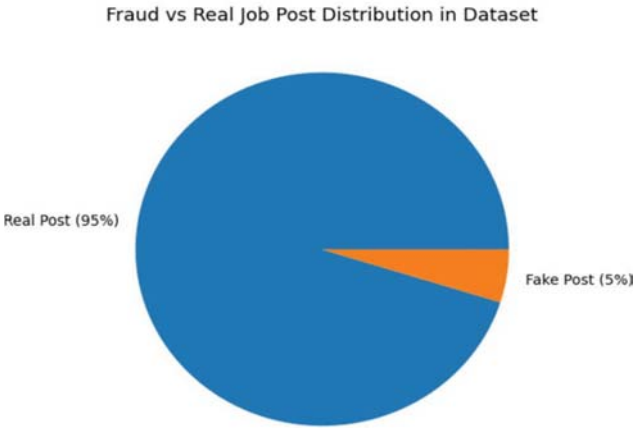

Fig. 2 Amount of fake vs real job posts on the data base

There were 866 fake job posts and 17,014 real job posts.

C. Visualizing the relative number of words in real job post

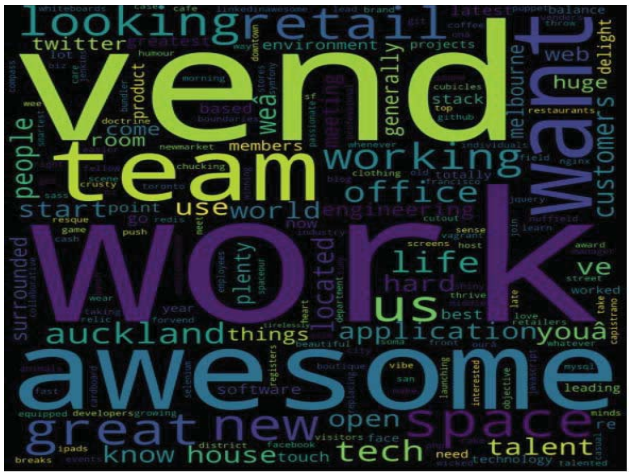

Fig. 3 WordCloud to show relative occurrence of common words in real post

D. Visualizing the relative number of words in fake job post

Fig. 4 WordCloud to show relative occurrence of common words in fake post

The differences in most occurring words in fake and real post can be directly observed and used as a potential tool to separate them.

E. Total number of words in a real job post

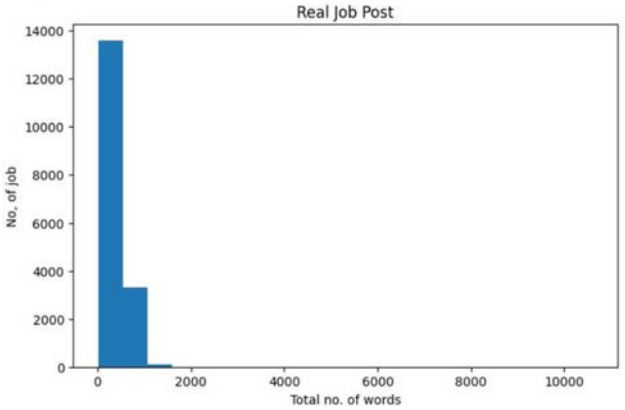

Fig. 5 Histogram to depict range of words in all real job post

If we observe carefully the number of words the real job posts range till 10,000 and most of them is around 1000.

F. Total number of words in a fake job post

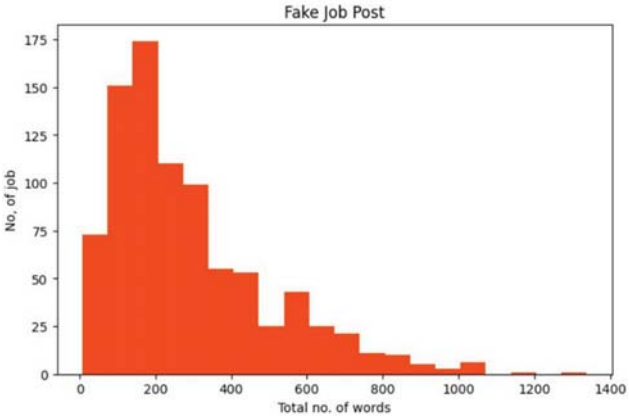

Fig. 6 Histogram to depict range of words in fraud job posts

If we observe closely we find that highest number of words in any fake job description was merely 1400 and mostly less than 200 words.

V. Preprocessing the Data

To get accurate results and minimize the computation costs, preprocessing the data is a very important step. Before building our model we preprocessed the available data by firstly by removing all the Null values as they can give us erroneous results. After that we combined all the available text. Then we converted them to lower case, remove all the punctuations, remove all the numerical data, remove all the links using regular expressions and Vectorizer from SciKit Learn Library.

Then we remove all the stop words that occur frequently and does not make much sense to the results by using nltk toolkit and then make the data compatible for use by using preprocessing function from Scikit Learn library.

These steps will allow minimum noise to enter in the data and provide more accurate and robust model.

VI. Proposed Machine Learning Model

As observed in Exploratory Data Analysis 'D' and 'E'; we deduced how the description, number of words and its sentiment was strongly related to the job being fraudulent. This guided us in using Natural Language Processing (NLP) to perform sentiment analysis, find semantics pattern of words and build our model around it.

NLP deals with understanding of human natural language, analyze large amount of natural language data, understand logic, context and semantics, to analyze the patterns and build a ML model around it. To identify various types of jobs, we need to understand the contextual and logical meaning of each word. GloVe, Word2vec and Fasttext are the most popular word embedding models used worldwide which can solve our problem.

We decided to use GloVe [4] model which stands for Global Vector model. GloVe is the most recent and advanced NLP algorithm developed by researchers from Stanford based on the principle of Word Embedding where we incorporate context and meaning of the word. It is preferred over other models as, unlike other models it preserves the data of word for a long time and hence finding the context of the word with respect to global corpus. It is a global log-bilinear regression model with weighted least square objective. It incorporates properties of global matrix factorization and local context window method.

Here each word is represented as a vector with a specific value. Word vector method relies on distance or angle between pair of word vectors as primary method for evaluating intrinsic quality of set of words. Euclidean distance (or cosine similarity) between word vectors relates to measure of linguistic and semantic similarity between words which in turn gives word vectors.

```
1 glove_model.most_similar("man")

/usr/local/lib/python3.6/dist-package
  if np.issubdtype(vec.dtype, np.int)
[('woman', 0.6998662948608398),
 ('person', 0.6443442106246948),
 ('boy', 0.620827853679657),
 ('he', 0.5926738977432251),
 ('men', 0.5819568634033203),
 ('himself', 0.5810033082962036),
 ('one', 0.5779520869255066),
 ('another', 0.5721587538719177),
 ('who', 0.5703631639480591),
 ('him', 0.5670831203460693)]
```

Fig. 7 Euclidean of words man with most nearest 10 words

The word vectors' difference is used to estimate relation of a word with other words but the juxtaposition of words may not be enough to understand complete sense of the words' relation. So linear sub-structures are made where the words of similar categories, family, class, sex, age, etc. are clustered together. This provides more than just a relation between two words.

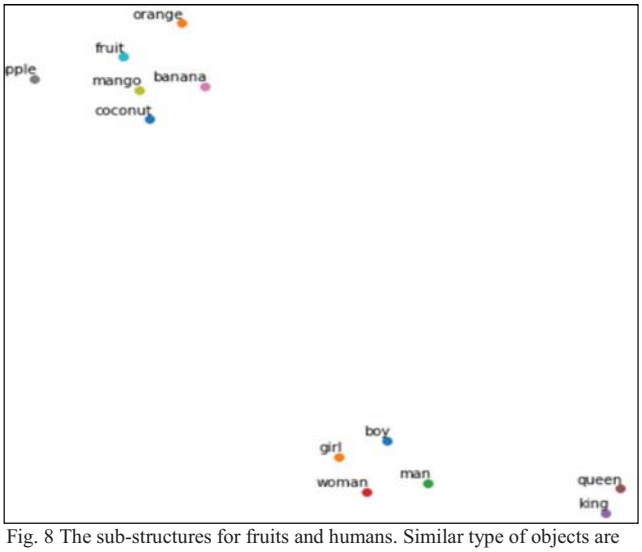

Fig. 8 The sub-structures for fruits and humans. Similar type of objects are clustered together

The algorithm is trained only on non-zero word-word co-occurrence matrix which minimizes its run-time as compared to traditional NLP algorithms. The co-occurrence matrix is obtained from the glove data provided by Stanford [5].

The algorithm is based on the principle that ratio of word-word co-occurrence probabilities have potential for encoding some form of meaning.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad (1)$$

; where w's are word vector and P_{ik}'s are probability of occurring of word 'i' given word 'j'.

Further upon evaluating we can prove that:

$$w_i^T \widetilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i) \quad (2)$$

; where w_i^T is transpose of w_i, X_{ik} is number of times word 'k' occurs in the context of 'i'

Finally adding bias and simplifying we get:

$$w_i^T \widetilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) \quad (3)$$

; where b_i represents the bias

The previous equation serves as the base of the algorithm and later all the other steps are performed like any other regression or classification algorithm by making a neural network.

VII. Building the Model

- First step is loading the data in the notebook through .csv file.
- Then we perform EDA as explained in Section IV.
- After that data is preprocessed as explained in Section V.
- Then we build the model [6] on the principle explained in the Section VI. For that we load the pre-trained glove vectors and tokenize them to make it usable.
- Then we normalize the vectors for each sentence and transform them to an array to feed to model.
- Then we made a sequential Neural Network of 2 layers 'relu' and 'sigmoid' activation.
- Then the model was compiled with 'binary cross-entropy' as loss function and 'adam' as optimizer.
- After that we compiled the model and evaluated loss, accuracies and AUC score.
- Then performance of model with respect to epoch was plotted.
- In the end we took pre-processed data obtained by web-scraping LinkedIn to do more realistic evaluation of model.

VIII. Performance Evaluation

After compiling the model for 10 epochs and with batch size of 64, we achieved accuracy of 97.94% and validation accuracy of 97.58%

```
- loss: 0.0791 - accuracy: 0.9734 - val_loss: 0.0919 - val_accuracy: 0.9727
- loss: 0.0731 - accuracy: 0.9749 - val_loss: 0.0832 - val_accuracy: 0.9749
- loss: 0.0666 - accuracy: 0.9790 - val_loss: 0.0861 - val_accuracy: 0.9756
- loss: 0.0608 - accuracy: 0.9792 - val_loss: 0.0883 - val_accuracy: 0.9749
```

Fig. 9 model details after 10 epochs

Later we plotted accuracies and losses of model with respect to epochs to evaluate the performance of models.

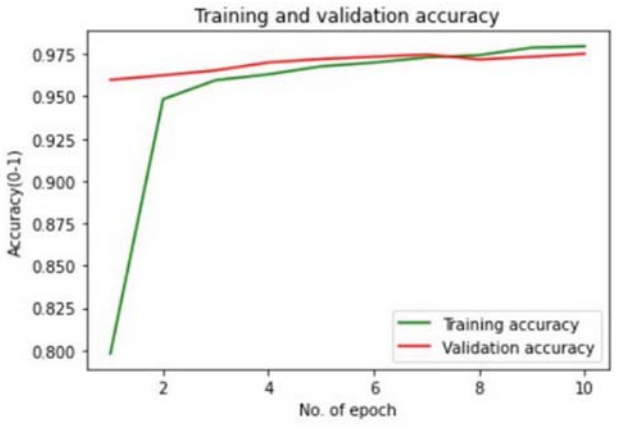

Fig. 10 Accuracies with respect to epochs

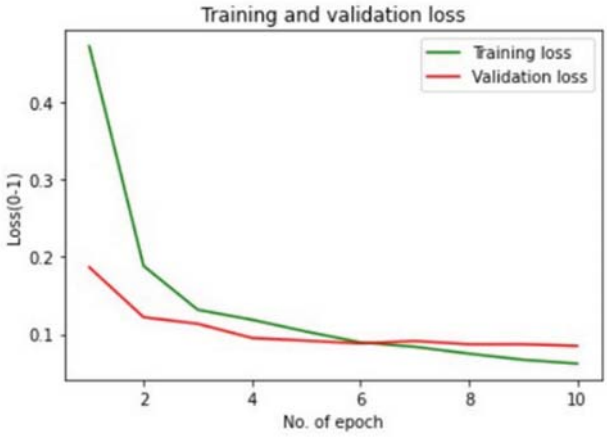

Fig. 11 Accuracies with respect to epochs

Then we took 138 non-fraudulent job-posts and predicted their output. The model predicted 136 jobs as non-fraudulent and 2 as fraudulent with an accuracy of 98.38%

```
1
The accuracy obtained using the data from Linked is:  99.27536231884058
```

Fig. 12 Accuracy of prediction of jobs from LinkedIn

The high accuracy proves that the GloVe algorithm is very realistic and can be used directly for all the real job posts. One needs to extract the job description and feed it to the model which will tell whether job is real or not.

Part No: CFP2058A-ART; ISBN No:978-1-7281-8524-8/20

To further make our model robust, we used remove_stopwords function from genism package and the result was positive. There was slight increase in the accuracy.

Fig. 13 Accuracy of prediction of jobs from LinkedIn using gensim package function

IX. CONCLUSION

From the results we can conclude that the model was highly accurate achieving the accuracy more than 97%. Moreover the model was robust and realistic as the accuracy of model even with the realistic LinkedIn data was more than 98%. The popular GloVe algorithm is easy to deploy and can be used in real world NLP applications. Thus, this easy to deploy, realistic model can be conveniently used by the users to get highly reliable prediction, alerting them and assist them.

X. FUTURE WORK

The results obtained through GloVe algorithm is very promising. Later, its performance can be compared to similar version Word2vec which is also a popular NLP algorithm for sentiment analysis. We are planning to combine both the models with certain weightage to each model and find the optimal weight for both models which gives us the best results.

XI. REFERENCES

[1] C. Crane, "Fake Jobs: Cybercriminals Prey on Job Seekers via Fake Job Postings," *Hashed Out by The SSL Store™*, 28-Jan-2020. [Online]. Available: https://www.thesslstore.com/blog/fake-jobs-cybercriminals-prey-on-job-seekers-via-fake-job-postings/. [Accessed: 3-May-2020].

[2] G. Kambourakis, *"Employment Scam Aegean Dataset* http://emscad.samos.aegean.gr" Unpublished, 2017, doi: 10.13140/RG.2.2.12872.72962.

[3] P. Simard, D. Steinkraus, and J. Platt, *"Best practices for convolutional neural networks applied to visual document analysis,"* Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings

[4] J. Pennington, R. Socher, and C. Manning, *"Glove: Global Vectors for Word Representation,"* Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[5] *Glove dataset*, Stanford,edu , April 2020 [Online]. Available : http://nlp.stanford.edu/data/glove.6B.zip

[6] S. Bansal, "[Real or Fake] Fake JobPosting Prediction," *Kaggle*, 29-Feb-2020. [Online]. Available: https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction. [Accessed: 6-March-2020].