Online Recruitment Fraud Detection using ANN

Ibrahim M. Nasser
Independent Researcher
Gaza, Palestine
ibrahimnasser.research@gmail.com

Amjad H. Alzaanin
Faculty of Information Technology
Islamic University of Gaza
Gaza, Palestine
amjad@live.it

Ashraf Yunis Maghari
Faculty of Information Technology
Islamic University of Gaza
Gaza, Palestine
amaghari@iugaza.edu.ps

***Abstract*—Online recruitment provides job-seekers an efficient search and reach for jobs. It also helps recruiters searching for qualified candidates which improves the recruitment process. However, employment scam has emerged as a critical issue. Some job posts are legitimate, and others are fraud. In this paper, an Artificial Neural Network based model is proposed to detect fraud job posts. The public Employment Scam Aegean Dataset (EMSCAD) is used with proper text preprocessing techniques for training and testing the proposed model. Our model has precision, recall, and f-measure of 91.84%, 96.02%, and 93.88% respectively. The results show that the proposed ANN-based model outperforms similar existing models in detecting fraud jobs.**

Keywords—Fraud Detection, Text Classification, ANN.

I. INTRODUCTION

Due to the recent global pandemic, a lot of job recruiters have shifted to online recruiting (work from home). People now can apply to several jobs available on the internet which is more effective and takes less time than the traditional way of jobs applying and recruiting. Online recruitment provides the ability for candidates to search and reach jobs with just a click of a button. Moreover, it helps recruiters in looking up for qualified candidates which improves the efficiency of the recruitment process [1]. However, fraud people can scam job seekers through online recruitment by posting fake jobs to steal job-seekers' money which is called "Employment Scam" (ES). ES is one of the major issues in the domain of Online Recruitment Fraud (ORF). ORF is considered a cybercrime because it violates organizations' privacy and funds [2]. It provides fraud people the ability to damage the reputations of many reputed organizations [3].

ORF detection is a Text Classification (TC) problem. TC is one of the fundamental disciplines in the context of Natural Language Processing (NLP). Nowadays, there is a huge amount of text data available online. Text data have rich and important information, however, extracting these information from such large documents takes too much time without NLP Machine Learning (ML) models. Problems vary within the TC domain, including categorizing new articles, sentiment analysis, spam messages detecting, text retrieval, and more. TC systems consist of three common phases: preprocessing, tokenization, and feature extraction. After all these steps, the input becomes suitable to be fed into an ML model and start learning [4]. Fig. 1 illustrates the general NLP pipeline.

In this paper, an Artificial Neural Network (ANN) based model is proposed to detect the fraud job posts using the EMSCAD public dataset that contains fraud and real job posts [5]. The model starts with preprocessing steps, training the ANN, and then the evaluation.

The rest of the paper discusses the previous related works, our proposed model, testing results, and lastly our conclusion and future work.

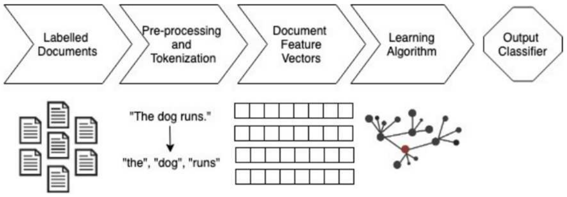


Fig. 1. General NLP Pipeline

II. LITERATURE REVIEW

The first work published using EMSCAD dataset is by Vidros et al [6]. Moreover, they are the creators of this public data which was the first publicly available dataset about fraud and legitimate job offers. They have tested the Bag-Of-Words model (BOW) and empirical rulesets analysis. Furthermore, they have tested several classifiers on both balanced (processed) data, and imbalanced (the created original data). BOW worked well on balanced data; Random Forest (RF) achieved the highest accuracy, precision, and recall of 91.2%, 91.4%, 91.2% respectively. The ruleset model resulted in higher accuracy (2%-13%) for many ML models except for RF's accuracy which decreased by 0.5%. Lastly, they have tested RF that trained on empirical ruleset against the imbalanced data (original). Research findings showed that RF achieved an accuracy of 89.5%. RF's precision and recall for non-fraud were 98.6% and 90.3%. However, for the fraud class they were 28.2% and 75.1% respectively.

Furthermore, Alghamdi and Alharby [2] have applied feature extraction using Support Vector Machine (SVM) and then they used RF for classification. Utilizing the WEKA tool, the best features were: company profile, is there a company logo, and industry. The final preprocessed data has consisted of the attributes: "location, salary range, company profile, description, company logo, has questions, employment type" which are all binary and nominal data. They have fed the imbalanced data to the classification model RF. Their research findings showed that RF has achieved 0.88, 0.54, and 0.67 for precision, recall, and F-measure of the fraud class accordingly.

Lal et al. [7] Have proposed an ensemble machine learning to detect fraud jobs. The ensemble techniques used are Average Vote (AV), Majority Vote (MV), and Maximum Vote (MXV). Also, the ensemble classifier consisted of three base classifiers, which are: J48 Decision Tree (DT), Logistic Regression (LR), and RF. They have trained the proposed model on imbalanced dataset. And they have used the empirical ruleset from [6] as features. Their proposed model resulted in 94% f-measure. Moreover, the model is 8% higher in specificity (77.8%) than the models in their research literature. Precision, recall and sensitivity were 94.9%, 95.6%, 95.7% respectively.

The performance of Artificial Neural Network (ANN) has been investigated by Kim et al [8]. They have contributed to the state-of-the-art by proposing a hierarchical cluster-based deep neural network (HC-DNN) to detect fraud recruitment. The core idea of their proposed model is to train a deep neural network that has pre-trained clusters of several layers to improve the performance of detecting fraud jobs. The pre-trained hierarchical clusters are used for weight initializing through autoencoding and the DNN for the prediction. The dataset was transformed by principal component analysis (PCA) and resulted in 48 features. The proposed model achieved 98.4% accuracy and 0.09 f-measure, which, as they have mentioned, better than the traditional NN that achieved accuracy and f-measure of 0.55 and 0.03 accordingly. The HC-DNN is a great contribution. Nevertheless, the dataset is different from the one we work on and there are no other metrics provided other than accuracy and F-measure.

In addition, Anita et al [1]. have applied several machine learning and deep learning methods to detect fake jobs. They have applied the simple machine learning models: K-Nearest Neighbors (KNN), RF, and LR. And applied Bi-directional Long Short-Term Memory Neural Network (Bi-LSTM-NN) which is a deep learning model. The best performance was achieved by RF and Bi-LSTM-NN. They have achieved 98% accuracy. However, the Bi-LSTM-NN achieved more recall and f-score, in detecting the fraud jobs (0.64>0.50, 0.72>0.63). Bi-LSTM-NN model achieved high recall and f-measure in detecting fraud. It also achieved a precision 0.82. However, these results are not high enough for efficient fraud detection. Probably, the reason behind this is that research did not handle the problem of imbalanced data. A lot of information would be helpful if authors have provided the resulted confusion matrix.

The main goal of ORF detection is to spot the fraud jobs. However, the data contains more legitimate data than fraud. This problem resulting in high performance detecting legitimate jobs, and low performance in detecting fraud. For instance, we could implement a classifier that returns just "legitimate" and results in 99% accuracy because there is only 1% of fraud data. That is why other metrics are important such as precision, recall, specificity, f-measure, and more which we explain later in the paper. A researcher should monitor the results of these metrics on the fraud class, because we are interested in detecting the fraud more than legitimate.

In our research, we handle the imbalanced data problem that was not handled in many papers, by down-sampling the data, not over-sampling it as in [8]. The common phases of text pre-processing are applied. Moreover, the resulted Confusion Matrix (CM) is illustrated as well as a variety of performance measures, including accuracy, precision, recall, specificity, and more. Our MLP is simple, we did not use any pre-trained models as in [8], due to their computational complexity [9].

III. The Proposed Model

Details about our proposed model are explained in this section. Fig. 2 shows the general steps of the proposed classification model.

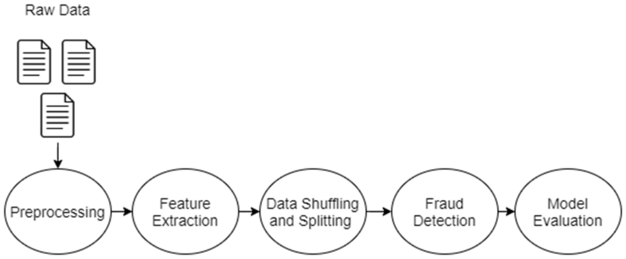


Fig. 2. Our Proposed Model

A. Dataset

The original EMSCAD dataset contains 17,880 real-life job advertisements. Ads are classified into fraudulent (1) and legitimate (0). There are 17,014 legitimate data points, and 866 frauds. The dataset attributes are described in Table 1.

Table 1. Original Dataset Description

String	
Title	The title of the job post
Location	Geographical location of the job post
Department	Corporate department (e.g., sales)
Salary range	E.g., $50,000-$60,000
HTML fragment	
Company profile	A short company description
Description	Details of the job ad (post)
Requirements	Required knowledge to apply
Benefits	Employer offered benefits
Binary (true (1), false (0))	
Telecommunicating	True for telecommunicating positions
Company logo	True if there is a company logo
Questions	True if there are questions for applicants
Fraudulent	Classification attribute
Nominal	
Employment type	Full-time, part-time, etc.
Required experience	Entry level intern, etc.
Required education	Bachelor, master, etc.
Industry	IT, health care, etc.
Function	Research, Engineering, etc.

B. Data Pre-Processing

Imbalanced Data Handling

We have discussed the problem of imbalanced data in literature review section. Fig. 3 shows the imbalanced nature of the raw data. Common ways to handle such problem include over-sampling, and down-sampling. In over-sampling [10], data from the minority class (fraud) are duplicated until the imbalance is banished. On the other hand, down-sampling method [11] work on eliminating random data points from the majority class (legitimate) to rebalance the data. We choose the down-sampling approach due the nature of neural networks architecture, as stated in [12], ANNs treat the minority class as noise and therefore discard those data points. The researchers have demonstrated that the down-sampling technique outperforms the over-sampling, because the later have been resulted in slowing down the ANN convergence and does not bring new information. After data down-sampling, the resulted balance is shown in Fig. 4.

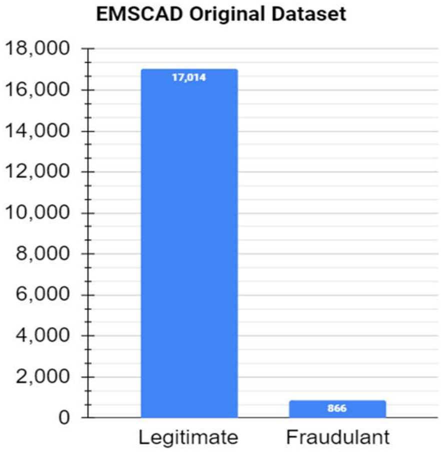


Fig. 3. Imbalanced Data

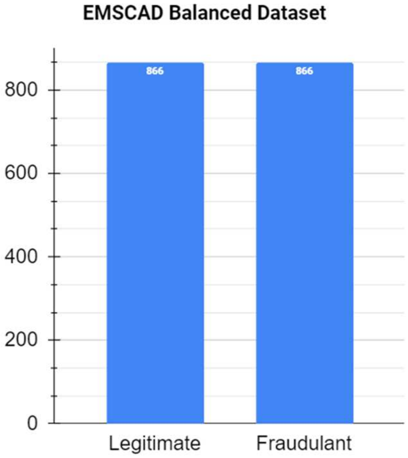


Fig. 4. Balanced Data

Cleaning

Irrelevant features, namely; “title, location, department, salary range, and function” are removed from the data. The Null values in the columns: “employment type, required experience, required education” are replaced with “unknown”.

On the other hand, the HTML fragment data was handled differently; Null values of HTML features led to the deletion of the entire datapoint. The resulted data still have (some) Null HTML fields, those are replaced by nothing (space).

The binary columns: “company logo, and questions” and the nominal: “employment type, required experience, required education”, are transformed to better representation. For instance, instead of one column for employment type, there are many columns with all the types, each column contains binary data. The feature “industry” contains 513 null values which are also replaced with nothing.

So far, just the following fields contain string data: “company profile, description, requirements, benefits, and industry”. A function is applied on them for lowercasing the alphabetics, and removing text in brackets, URL links, HTML tags, punctuations, and line breaks. Then a data shuffling is applied randomly.

Afterward, the cleaned resulted data is separated to two; the input features and the classification labels. The input features are the input of the steps from tokenization to feature extraction.

Tokenization

Tokenization is the process of breaking a text into relevant elements called tokens (in our case, words). This process aims to explore the words in given sentences [13], [14].

Stop words removal

Text data contains several words that do not have a significance weight in classification problems, such as: {“a”, “about”, “after”, “again”, etc.}. The proper way to handle them is a total deletion from tokens as stated in [15].

C. *Feature Extraction*

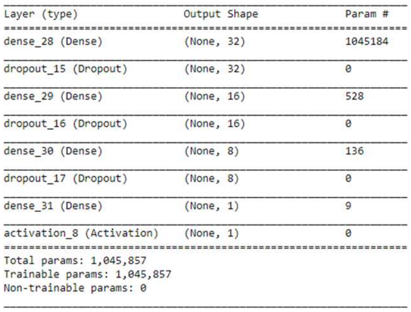
ML models does not accept (string) data as input, instead, they deal with numerical data (int, float, binary). For this purpose, we used the function Count Vectorization (CV). Vectorization is a technique for feature extraction, which considers the BOW approach that aims to represent each word as a feature so the whole data would be a set of features. CV assigns the count of each word in each text field [16]. The resulted vectorized input contains 32,629 features.

D. *Data Shuffling and Splitting*

Data is being shuffled randomly to ensure balancing in classes distribution. Also, data has been split into 80% for training and 20% for validation.

E. *Fraud Detection*

Our ANN contains three basic hidden layers with the activation function: *ReLU* [17] and with number of neurons: 32, 16, 8 accordingly. After each hidden layer we have added a *dropout* layer of 0.2 probability to prevent over fitting [18]. Lastly, the output layer is added with one neuron applying the *sigmoid* activation function [19] because it is a binary classification task. The binary cross entropy has been chosen to be the model loss function. Which is a special case of the cross entropy function proposed by Kullback [20] where the prediction task is binary [21]. For optimization, we chose the optimizer function *ADAM* [22]. The architecture of our ANN is shown in Fig. 5.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_28 (Dense)             (None, 32)                1045184
_________________________________________________________________
dropout_15 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_29 (Dense)             (None, 16)                528
_________________________________________________________________
dropout_16 (Dropout)         (None, 16)                0
_________________________________________________________________
dense_30 (Dense)             (None, 8)                 136
_________________________________________________________________
dropout_17 (Dropout)         (None, 8)                 0
_________________________________________________________________
dense_31 (Dense)             (None, 1)                 9
_________________________________________________________________
activation_8 (Activation)    (None, 1)                 0
=================================================================
Total params: 1,045,857
Trainable params: 1,045,857
Non-trainable params: 0
_________________________________________________________________
```

Fig. 5. Our ANN Architecture

F. Evaluation Metrics

The proper way to evaluate an ML model is using the Confusion Matrix (CM). CM is a visual evaluation tool for any ML method. The columns of a CM represent the prediction class results, the rows, however, represent the actual class samples [23]. In our case, a binary CM is used, which is 2 * 2 CM as shown in Table 2. Where TP is number of positive instances that are predicted positive. FN is the number of positive instances that are predicted as negative. TN is the number of negative instances that are predicted as negative. And FP is the number of negative instances that are predicted as positive.

Table 2. The Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Here is an explanation of the evaluation metrics we derived and used to evaluate our proposed model:

1. Recall (sensitivity): also called true positive rate (TPR). It calculates the fraction of correctly predicted positive instances out of all the actual positive instances (TP/TP + FN).
2. Specificity: also called true negative rate (TNR). It calculates the fraction of truly predicted negative instances out of the actual negative instances (TN/TN+FP).
3. Precision: also called positive predicted value (PPV). It calculates the fraction of truly predicted positive out of all the positive predicted instances (TP/TP+FP).
4. Negative predicted value (NPV): calculates the fraction of truly predicted negative out of all the negative predicted instances. (TN/TN+FN).
5. Accuracy: Number of truly predicted over the number of all samples. ((TP+TN)/all).
6. F (1) score for positive class: also called f-measure. It is the harmonic of recall and precision of the positive class as shown in equation (1).
7. F (1) score for negative class: also called f-measure. It is the harmonic mean of recall and precision of the negative class as shown in equation (2).

$$F\ measure_{positive} = 2 * \frac{PPV * TPR}{PPV + TPR} \tag{1}$$

$$F\ measure_{negative} = 2 * \frac{NPV * TNR}{NPV + TNR} \tag{2}$$

IV. TESTING RESULTS

After 20 training epochs, the resulted confusion matrix is shown in Table 3. The evaluation metrics results are shown in Table 4.

Table 3. The Resulted Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	155	15
Actual Positive	7	169

Table 4. Evaluation Metrics Results

Metric	Result
TPR	**96.02%**
TNR	91.17%
PPV	**91.84%**
NPV	95.67%
Accuracy	93.64%
F (1) positive	**93.88%**
F (1) negative	93.36%

Whether we are talking about detecting fraud, or knowing the legitimate, the overall performance of our classification model is outstanding. However, since we are interested in detecting the fraud (positive) out of the job posts, the recall (sensitivity/TPR) is the most important metric. Results show that the percentage of correctly predicted fraud instances out of all the actual fraud instances is 96.02%.

Considering the fraud detecting problem, the situation of not detecting the job as fraud (low sensitivity) could be threatening for job-seekers. Whilst the low specificity (predicting legitimate job as fraud) may only cause a further inspection by a human given the fact that real jobs would be obvious to realize. However, the problem lies in tricking people with fraud jobs that may look like real ones. That why authors of this paper gave the sensitivity a higher relative importance than the specificity. Table 5 shows a comparison between the proposed model and the existing models considering recall as the most important metric (ascending sorted by recall).

Table 5. Models Comparison

Ref.	Data	Features	Method	Recall	F (1)
Anita [1]	Imbalanced	-	RF	50%	63%
Alghamdi [2]	Imbalanced	SVM	RF	54%	67%
Anita [1]	Imbalanced	-	BI-LSTM	64%	72%
Vidros [6]	Imbalanced	Ruleset	RF	75.1%	41%
	Balanced	Ruleset	RF	90.6%	90.6%
	Imbalanced	BOW	RF	91.2%	91.2%
Lal [7]	Imbalanced	Ruleset	OFRDetector	95.9%	94%
Kim [8]	Different data	PCA	HC-DNN	-	9%
			ANN	-	3%
Proposed	**Balanced**	**BOW**	**ANN**	**96.02%**	93.88%

In [7], the researchers' ensemble method was 8% higher in specificity (77.8%) than the models in their literature. Our model, however, achieved a higher specificity which is 91.17%.

V. Conclusion and Future Work

Employment scam is a critical issue in Online Recruitment Fraud. It is also considered a cybercrime. Fraud people can scam job seekers through online recruitment by posting fake jobs to trick job-seekers and violating organizations' privacy and funds.

In this paper, an ANN-based model was proposed to detect the fraud job posts using a public dataset contains fraud and real job posts. The model started with raw-data preprocessing phases including imbalanced data handling, and count vectorization which was used as feature extraction. At last, ANN has been trained and evaluated.

Evaluation results showed that our proposed model outperforms the existing ANN-based models in the literature. It achieved recall and f-measure of 96.02%, and 93.88. respectively. The model also achieved a NPV score of 95.67% and f-score of 93.36% for the legitimate class.

In the future, we intend to investigate different preprocessing methods, and different feature extraction techniques so we can derive semantic information from the raw text data. We believe that when the classification model understands semantic relations in text data, it will result in better performance that leads to higher accuracy in detecting fraud job advertisement.

References

[1] C. S. Anita, P. Nagarajan, G. A. Sairam, P. Ganesh, and G. Deepakkumar, “Fake Job Detection and Analysis Using Machine Learning and Deep Learning Algorithms,” *Rev. GEINTEC-GESTAO Inov. E Tecnol.*, vol. 11, no. 2, pp. 642–650, 2021.

[2] B. Alghamdi and F. Alharby, “An intelligent model for online recruitment fraud detection,” *J. Inf. Secur.*, vol. 10, no. 03, p. 155, 2019.

[3] “Report | Cyber.gov.au.” https://www.cyber.gov.au/acsc/report (accessed Jun. 19, 2021).

[4] A. Pagotto, “Text Classification with Noisy Class Labels.” Carleton University, 2020.

[5] “Employment Scam Aegean Dataset.” http://emscad.samos.aegean.gr/ (accessed Jun. 19, 2021).

[6] S. Vidros, C. Kolias, G. Kambourakis, and L. Akoglu, “Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset,” *Futur. Internet*, vol. 9, no. 1, p. 6, 2017.

[7] S. Lal, R. Jiaswal, N. Sardana, A. Verma, A. Kaur, and R. Mourya, “ORFDetector: ensemble learning based online recruitment fraud detection,” in *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 2019, pp. 1–5.

[8] J. Kim, H.-J. Kim, and H. Kim, “Fraud detection for job placement using hierarchical clusters-based deep neural networks,” *Appl. Intell.*, vol. 49, no. 8, pp. 2842–2861, 2019.

[9] T.-J. Yang, “Neural network simplification using a progressive barrier based approach.” Massachusetts Institute of Technology, 2018.

[10] C. X. Ling and C. Li, “Data mining for direct marketing: Problems and solutions.,” in *Kdd*, 1998, vol. 98, pp. 73–79.

[11] M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets: one-sided selection,” in *Icml*, 1997, vol. 97, pp. 179–186.

[12] N. Japkowicz, C. Myers, and M. Gluck, “A Novelty Detection Approach to Classification, in proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95).” Montreal, Canada, 1995.

[13] G. Gupta and S. Malhotra, “Text document tokenization for word frequency count using rapid miner (taking resume as an example),” *Int. J. Comput. Appl.*, vol. 975, p. 8887, 2015.

[14] T. Verma, R. Renu, and D. Gaur, “Tokenization and filtering process in RapidMiner,” *Int. J. Appl. Inf. Syst.*, vol. 7, no. 2, pp. 16–18, 2014.

[15] H. Saif, M. Fernandez, Y. He, and H. Alani, “On stopwords, filtering and data sparsity for sentiment analysis of twitter,” 2014.

[16] U. Suleymanov and S. Rustamov, “Automated news categorization using machine learning methods,” in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 459, no. 1, p. 12006.

[17] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” 2010.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] V. S. Bawa and V. Kumar, “Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability,” *Expert Syst. Appl.*, vol. 120, pp. 346–356, 2019.

[20] S. Kullback, “Information theory and statistics". John Wiley & sons Inc., New-York,” 1959.

[21] A. U. Ruby, D. I. Prasannavenkatesan Theerthagiri, and Y. Vamsidhar, “Binary cross entropy with deep learning technique for Image classification,” *Int. J.*, vol. 9, no. 4, 2020.

[22] D. P. Kingma and J. L. Ba, “Adam: A Method for stochastic Optimization,” 2015.

[23] J. Xu, Y. Zhang, and D. Miao, “Three-way confusion matrix for classification: A measure driven view,” *Inf. Sci. (Ny).*, vol. 507, pp. 772–794, 2020.