# Sure-Bo: A Data-Driven Web Application to Detecting Online Job Scams

*INF1002 - P10 - Group 64*

*Source code:* https://github.com/FishPain/sure-bo

Yu Haotong
2302705
2302705@sit.singaporetech.edu.sg

Bryan Chan Zi Jie
2302717
2302717@sit.singaporetech.edu.sg

Leena Soo Wei Qi
2302724
2302724@sit.singaporetech.edu.sg

Wong Khin Foong
2302728
2302728@sit.singaporetech.edu.sg

Abdul Haliq Bin Abdul Rahim
2302747
2302747@sit.singaporetech.edu.sg

***Abstract*— Through data and text analytics enhanced by Natural Language Processing and AI technologies, this study addresses the growing problem of Telecommunication Network Fraud, notably online employment frauds, in Singapore. The study makes use of the Employment Scam Dataset (EMSCAD) and thorough preparation to address issues like class imbalance and null data. By extracting and analysing data from jobstreet.com.sg using a trained machine learning model, the integration of the online application Sure-Bo helps users detect probable job frauds. The LIME explanation improves explainability of the findings, which provide insights into fraud trends, critical variables, and potential red flags. The goal of the study is to provide people with the analytical abilities they need to spot false job offers quickly.**

***Keywords*— Job Scam, Machine Learning, Data Analysis, NLP**

## I. Introduction

The prevalence and sophistication of scams have increased significantly in today's technology-driven era [3] In Singapore, the issue of Telecommunication (Telecom) Network Fraud has become a matter of growing concern due to the rising number of reported cases. In 2021, there were 23,933 reported scams, which surged to 31,728 in 2022, marking a 32.6% increase compared to the previous year. These scams resulted in monetary losses totalling $660.7 million in 2022 [4]. Online job scams have gained notoriety, affected 6,494 victims, and led to losses exceeding $117.4 million by December 2022 [2].

An online job scam is a fraudulent scheme designed to deceive job seekers by offering false employment opportunities or promising high-paying positions. Scammers create enticing job postings on various platforms, such as online job posting forums, social media, and websites, which often appear legitimate and promise attractive salaries, flexible working conditions, and career growth.

To address this issue, we intend to employ data and text analytics techniques, supported by AI technologies like Natural Language Processing (NLP). Our approach involves using the Employment Scam Dataset from EMSCAD, which provides insights into common tactics used in job scams. This effort aims to equip individuals with the analytical skills needed to identify and mitigate these deceptive job offers.

## II. Related works

Fraud detection systems have a well-established history in the tech world, ranging from simple email spam classifiers to cutting-edge AI-powered anti-fraud systems developed by banking institutions in Singapore [5]. In the following section, we will discuss studies that address unstructured data, data preprocessing, data analytics, and various modelling techniques.

### A. Email Spam Classification

The content of an email is quite similar to the job descriptions posted online in terms of length and the lexical intensity of the words used. Kumar and his colleagues in their 2020 paper executed a series of preprocessing techniques, including the removal of stop words to elevate the significance of higher-level vocabulary during the modelling process, tokenization to split the text into corpus and tokens, and the utilisation of the Bag of Words approach for feature extraction. They explored various modelling techniques and observed that the Naïve Bayes algorithm achieved the highest F1 score after hyperparameter tuning [6].

### B. *Fake News Detection*

Due to its “free-to-write” nature, fake news on social media is incomplete, unstructured, and noisy. Similarly, it is often in the form of yellow-journalism, which represents those articles that do not contain well-researched news, but instead rely on eye-catching headlines with a propensity for exaggeration, sensationalization, scare-mongering, etc. [8]. This is like the structure of fake job descriptions which are often in point form or short text corpus and utilizes eye catching titles and attractive compensations. Abdullah-All-Tanvir and team’s 2019 paper on “Detecting Fake News using Machine Learning and Deep Learning Algorithms”, demonstrated that SVM and Naïve Bayes algorithm both achieved a promising F1 score of 94% with TF-IDF feature [7].

### C. *Fake Job Recruitment Detection*

With the increase of people’s reliance on the internet, many companies have resorted to posting job recruitment notices online, allowing for ease of access by job seekers around the globe. This allows an opportunity for scammers to prey on these job seekers, using fraudulent activity they aim to damage the reputation of companies and earn a quick buck simultaneously [1]. With the purpose of denying these frauds, Shawni Dutta and Professor Samir Kumar Bandyopadhyay applied a machine learning approach to recognize these fake job posts. Using a classification tool, they aimed to isolate fake job posts from a larger set of job posts and alert the user. Many different types of classifiers were used to identify which classifier would provide the best results pertaining to this mission. Some of these classifiers include the Naïve Bayes, AdaBoost, Gradient Boosting and Random Forest Classifier. Utilising different types of classifiers, they managed to achieve an accuracy rate of 98.27% with the Random Forest Classifier which was much higher than existing methods used in 2020.

However, we believe that the data pre-processing steps performed in the works referenced above can be improved. Similarly, their model is trained using a more general dataset, which may not fit well into Singapore's context. Therefore, we aim to provide a solution that is more relevant for the context of Singapore.

## III. METHODOLOGY

### A. *System Overview*

Sure-Bo can be categorised into 2 core components, ML Core and Sure-Bo UI. The ML Core consists of all the machine learning and preprocessing logic put in place to retrieve the pre-trained model, preprocess, and perform inference of the data. This mainly includes the data preprocessing worker, model inference worker.

On the other hand, the Sure-Bo UI will provide the users with a user interface. Currently, it will allow users to input job descriptions that they find suspicious, and it will return the likelihood if it is a fraudulent job posting. For advanced users, we enabled the option to allow them to choose different models to use for inference. The user will also be able to identify the model explainability.

### B. *Dataset Description*

The dataset being utilized is the Employment Scam Aegean Dataset (EMSCAD). It is publicly available and contains 17,880 real life job ads from 2012 to 2014. The purpose of the dataset is to provide a clear visualization on the issue pertaining to Employment Scams. It provides researchers working within the field a valuable resource to utilise for their research. The records within the dataset were manually classified into 2 different categories, leaving us with 17,014 real job ads and 866 fake job ads.

Tables A1 and A2, found in appendix A, provide comprehensive information on the data types, examples, and row counts for each feature and label column within EMSCAD. Upon careful examination, we identified several issues with this dataset. These include challenges related to class imbalance, an abundance of null values, the presence of stringified Boolean values, HTML tags, and escape characters. In the subsequent discussion, we will delve deeper into these challenges and outline mitigation steps that we intend to implement for resolution.

*Class Imbalance Issue* — The EMSCAD dataset is structured as a binary class dataset, with labels categorised as either “t” or “f”. However, a significant challenge arises from a class imbalance issue within the dataset. Specifically, there are only 866 instances of fraudulent cases out of a total of 17,880 rows in EMSCAD. This imbalance poses a notable concern, as it increases the risk of misclassifying fraudulent cases due to the limited representation of this class in the dataset. Given that the primary objective of our machine learning model is to prevent individuals from falling victim to job scams, a high true-negative rate is unacceptable. Nevertheless, the challenge of class imbalance is not new, and various approaches exist to overcome it. In our analysis, we will predominantly explore three strategies: data resampling, Synthetic Minority Over-sampling Technique (SMOTE), and ensemble learning.

*Null Values Handling* — Image 1 in the appendix illustrates the count of null values for columns with more than one null entry. Certain columns, like "salary_range" and "department," show a high prevalence of null values, with 83% and 64%, respectively. Conversely, columns such as "location" demonstrate a mere 1% of null values. This significant difference emphasises the importance of being careful in how we deal with missing values. Using the same method for all features wouldn't work well because the null values are spread out differently across each column. While we could easily drop rows with null values in the "location" column, it's not that straightforward for the others. Interestingly, these null values turn out to be quite helpful. They signal missing information in job descriptions, which, more often than not, indicates a potential scam job posting. To make the most of this, we plan to transform these columns into Boolean columns to flag whether there's missing data.

*Boolean String* — Boolean strings such 't' and 'f' help us to easily classify data within the columns that utilise them. The column "fraudulent" is one such relevant example. As Boolean strings are unreadable by the system, we alter them into binary, having all 't' values be equivalent to '1' and all 'f' values be equivalent to '0'. By altering the data to binary, we are able to easily categorise each row in the dataset as a fraudulent or legitimate job posting. This gives our machine learning model a direct way to determine if a job posting should be classified as a fraud.

*HTML Tags and escape characters* — We remove HTML Tags and escape characters as they interfere with a few concepts in our analysis. Firstly, it is important to ensure that all our data is uniform. By keeping a clean and consistent format, we can standardise the text, allowing our analysis to be based on actual content and not the formatting. It also assists in the readability of the data, with HTML tags, it can make the data difficult to read and in turn interfere with our analysis. Removing escape characters helps us to prevent possible encoding issues, ensuring proper representation of data in a consistent and accurate manner.

### C. Systems Design

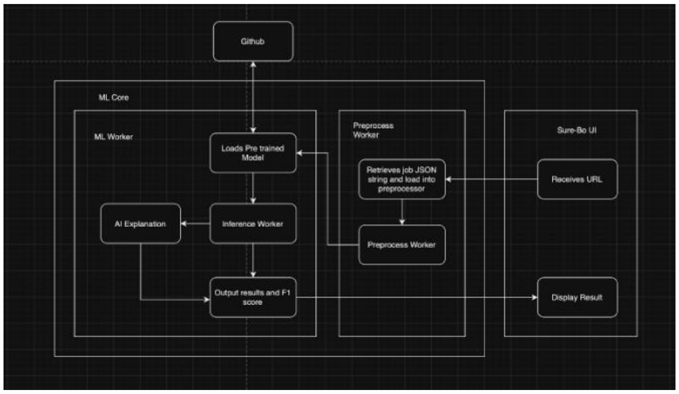


Fig. 1. General system design structure.

### D. Data Pre-processing

The dataset will then undergo a series of processing that will then allow us to migrate from raw data into a more integrated, filtered, and augmented dataset that will then allow us to conduct proper and accurate analysis and achieve our objectives. Primarily we will be using the pandas library to process the data.

*Remove Null Values* — The dataset contains null values across several columns. We have addressed this issue by removing null values specifically in the "location" column. Additionally, for other columns with null values, we have introduced new columns to quantify the occurrence of null values in each respective column.

*Convert stringified Booleans* — The dataset comprises Boolean columns [telecommuting, has_company_logo, has_questions, fraudulent, in_balanced_dataset] represented by string types ["t", "f"]. However, conventional practice dictates that Boolean data should be in either Boolean or integer type. To facilitate subsequent analysis and modelling, I converted the data into integer type [1, 0].

*Remove all non-alphabetical characters* — The dataset contains non-alphabetical characters such as symbols, numbers, punctuation marks across several columns (title, salary_range, location etc). By removing these characters, we can clean and prepare text data for further analysis or natural language processing tasks.

*Change all words to lowercase* — We then proceed to change all words in the dataset to lowercase as it can provide us with a foundation of consistency and normality for subsequent analysis. By ensuring that all words are uniformly represented, this "data pre-processing" step enhances the accuracy and reliability of various strings (words) which can contribute significantly to the overall quality of our data analysis and research.

*Tokenize the text corpus and filter short words* — As we engage in text analysis, we will tokenize words into unigrams. Additionally, we'll exclude words with less than 3 characters, as they're likely less useful for our analysis. This refinement will result in a clearer word cloud, enhancing the overall quality of our analysis.

*Count the Null values for each category* — Lastly, counting the null values for each category in a dataset is an essential step in data analysis and preprocessing. Because some columns (department, salary_range, benefits etc.) contain null values, we aren't sure if these means jobseekers didn't indicate their answer for that or if it could be a sign of fraud hence why we count the null values for each category based on the fraudulent columns before we proceed with the data analysis.

### E. Data analysis and algorithms design

*Matplotlib* — The figures shown in our visualization of data often used matplotlib. Being a comprehensive library for creating various visualizations in python, it enables us to generate graphs and pie charts to display our data clearly. We can even filter out the data such that only the most relevant data is displayed in our diagrams.

*Word Cloud* — Word cloud, a visual representation showcasing the most prominent words in text, is instrumental in deriving insights from our dataset rich in textual data. Whether discerning common words within job descriptions or capturing the overall sentiment, we utilize word clouds. Furthermore, when combined with tools like part-of-speech tags (pos-tags), this approach refines our analysis, allowing for a more focused examination of diverse job descriptions.

## IV. Results and Insights

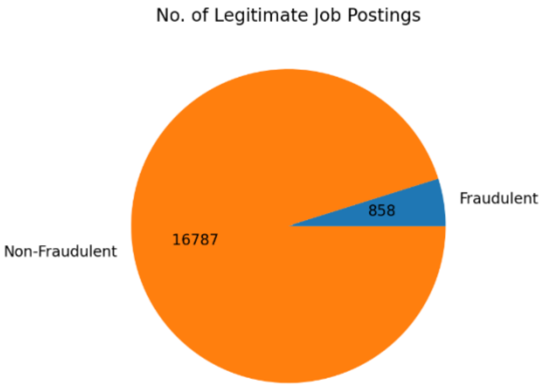


Fig. 2. Number of Legitimate Job Postings

Visualisation of the number of Fraudulent and Legitimate Job Postings within the dataset. From figure 2, it can be observed that there are 858 fraudulent and 16787 legitimate job postings within the dataset. Resulting in a 4.86% fraudulent job percentage in the dataset.

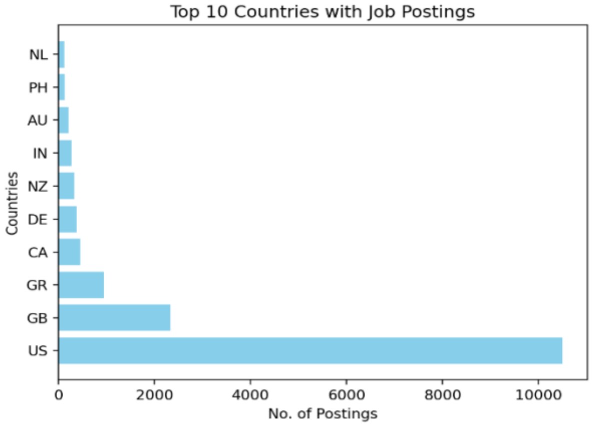


Fig. 3. Top 10 Countries with Job Postings.

As displayed in Figure 3, the top 10 countries with job postings in this dataset are the US, GB, GR, CA, DE, NZ, IN, AU, PH and NL respectively. The dataset contains a large majority of data from the US with 10495 data entries. The next highest being 2336 job postings from GB, followed by 939 postings from GR.

TABLE 1

| NUMBER OF DATA IN EACH JOB CATEGORY | |
|---|---|
| **Job Category** | **No. of Data** |
| Other | 9835 |
| IT | 4276 |
| Business | 2746 |
| Engineering | 1637 |
| Educator | 848 |
| Human Resource | 493 |
| Manufacturing | 291 |
| Construction | 228 |

From the table above, we can conclude the top job category based on our category mapping is IT. Followed by Business and Engineering making the bulk of our dataset.

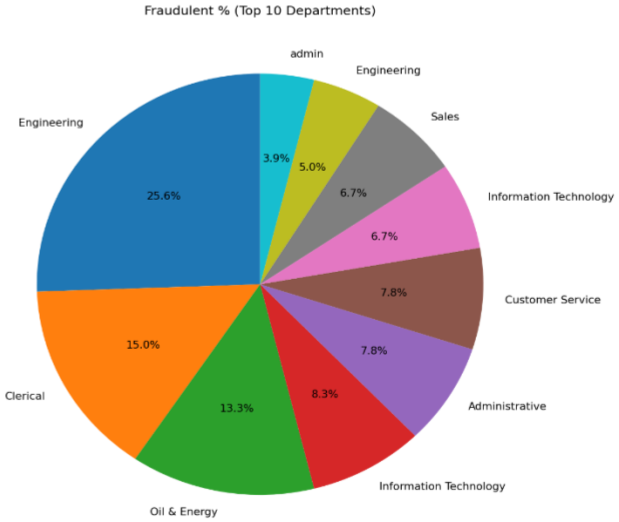


Fig. 4. Percentage of Top 10 Fraudulent Jobs by Department.

Figure 4 shows the top 10 departments with the most fraudulent job postings. The top 3 departments, according to the pie chart, are Engineering (25.6%), Clerical (15%), and Oil & Energy (13.3%). Although the dataset does not explain why or how people from these 3 departments become victims of fraud, it is safe to assume that it may be because the jobs are lucrative enough to easily lure victims into signing up for these fake job offers. These results also showcase a mix of white- and blue-collar jobs and could indicate that fraudsters are targeting a wide range of jobseekers, hence they should be especially wary of fraudulent job postings in these departments.

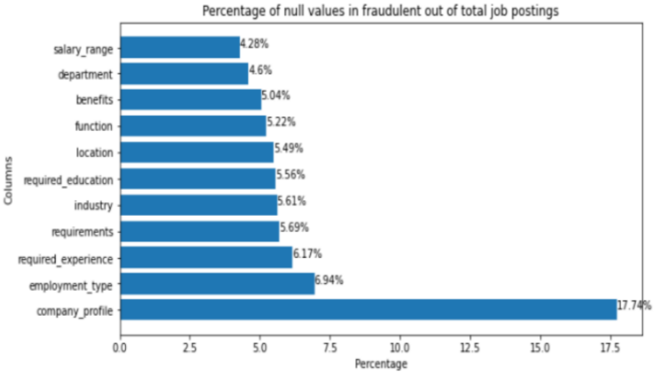


Fig. 5. Bar chart representing the percentage of null values that fraudulent data has compared to the total null values within the dataset.

Figure 5 illustrates that fraudulent job postings often lack information in these top three fields: company profile, employment type, and experience requirements. Notably, the company profile field emerges as the most frequently omitted detail among these fields. The prevalence of null values in these key aspects suggests a pattern associated with fraudulent job postings, and recognizing these patterns facilitates the identification of potentially fraudulent job postings, offering a primary tool in distinguishing legitimate opportunities from dubious ones.

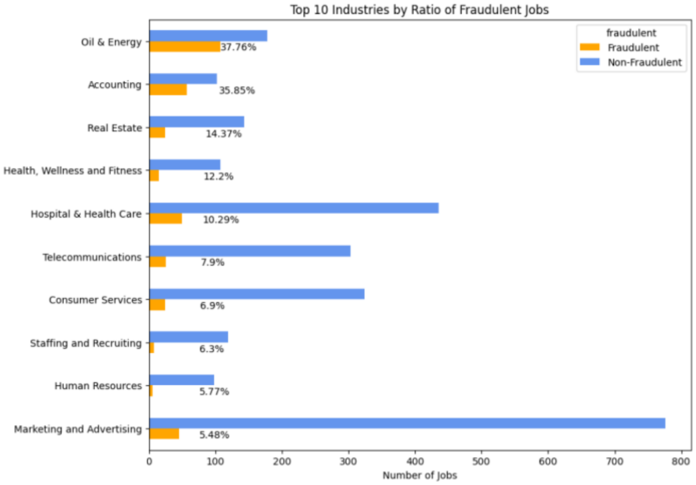


Fig. 6. Top 10 Industries by Ratio of Fraudulency

In figure 6, we can conduct further analysis on our dataset, by identifying the industries with the highest ratio of fraudulent jobs. It is also noted that we only highlight industries with samples data greater than 100, as it allows us to have a fair degree of accuracy in justifying our predictions. This then can allow us to have a brief overview of the types of industries that have a high fraudulency rate. Allowing us to ascertain potential fraudulent jobs. Based on our dataset, industries in Oil & Energy and Account have the highest ratio of fraudulent jobs at 37.76% and 35.85% respectively. This data further supports the claim in Figure 4, and that jobseekers should be more wary in these fields.

Another point of note is within the Marketing and Advertising industry, where we see a significant ratio of fraudulent jobs whilst maintaining one of the relatively largest proportion of data with regards to specific industries. As a result, we can note that spheres within Marketing and Advertising are very saturated but still retains a relative significance in its ratio of fraudulency, possible factors such as the general work scope for such jobs may be more manageable and accessible by most persons and the job plays a major role in almost any company suggest why such an industry may be highly saturated and job scams may be more prevalent. In such cases, our users should be vigilant in pursuing such industries.

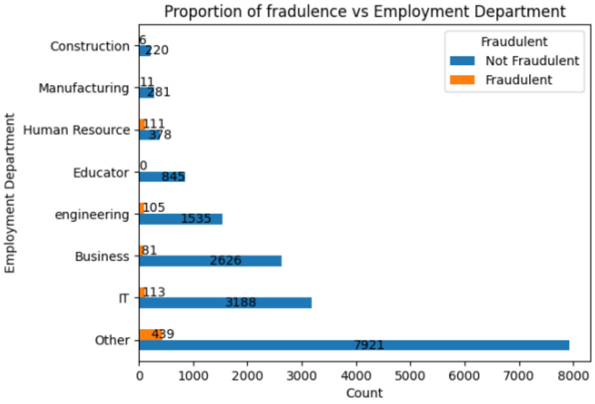


Fig. 7. Proportion of Fraudulent Jobs in each department.

Figure 7 displays the proportion of fraudulent jobs in each department. One insight that we can achieve with the data is by referencing the categorical variables of our job posting where we have the common job categories taken from Job Street, to have a rough idea of the different job sectors available. In such scenarios we can depict the type of job position against the likelihood of whether it is more likely or less likely to be a scam. In such cases, our users can be more aware and cautious of the job offers they receive.

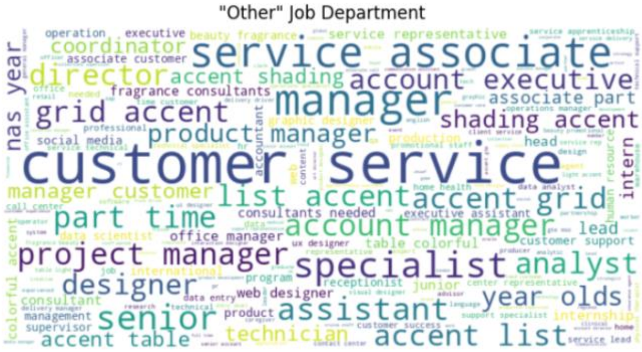


Fig. 8. Word cloud on job title under others

As seen in Figure 7, the ‘Other’ in job title has the highest amount of data compared to the rest. Hence, a word cloud, represented in Figure 8, is used to understand the frequent appear words that classifies the title to be in “Other”.

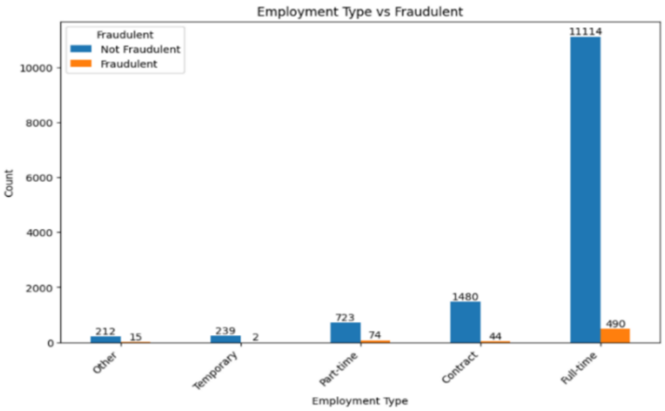


Fig. 9. Proportion of Fraudulence vs Type of Jobs

The reason behind this cluster bar graph (Figure 9) is for us to depict that employment type shows the likelihood of fraudulence. It is common to think that usually the temporary and contract employment type will lead to the job posting to be fraudulent. To prove the misconception, the bar graph shown above shows that full-time employment type is most likely to be fraudulent compared to the others with 480 number of fraudulent data. This illustrates that we should not base fraudulency on stereotypical knowledge and be vigilant in such cases when considering a job application based on the type of position.

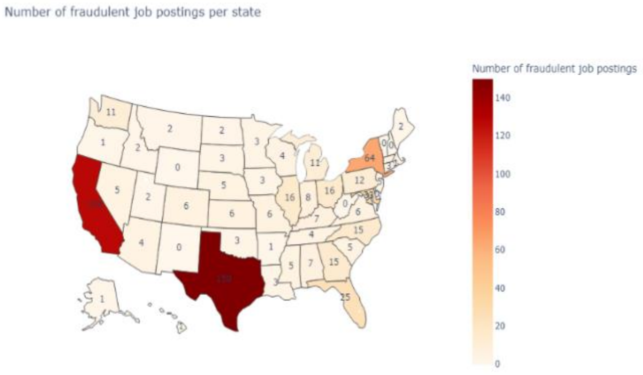

Number of fraudulent job postings per state
Number of fraudulent job postings
140
120
100
80
60
40
20
0

Fig. 10. Distribution of fraudulent jobs postings per American state.

Figure 10 indicates that Texas, California, and New York are the primary locations featured in fraudulent job postings. These states serve as significant economic hubs with abundant job opportunities, making them attractive to scammers. The substantial number of job seekers drawn to these areas creates an environment where scammers target potential victims by strategically incorporating these prime locations into their deceptive job postings.

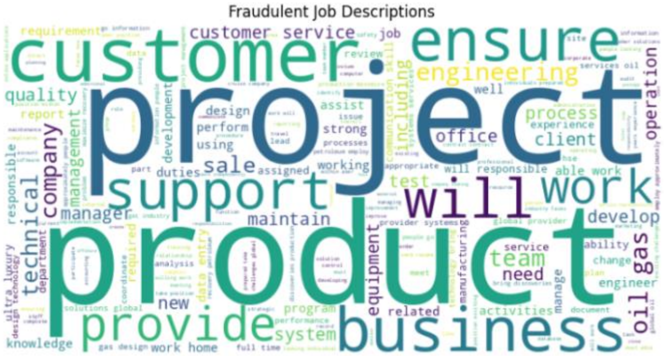


Fig. 11. Word Cloud of Fraudulent Job Descriptions.

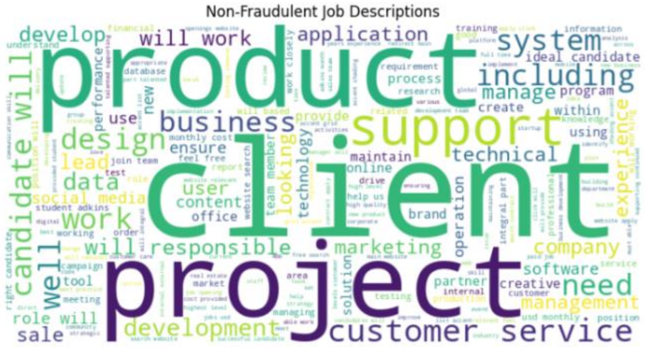


Fig. 12. Word Cloud of Non-Fraudulent Job Descriptions.

Further exploring job posting, as mentioned above, job posting descriptions place a huge part as the determining factor whether it is indeed fraudulent or not. To visualise the fraudulency of a job posting, the Word Cloud is used to assist us in visualising what are the most occurring words used in the non-fraudulent and fraudulent job listings. With reference to Figure 11 which shows the Word Cloud for fraudulent job descriptions, the most occurring words are “customer”, “project” and “product”.

With reference to Figure 12 which shows the Word Cloud for non-fraudulent job descriptions, the most occurring words are “product”, “client” and “project”. We can conclude that, in this case it is difficult to differentiate the real and fake job postings. However, two key factors that we can conclude is fraudulent job postings are rather vague where they do not mention the specific role unlike non-fraudulent, they are more detailed, mentions the job title with the relevant descriptions. For the second would be the mentioning of “work from home” in fraudulent job posting.

## V. Modelling and Integrating

We will create a pre-trained ML model using the EMSCAD dataset and integrate it with Sure-Bo UI to assist our user with identifying potential job scams. For demo purposes, we will only be providing support for jobs listed on *jobstreet.com.sg.*

### A. *Preprocessing*

As the EMSCAD dataset comes pre-labelled, eliminating the need for manual labelling, we directly employ a supervised learning approach. Our features are derived by combining the *description*, *requirements*, and *benefits* columns, while the *fraudulent* column serves as the label. Given that the combined feature column comprises unstructured data, we execute the following text preprocessing steps to ready it for modelling:

*Extract text from HTML wrappers* — All text is encapsulated within HTML tags, which introduce additional noise to our model if not removed. To address this, we employ Beautiful-Soup's HTML parser to extract the text.

*Remove all non-alphabetic characters* — Non-alphabetic characters, such as punctuation, numbers, and special characters, are irrelevant to our model. Hence, we employ methods like String.isalpha() and String.isspace() to remove them.

*Tokenise and remove stopwords* — The text is tokenized into unigrams, and stop words are removed using the NLTK library. This step helps eliminate noise in the model.

*Stemming* — All words are stemmed to their root form, reducing the model's dimensionality, and creating a more consistent and concise representation of the text data, thereby contributing to enhanced model performance.

*Vectorisation* — Following basic text preprocessing, we transform the data into structured form using the TF-IDF algorithm. This method considers not only captures word frequency within a document but also the importance of words across a collection of documents, aiding in the extraction of crucial features.

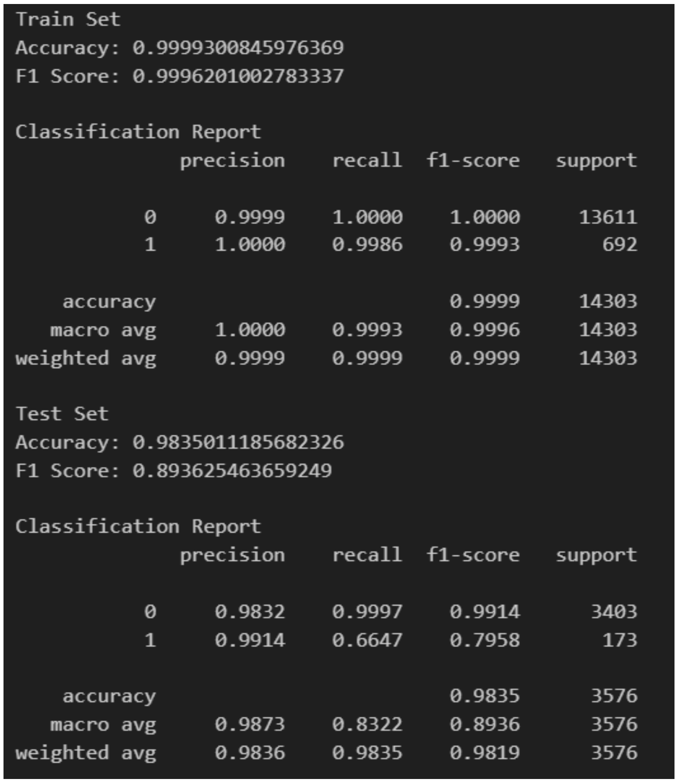
*Resample all classes* — Addressing the heavy imbalance in this dataset (with only 4.8% labelled as fraudulent), we utilize the SMOTE Tomek algorithm from imblearn to resample our classes. This results in a 6% improvement in our F1 score, enhancing recall compared to not resampling.

*Normalisation* — Features are normalized using sklearn's normalizer, a common operation for text classification. L2-normalized TF-IDF vectors allow for cosine similarity computation, a base similarity metric for the Vector Space Model commonly used in Information Retrieval.

*Train test split* — Data is divided into training and testing sets with an 80:20 ratio.

### B. *Training*

We explored several modelling techniques and algorithms such as the Naïve Bayse algorithm, as well as ensemble learning algorithms such as Gradient Boosting and random forest, hoping that we can overcome the class imbalance issue through machine learning algorithms. We observed that the random forest algorithm outperformed the other two algorithms mentioned above with a F1 score of 89% on the test data.



```
Train Set
Accuracy: 0.9999300845976369
F1 Score: 0.9996201002783337

Classification Report
              precision    recall  f1-score   support

           0     0.9999    1.0000    1.0000     13611
           1     1.0000    0.9986    0.9993       692

    accuracy                         0.9999     14303
   macro avg     1.0000    0.9993    0.9996     14303
weighted avg     0.9999    0.9999    0.9999     14303

Test Set
Accuracy: 0.9835011185682326
F1 Score: 0.893625463659249

Classification Report
              precision    recall  f1-score   support

           0     0.9832    0.9997    0.9914      3403
           1     0.9914    0.6647    0.7958       173

    accuracy                         0.9835      3576
   macro avg     0.9873    0.8322    0.8936      3576
weighted avg     0.9836    0.9835    0.9819      3576
```

Fig. 13. Performance report on training using random forest classifier.

In Figure 13, the performance report for the random forest classifier is presented. Despite its superior performance compared to the other two algorithms, a challenge persists with an imbalanced dataset. This is evident in the model's notably high precision but relatively low recall on the test dataset. Nevertheless, given the constraints of time and the specific focus of this paper, we opt to utilize this model as the baseline for our application.

### C. *Integration and Inference*

In order to enhance our users' experience with our model and provide them with effective tools to identify and steer clear of online job scams, we integrated our machine learning model with into a web application that we created. The entire process is outlined in three distinct phases, each detailed below:

*Web Application* — We've crafted a single-page web application known as Sure-Bo, optimizing its functionality by incorporating Python Flask as the backend server to seamlessly integrate our machine learning model. Furthermore, we harnessed the power of Jinja templates together with jQuery to dynamically load our data onto the frontend application.

*Scrapping* — For demonstration purposes, we've currently enabled scraping exclusively for data sourced from jobstreet.com.sg. Users have the flexibility to paste links from JobStreet that they find suspicious directly into Sure-Bo. Upon doing so, our system will conduct web scraping on the provided URL, extracting the pertinent job data, and subsequently passing it for inference through our machine learning model. This functionality offers users a straightforward way to assess the legitimacy of job postings on JobStreet within the Sure-Bo platform.

We employed the BeautifulSoup library for our web scraping task. Differs from common practices that often involve scraping based on raw HTML tags or utilizing XPath, our approach involves capturing the raw JSON data that the JobStreet server returns to the client for a given URL. This method provides a more direct and efficient means of extracting relevant information, allowing us to seamlessly integrate the retrieved data into our application.

*Inference* — The data is subsequently channelled into the preprocessor worker, where the feature columns undergo cleaning and vectorization. Following this preprocessing step, the data is transmitted to the model for inference. However, a crucial preliminary step is taken: the server retrieves the model from GitHub, downloading it locally in preparation for the inference task.

This pre-emptive download is necessary due to GitHub's constraints on file upload size. In light of this limitation, we've adopted a strategy wherein the model is dynamically loaded into the local file path before inference can take place. This approach ensures that the application functions seamlessly, even without the need for hosting. Given that the application currently can only be ran locally, users are prompted to either dynamically load the model or manually download it, ensuring a smooth and efficient execution of the application.

### D. Explainability

In our commitment to bolster AI ethics and governance, we've integrated explainability features into our application through the use of the LIME explainer. Upon displaying the model's prediction, our application goes a step further by presenting the corresponding text that contributed to the prediction made by the model through analysing the feature importance within the inference text.

This not only empowers users to verify the prediction, mitigating the opacity often associated with machine learning models, but also serves as an educational tool. Users gain insights into the keyword trends exploited by scammers within fake job postings. By providing this contextual information, our application ensures transparency, making it more than just a black-box tool. This educational aspect is designed to equip users with a heightened awareness of potential red flags in job postings, fostering vigilance even in situations where direct access to our application might not be available.

## VI. Conclusion

In summary, our project delves into the pressing concern of Telecommunication Network Fraud in Singapore, with a specific focus on addressing the surge of online job scams. By harnessing advanced data and text analytics, complemented by AI technologies like Natural Language Processing (NLP), we developed Sure-Bo — a web application aimed at empowering individuals to avoid potentially deceitful job postings. Overcoming challenges embedded in the EMSCAD dataset, including class imbalance and null values, required the application of sophisticated techniques like SMOTE and data resampling. Our thorough data analysis provided valuable insights into the prevalence of scams across diverse job categories, departments, and geographical locations.

The machine learning model, employing the random forest algorithm, demonstrated promising outcomes with an F1 score of 89%. The integration of Sure-Bo with explainability features not only ensures transparency in predictions but also serves an educational purpose, enlightening users on the subtle indicators of fraudulent job postings. Looking ahead, potential areas for future exploration include refining data preprocessing steps in improving class imbalance issue, explore other models, and extending platform support for a broader societal impact.

In essence, if something seems too good to be true, then it probably is. There are cases where most would say only to look out for jobs that are contract or temporary because they tend to be fraudulent. However, through our analysis, shows to pay attention to full time jobs. Also, be aware of job descriptions that are rather vague in their requirements. The ratio of fraudulent to non- fraudulent job postings is rather large, which results to unfair comparisons. Therefore, it’s imperative for individuals to conduct through research, verify the legitimacy of the employer’s profile and job description. By adopting a discerning approach, jobseekers can prevent themselves from becoming a victim against such fraudulent schemes.

## Acknowledgment

We extend our gratitude to every team member for their crucial contributions to this project. It is essential to recognize that the collaboration and combined efforts of the team were instrumental in overcoming challenges and achieving our desired deliverable.

# References

[1] Dutta, S., & Bandyopadhyay, S. K. (2020, April). Fake Job Recruitment Detection Using Machine Learning Approach. International Journal of Engineering Trends and Technology. Retrieved September 11, 2023, from https://ijettjournal.org/archive/ijett-v68i4p209s

[2] National Crime Prevention Council. (n.d.). Job Scam. Scam Alert. Retrieved September 8, 2023, from https://www.scamalert.sg/scam-details/job-scam

[3] Ng, J., & Ng, R. (2022, December 15). Commentary: As scams get more sophisticated, young and digitally savvy individuals are more likely to fall prey. CNA. https://www.channelnewsasia.com/commentary/pig-butchering-scam-love-investment-crypto-crime-prevent-3143866

[4] Singapore Police Force. (2023, February 8). Annual Scams and Cybercrime Brief 2022. Singapore Police Force. Retrieved September 11, 2023, from https://www.police.gov.sg/-/media/Spf/PNR/2023/Feb/Police-News-Release---Annual-Scams-and-Cybercrime-Brief-2022.ashx

[5] Sun, D. (2021, April 6). Bank fraud experts in S'pore use AI to predict scammers' next move. The Straits Times. https://www.straitstimes.com/tech/tech-news/anti-fraud-experts-use-ai-to-predict-cheaters-next-move

[6] N. Kumar, S. Sonowal and Nishant, "Email Spam Detection Using Machine Learning Algorithms," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 108-113, doi: 10.1109/ICIRCA48905.2020.9183098

[7] Abdullah-All-Tanvir, E. M. Mahir, S. Akhter and M. R. Huq, "Detecting Fake News using Machine Learning and Deep Learning Algorithms," 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, 2019, pp. 1-5, doi: 10.1109/ICSCC.2019.8843612

[8] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. SIGKDD Explor. Newsl. 19, 1 (June 2017), 22–36. https://doi.org/10.1145/3137597.3137600