

Fake Job Post Prediction Using Data Mining

Shri Udayshankar B¹, Veeraj R Singh², Sampras P³, Aryan Dhage⁴

¹Professor, Department of Computer Science, PDA College of Engineering, Kalaburagi, India.

²Student, Department of Computer Science PDA College Of Engineering, Kalaburagi, India.
veeraj796@gmail.com

³Student, Department of Computer Science PDA College Of Engineering, Kalaburagi, India.
sampraspradeep@gmail.com

⁴Student, Department of Computer Science PDA College Of Engineering, Kalaburagi, India.
rndhage168@gmail.com

ABSTRACT

The proliferation of online job boards is a testament to the ease with which new positions may be publicized in today's connected society. As a result, the problem of predicting fraudulent job postings will be of paramount importance. Predicting the outcome of a bogus job posting is a challenging classification assignment, similar to many others. In order to determine if a job posting is genuine or not, this study proposes using a variety of data mining methods and classification algorithms, including KNN, decision tree, support vector machine, naive bayes classifier, random forest classifier, multilayer perceptron, and deep neural network. The Employment Scam Aegean Dataset (EMSCAD) was used for our experiments; it consists of 18000 data points. Using a deep neural network as a classifier yields excellent results in this setting. This classifier is a deep neural network with three thick layers. Classification accuracy (DNN) for identifying fake job postings is roughly 98% thanks to the trained classifier.

Keywords- Fake job, Machine learning, Python

I. INTRODUCTION

As a result of technological and industrial advancements, more and more types of jobs are available to people today. Advertisements for these openings allow prospective employees to research their possibilities in light of their available time, skills, experience, and other factors. The internet and social media have become significant factors in the recruitment process. Given that advertising is crucial to the success of any recruiting process, the influence of social media in this area is substantial. Opportunities to disseminate employment information are multiplying with the rise of social media and online advertising. However, as the number of channels via which job announcements may be shared has expanded rapidly, so too has the number of fraudulent job advertisements that irritate applicants. Therefore, individuals are less likely to respond to job advertising because they want to ensure the integrity of their personal, academic, and professional records. As a result, it will be very difficult for the genuine purpose of legitimate job posts through social and electronic media to gain people's trust and dependability. Technologies exist to improve and simplify our lives, but they should not be used in a way that makes our professional lives less safe. It would be a huge step forward in the recruitment process if job postings could be screened in such a way that bogus job postings could be predicted. False job postings make it more difficult for genuine job seekers to obtain suitable positions, which is a waste of everyone's time. The advent of an automated method to foresee fake job postings presents a fresh opportunity to address challenges in HRM.

1.1 Problem Statement

An application built on machine learning-based categorization algorithms is offered in the project as a means of preventing fraudulent online job postings. Several classifiers are employed to analyze potentially fraudulent online job postings, and their findings are compared so that the most effective classifier may be chosen. It's useful for weeding out the many hoaxes among the thousands of job postings.

1.2 Objectives

Gaining valuable insights from the job posting by determining if the postings are genuine or fraudulent on the basis of an appropriate model and collecting this data.

II. SYSTEM ANALYSIS

Existing System

In order to identify potential instances of fraud inside an online recruiting system, Alghamdi [2] et al. devised a model. The team performed several tests using a machine learning algorithm on the EMSCAD dataset. Data preprocessing, feature selection, and fraud detection using a classifier were the three phases of their work on this dataset. The basic text pattern was maintained after the preprocessing phase of removing noise and html elements from the data. To streamline the process, they used a feature selection approach to narrow down the variables. To identify fraudulent job postings in the validation set, a Support Vector Machine was utilized to pick features for an ensemble classifier built using random forest. The random forest classifier was thought to be a tree-based classifier that served as an ensemble classifier by using a majority voting approach. This classifier has a 97.4% accuracy rate in identifying bogus job postings.

Proposed System

Our work dataset is used to train the K-Nearest Neighbor, Random Forest, Decision Tree, Naive Bayes, RBF kernel, and MLP classifiers, as well as the Support Vector Machine and Multilayer Perceptron. EMSCAD has been used to identify bogus job postings in datasets. There are 18 characteristics per row in this data set (including the class label) and there are 18000 samples total. These include: has_company_logo, has_questions, employment_type, required_experience, required_education, industry, function, fraudulent (class label), and job_id. Other details include location, department, pay range, company profile, requirements, benefits, telecom, and a has_company_logo and/or has_questions attribute. Only 7 of the 18 qualities were utilized after being transformed into categories. Categories are created from text values for telework, has_company_logo, has_questions, employment_type, necessary experience, required education, and fraudulent. For instance, the value of "employment_type" may be changed from 0 to 1 to indicate "none," 2 to indicate "part-time," 3 to indicate "others," 4 to indicate "contract," and 5 to indicate "temporary." The primary motivation for making this transformation is to categorize fake job postings into appropriate categories without resorting to text processing or natural language processing. We have relied only on those kind of characteristics in our investigation.

Feasibility Study

Preliminary research looks at whether or not the project is doable and whether or not the system will be valuable to the business. The primary goal of the feasibility study is to examine the technical, operational, and financial viability of implementing new modules and fixing bugs in existing, operational systems. If there had limitless time and materials, any system would be doable. The feasibility analysis is a managerial task. A feasibility study is conducted to determine the viability of an information system project and to provide viable alternatives.

In the scope of the preliminary examination, the feasibility study includes:

- ✓ Technical Feasibility
- ✓ Operational Feasibility
- ✓ Economical Feasibility

Operational Feasibility

It is the likelihood that the product will function as intended. Some goods may function perfectly in the lab but fall short when put to the test in the real world. Additional human resources and their Technical knowledge are analyzed. Projecting whether the system will be utilized once created and deployed is reliant on the availability of human resources for the project. The degree to which a proposed system addresses the issues and seizes the possibilities indicated during scope definition and the extent to which it meets the criteria determined during analysis. The report analyzes whether the company is ready to back the new system. This viability must be established by assessing the level of managerial support for the proposed initiative.

Technical Feasibility

This question asks whether the current application is fully supported by commercially-available software. It analyzes the benefits and drawbacks of implementing a certain piece of software into the development process. It also investigates the supplementary instruction required of the workforce for the application to succeed. Next, the organization's technical capacity is measured against the technical needs. If the organization's current level of technical expertise is enough to meet the needs of the systems project, then it is regarded technically viable. The analyst is tasked with determining whether the requested enhancements or additions can be made to the existing technological resources.

Economic Feasibility

It's a measure of how much money we're making off of the product relative to how much we're putting into it. It is not practical to build the product if it is functionally equivalent to the previous system. Cost-benefit analysis is another name for economic analysis. The efficacy of a new system is often measured using this technique. The standard practice in economic analysis is to weigh the predicted savings and benefits against the total cost of a potential system. The choice to develop and deploy the system is taken if the advantages are greater than the expenses. An entrepreneur has to carefully consider the potential outcomes before making any decisions.

III. SYSTEM ENVIRONMENT

Hardware Requirement

❖ Processor	: Pentium CoreI5 and Higher
❖ RAM	: 4GB or more.
❖ Hard Disk	: 500GB or more.
❖ Monitor	: 15 inch Color Monitor
❖ Keyboard	: 102/104 Keys
❖ Mouse	: Optical Mouse

Software Requirement

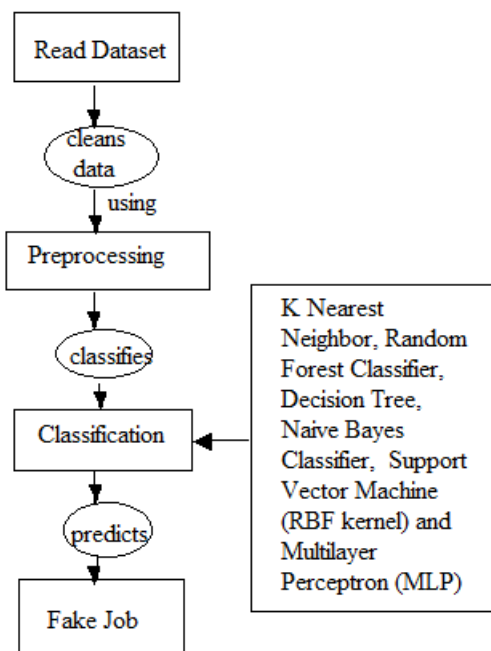
❖ Operating System	: Windows 10
❖ Front End	: Python
❖ Back End	: SQLite3

Module Description

- **Collecting Dataset:**
All of the fields in the data set that will be used as input up until the very last one will be referred to as features, while the final cell will be referred to as labels and will be formatted as a 0 or 1.
- **Data set processing:**
Since the provided data is not in the correct format, we must first clean it up so that we may utilize the remaining fields as features and convert the data to a scalar format.
- **Pre-processing:**
After transforming the input into a scalar format, we generate fresh features to feed into the algorithm; these are then stored in x, while the labels are recorded in y.
- **Algorithm Fit:**
This phase involves importing the trained model into the system, which includes fitting the features and labels of the training data to the algorithm.
- **Prediction:**
Prediction is made based on information provided in the form of a comma-separated values (csv) file including specifics on several profiles.

IV. SYSTEM DESIGN

4.1 Data Flow Diagram (DFD)



Introduction

The design phase's goal is to establish a strategy for resolving the issue described in the requirements papers. This is the first stage of transitioning from the domain of the issue to the domain of the solution. Simply said, design is the process through which requirements are defined, beginning with what is required.

UML Diagrams

By adhering to a standard set of syntactic, semantic, and pragmatic norms, the software engineer may express an analytical model in the unified modeling language.

Data Flow Diagram (DFD):

How data enters and exits the system, as well as its storage locations, may be shown graphically in a Data Flow Diagram (DFD). Any operational process in an organization may be represented effectively by means of a data flow diagram.

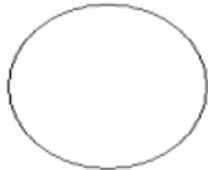



Advantages of DFD:

Users and system administrators alike will have no trouble making sense of these notations. Users may participate in DFD research to improve its precision. Inspecting charts and beginning to take precautions early on helps reduce the likelihood of a failed system.

The goal of the "bubble Chart" representation of a DFD, which is used to simplify system requirements and highlight important transformations that will become programs in system design, is to facilitate the design process. That's why it's the first step in designing everything down to the finest detail. The data flows of a system are represented as a sequence of bubbles in a data flow diagram (DFD).

Notations used to draw DFD are as follows:



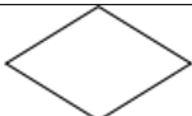

Table:1 symbol used in DFD

Name	Symbol	Meaning
process		Transforms of incoming data flows(s) to outgoing data flows(s).
Data Store		A repository of data that is to be store for use by one or more processes.
Data Flow		Movement of the data in the system.
External Entity		Sources and Destination outside the specified system boundary.

4.2 Entity Relationship Diagram (ER)

- Using entity relationship diagrams (ERDs), one may see how a database is organized.
- A graphical depiction of entities and their connections to one another is called an entity relationship diagram.
- The ERD is a graphical depiction of the connections between databases.

The Notations:

<u>Shapes</u>	<u>Name</u>	<u>Meaning</u>
	Rectangles	Entity.
	Ellipse	Attributes.
	Diamonds	Relationship among entity set.
	Lines	Link attributes to entity set and entity set to relationships

4.3 Use Case Diagram

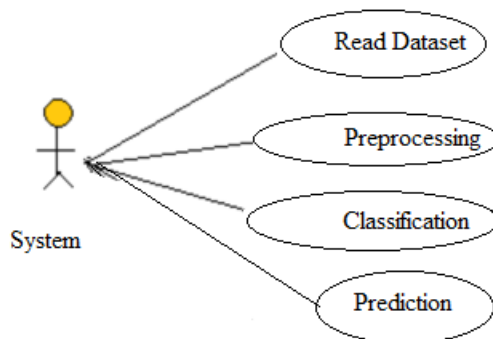


Figure 2: Case Diagram

4.4 Sequence Diagram

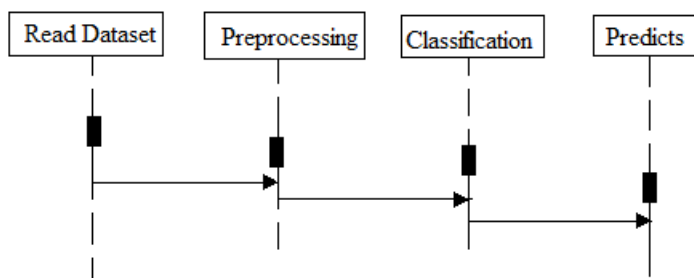


Figure 3: Sequence Diagram

V. FORM DESIGNS

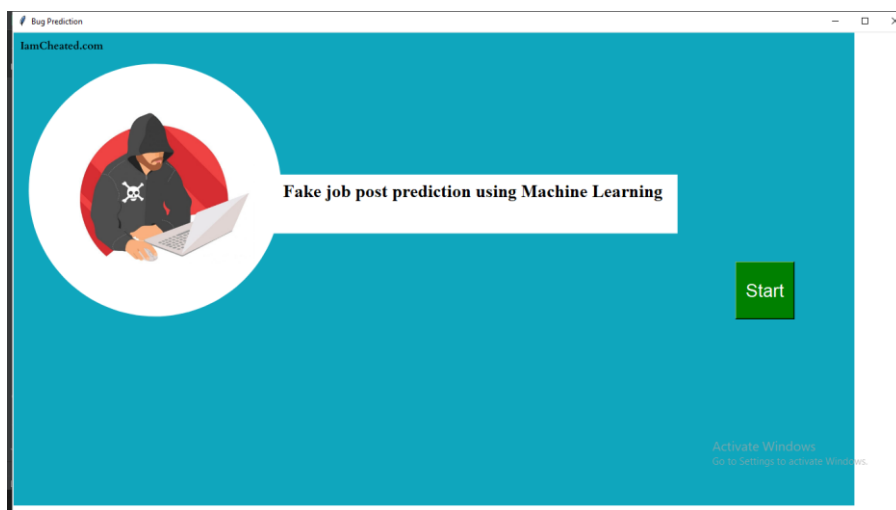


Fig-4: Main

The program's primary interface, from which one may launch it by pressing the Start button.

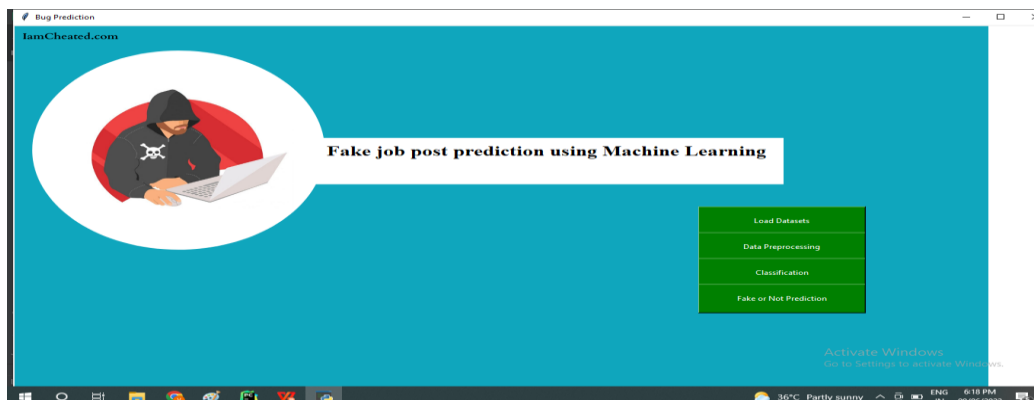


Fig-5: Menu

Load dataset, Data preparation, Classification, and Fake Job Prediction are the available options on this menu.

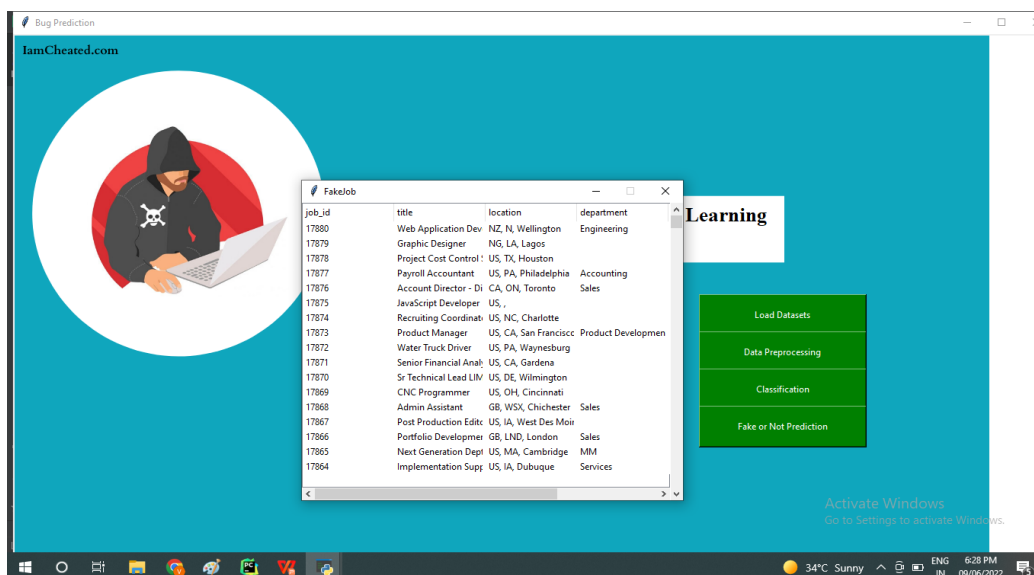


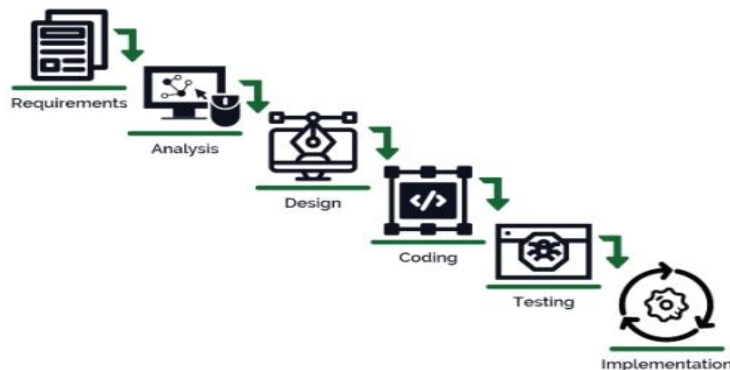
Fig-6: Read Dataset

Dataset reading tool. Attributes such as job_id, title, locations, department, etc. are included in the dataset.

VI. SYSTEM IMPLEMENTATION

6.1 Methodologies

The waterfall model is a traditional method to system development that takes a step-by-step, linear approach. Waterfall refers to the model's downward progression from one stage to the next. The waterfall methodology does not specify how to revert to a prior stage in the event of a requirement change. The first method used for creating software was called the waterfall method.



6.2 Programming Languages

Guido van Rossum creates Python. In 1989, Guido van Rossum initiated the introduction of Python. Python may be downloaded and used without cost. Python is a popular scripting language because of its high degree of abstraction, ease of use, and flexibility. Python was created with readability in mind. It relies heavily on English vocabulary whereas the other languages rely on punctuation. Fewer syntactic constructs exist in this language than in others. Python is interpreted, which means that the interpreter processes Python code at runtime. Your software may be run without first being compiled. Something like PERL or PHP. Python allows for direct interaction between the programmer and the interpreter, meaning that code may be written at a Python prompt.

Python is an object-oriented language, meaning that it can be used to write programs that encapsulate their logic in separate objects.

VII. SYSTEM TESTING

Introduction:

The software testing process is the last check on the specifications, designs, and codes that make up a piece of software. Executing a program to look for bugs is known as testing. The testing process involves running the software in question alongside a collection of test cases and analyzing the results to see whether they conform to specifications.

Testing Objectives:

- The term "testing" refers to the process of running a program with the intention of discovering bugs.
 - A well-designed set of test cases should be able to unearth previously unknown bugs.
 - A successful test is one that reveals a previously unknown flaw. These aforementioned goals need a radical shift in perspective.
- Software testing can only confirm the existence of faults; it cannot prove that none exist.

The following are the Testing methodologies:

Unit Testing:

When verifying software, unit testing isolates and evaluates individual modules. This test makes sure each component works as intended before moving on to the next. Unit testing describes the practice of testing individual modules.

Integration Testing:

It's a methodical approach of building independent software modules into a cohesive whole. This verification procedure finds and confirms module-wide faults.

Output Testing:

The purpose of output testing is to determine the accuracy of a particular output.

Validation Testing:

The program is complete and meets all requirements once it has undergone integration testing. However, verification against the specification is necessary to find any unforeseen future flaws and boost its dependability.

Software Testing Strategies:

A software developer may use a software testing plan as a guide. The actions involved in testing may be organized ahead of time and carried out in a methodical fashion. This is why it's important to develop a test case design approach for the software engineering process and a set of actions for software testing. The following elements are necessary for every software testing strategy:

1. Module-level testing is done first, and then "outward" to the full integration of the computer system.
2. At certain stages, different types of testing are warranted.
3. Both the software's creator and an external testing team are responsible for the process.
4. Although testing and debugging are distinct processes, debugging has to be considered when developing a testing plan.

VIII. CONCLUSION

With the help of fraud detection software, job-seekers will be directed to authentic employment opportunities. In this research, we present many machine learning methods as potential defenses for detecting job scams. The use of multiple classifiers for detecting employment fraud is shown by means of a supervised process. Based on the experimental data, it seems that the Random Forest classifier is superior than its contemporaries.

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, 2019, 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 classification and regression,|| Neurocomputing, vol. 2, no. 5–6, pp. 183–197,1991, doi: 10.1016/0925-2312(91)90023-5.
- [5] P. Cunningham and S. J. Delany, —K -Nearest NeighbourClassifiers,|| Mult. Classif. Syst., no. May, pp. 1–17, 2007,doi: 10.1016/S0031-3203(00)00099-6.
- [6] H. Sharma and S. Kumar, —A Survey on Decision Tree Algorithms of Classification in Data Mining,|| Int. J. Sci. Res., vol. 5, no. 4, pp. 2094–2097, 2016, doi:10.21275/v5i4.nov162954.
- [7] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A.O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: review, approaches and open research problems,|| Heliyon, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.
- [8] L. Breiman, —ST4_Method_Random_Forest,|| Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1017/CBO9781107415324.004.