© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715441

Real-Time AI Job Scam Detection System

ASAN NAINAR M[1], SURYA PRAKASH K[2]
[1]*Assistant Professor, Department of Computer Application, SRM Valliammai Engineering college, Anna University, Chennai, Tamil Nadu, India*
[2]*PG Student, Department of Computer Application, SRM Valliammai Engineering college, Anna University, Chennai, Tamil Nadu, India*

Abstract- ***The rapid growth of online recruitment platforms has increased opportunities for job seekers but has also led to a significant rise in fraudulent job postings and recruitment scams. Fake job advertisements often deceive candidates by requesting sensitive personal information or demanding fraudulent payments. Detecting such scams manually is difficult and time-consuming for users. This research proposes a Real-Time AI Job Scam Detection System, an intelligent web-based platform designed to automatically analyze job postings and identify potential fraudulent recruitment offers. The system uses machine learning and natural language processing techniques to evaluate job descriptions and classify them as Scam or Genuine. The proposed system supports multiple input sources including text job descriptions, job posting URLs, email-based job offers, and browser extension-based job scanning from recruitment platforms such as Naukri, Indeed, and Internshala. The system extracts job content from these sources, validates whether the content represents a legitimate job requirement, and then performs scam detection using trained machine learning models. The system also provides scam probability percentage, risk score, suspicious indicators, and explanation of prediction results, enabling users to understand the reason behind the classification. Additionally, a Chrome browser extension allows real-time job analysis directly from job portals, making the solution highly practical for everyday users. The proposed system demonstrates how artificial intelligence can be applied to enhance online recruitment security and protect job seekers from fraudulent employment schemes.***

I. INTRODUCTION

Online recruitment platforms have transformed the way organizations hire employees and how job seekers search for employment opportunities. Platforms such as Naukri, Indeed, and Internshala provide thousands of job listings across various industries. While these platforms improve accessibility to employment opportunities, they have also become a target for fraudulent recruiters who exploit job seekers through fake job offers.

Job scams have become a serious problem worldwide. Fraudulent recruiters often post attractive job offers with unrealistic salaries, request registration fees, or collect sensitive personal and financial information from applicants. Many job seekers are unable to identify such scams due to the lack of verification mechanisms in online job advertisements.

Traditional scam detection methods rely on manual verification or user reports, which are often slow and inefficient. With the increasing number of online job postings, automated solutions are required to analyze job advertisements and detect suspicious patterns effectively.

Recent advancements in machine learning and natural language processing (NLP) have enabled intelligent systems to analyze textual information and detect fraudulent content. By analyzing linguistic patterns, suspicious keywords, salary anomalies, and recruiter contact information, AI models can classify job postings as legitimate or fraudulent.

This research proposes a Real-Time AI Job Scam Detection System that automatically analyzes job postings using machine learning techniques. The system allows users to scan job content from multiple sources including text descriptions, job URLs, email-based job offers, and browser extension scanning of job portals.

The main objective of this research is to develop a secure, intelligent, and user-friendly platform that helps job seekers identify fraudulent recruitment advertisements before applying for them. By

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715441

integrating AI-based analysis with real-time job scanning technologies, the system provides an effective solution for preventing online job scams.

II. SYSTEM DESIGN

2.1 System Flow Diagram

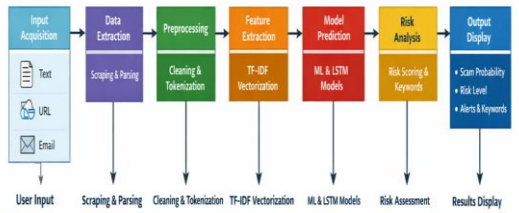

Fig.1- System Flow Diagram

The System Flow Diagram illustrates the complete working process of the *Real-Time AI Job Scam Detection System,* which is designed to analyze job postings and accurately identify potential scams. The process begins with the Input Acquisition stage, where the user provides job-related information in multiple formats such as text descriptions, job portal URLs, or email content. This multi-format support ensures that the system can effectively handle real-world job data collected from various online platforms and recruitment sources. Once the input is received, the system moves to the Data Extraction stage, where relevant job details are gathered. For URL inputs, web scraping techniques are applied to extract the job description from web pages, while email inputs are parsed to retrieve meaningful content. After extraction, the data enters the Preprocessing stage, where it is cleaned and normalized by converting text to lowercase, removing special characters, and performing tokenization. This step ensures that the data becomes structured and suitable for further analysis.

Following preprocessing, the cleaned data is passed to the Feature Extraction stage, where the TF-IDF (Term Frequency–Inverse Document Frequency) technique transforms the textual information into numerical feature vectors that machine learning models can process efficiently. The system then proceeds to the Model Prediction stage, where the transformed data is analyzed using trained Machine Learning algorithms along with a deep learning LSTM model to calculate the probability of the job posting being fraudulent or genuine. After prediction, the Risk Analysis stage evaluates the results by identifying suspicious keywords and patterns commonly associated with scams, and based on this evaluation, assigns a risk level such as Low, Medium, High, or Critical. Finally, in the Output Display stage, the system presents a detailed result to the user, including the scam probability score, risk classification, and highlighted suspicious indicators. This comprehensive output enables users to clearly understand the nature of the job posting and make informed decisions, thereby enhancing their safety while searching for employment opportunities online.

2.2 System Architecture Diagram

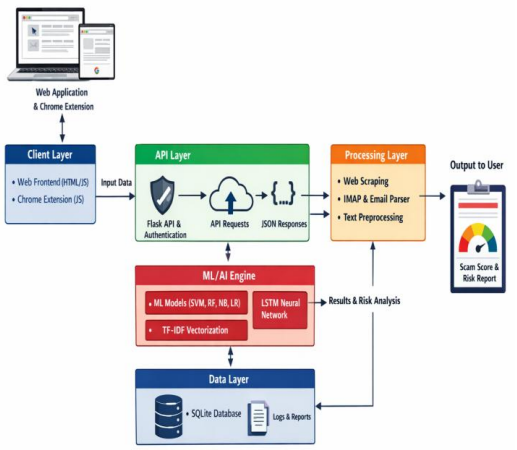

Fig.2- System Architecture Diagram

The System Architecture Diagram represents the overall structure and interaction between the different components of *the Real-Time AI Job Scam Detection System.* The architecture is designed using a layered approach to ensure scalability, efficiency, and modularity. At the top level, the Client Layer consists of the Web Application and Chrome Extension, which serve as the primary interfaces for user interaction. Users can input job-related data such as text, URLs, or email content through these interfaces. The input is then forwarded to the API Layer, which is built using a Flask-based backend. This layer is responsible for handling incoming requests, managing authentication, and securely transferring data between the frontend and backend components.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715441

It processes requests in the form of API calls and returns structured JSON responses, ensuring smooth communication across the system.

The next stage in the architecture is the Processing Layer, where the core data handling operations take place. This layer performs tasks such as web scraping for URL inputs, email parsing using IMAP protocols, and text preprocessing techniques including tokenization and cleaning. The processed data is then passed to the ML/AI Engine, which acts as the intelligence core of the system. This engine uses TF-IDF vectorization to convert textual data into numerical form and applies multiple Machine Learning models such as Support Vector Machine, Random Forest, Naive Bayes, and Logistic Regression, along with a deep learning LSTM model. These models work together to analyze patterns in the data and generate a prediction regarding the likelihood of the job being fraudulent. The ensemble approach improves accuracy and ensures reliable detection of scam patterns.

Finally, the architecture includes the Data Layer, which uses an SQLite database to store user information, prediction history, logs, and generated reports. This layer ensures data persistence and allows both users and administrators to track past analyses. The results generated by the ML/AI Engine are sent back through the Processing and API layers to the user interface, where they are displayed as a detailed risk report. This includes the scam probability score, risk classification, and identified suspicious indicators. Overall, the architecture ensures seamless data flow between components, enabling real-time processing, accurate predictions, and a user-friendly experience while maintaining system performance and reliability.

2.3 Database Diagram

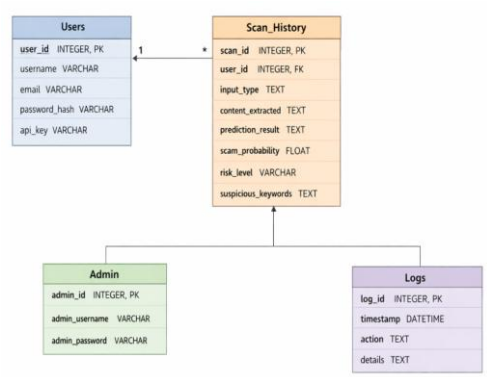

Fig.3- Database Diagram

The Database Diagram of the *Real-Time AI Job Scam Detection System* illustrates a structured and relational approach to managing all the data generated and processed within the system. At the foundation of this design is the Users table, which stores critical user information such as user ID, username, email, password hash, and API key. This table is essential for handling user authentication, authorization, and secure access to the application. Each registered user interacts with the system by performing job scans, and these interactions are systematically recorded. The relationship between the Users table and the Scan_History table is established as a one-to-many mapping, meaning a single user can have multiple scan records. This design ensures that every activity performed by a user is tracked and can be retrieved later for analysis, thereby enhancing personalization and accountability within the system.

The Scan_History table serves as the central repository for storing all job analysis results generated by the AI models. It contains detailed attributes such as scan ID, user ID (as a foreign key), input type (text, URL, or email), extracted job content, prediction result (genuine or scam), scam probability score, assigned risk level, and identified suspicious keywords. This comprehensive storage allows users to revisit previous scans, understand the reasoning behind predictions, and monitor patterns over time. In addition to this, the Admin table is incorporated to manage administrative operations, storing admin credentials that allow authorized personnel to oversee system functionality, manage user accounts, and generate reports. The Logs table

further supports the system by recording important events such as timestamps, user actions, API requests, and error messages, which are crucial for debugging, monitoring, and maintaining system transparency.

The relationships defined in the database ensure data integrity and efficient operation of the system. Primary keys uniquely identify each record, while foreign keys enforce referential integrity by linking scan records to valid users. This prevents inconsistencies and ensures reliable data management. The database is implemented using SQLite along with SQLAlchemy ORM, which simplifies complex database operations and ensures seamless integration with the Flask backend. Overall, the database design is optimized for performance, scalability, and security, enabling the system to efficiently handle large volumes of job scan data while maintaining accurate records and supporting advanced analytical capabilities.

III. SYSTEM IMPLEMENTATION

The implementation of *the Real-Time AI Job Scam Detection System* is carried out using a full-stack development approach that integrates frontend, backend, database, and machine learning components into a unified platform. The backend is developed using the Flask framework in Python, which acts as the central controller for handling user requests, routing, authentication, and communication with the machine learning models. The system is initialized through a main application file where all configurations, routes, and extensions are defined. User authentication is securely implemented using Flask-Login, and passwords are stored in a hashed format using Werkzeug to ensure data security. The frontend is designed using HTML, CSS, and Bootstrap, providing a responsive and user-friendly interface where users can input job data, view results, and navigate through different modules such as dashboard, scanner, and admin panel.

The core functionality of the system lies in its data processing and machine learning pipeline. When a user submits input (text, URL, or email), the backend processes the data using utility functions that perform tasks such as web scraping (using BeautifulSoup), email parsing, and text cleaning. The preprocessing stage includes converting text to lowercase, removing special characters, tokenizing the text, and eliminating stopwords using Natural Language Processing techniques. After preprocessing, the cleaned data is transformed into numerical features using a pre-trained TF-IDF vectorizer. These features are then passed into multiple trained machine learning models, including Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest, along with an optional LSTM deep learning model. The models generate prediction probabilities, which are combined using an ensemble approach to produce a final scam probability score.

Once the prediction is generated, the system performs a risk analysis by identifying suspicious keywords and patterns commonly associated with fraudulent job postings. Based on the probability score and detected flags, the system assigns a risk level such as Low, Medium, High, or Critical. The results are then formatted into a structured response and sent back to the frontend through API endpoints. The output includes a confidence score, risk classification, and highlighted suspicious indicators, which are visually presented using progress bars, color-coded alerts, and keyword tags. Additionally, all prediction results are stored in the SQLite database using SQLAlchemy, allowing users to track their scan history and enabling administrators to monitor system performance and generate reports.

The system also includes advanced features such as a Chrome Extension that interacts with the backend API to provide real-time scam detection directly on job portals. The extension captures job descriptions from web pages and sends them to the API for instant analysis, displaying the results without requiring users to leave the page. The implementation ensures modularity, scalability, and efficiency by separating concerns across different layers of the application. Overall, the system successfully integrates web technologies, machine learning models, and database management into a cohesive solution that delivers accurate, real-time job scam detection while maintaining security, usability, and performance

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715441

IV. METHODOLOGY

The proposed Real-Time AI Job Scam Detection System follows a modular and systematic approach to analyze job postings and detect fraudulent activities in real time. The methodology is divided into multiple interconnected modules, where each module performs a specific function such as data collection, preprocessing, feature extraction, prediction, and risk evaluation. This modular design ensures scalability, efficiency, and accurate detection of scam job postings. The system processes user input from various sources, applies Natural Language Processing (NLP) and Machine Learning (ML) techniques, and generates a structured output that includes scam probability, risk level, and suspicious indicators.

1. Input Acquisition Module

The Input Acquisition Module collects job-related data from multiple input sources such as direct text input, job posting URLs, and email content (EML or IMAP). This flexibility allows the system to handle real-world job data from job portals, recruitment emails, and user-provided descriptions. The collected input serves as the primary data source for further processing and analysis

2. Data Extraction Module

The Data Extraction Module is responsible for extracting meaningful job content from the provided input. For URL inputs, web scraping techniques are used to retrieve job descriptions from web pages, while email inputs are parsed to extract relevant content. In the case of direct text input, the content is taken as-is. This module ensures that only relevant textual information is passed to the next stage.

Pseudocode for Data Extraction

Algorithm DataExtraction
Input: UserInput
Output: ExtractedText

1. Read UserInput
2. If input is URL
    a. Perform web scraping
    b. Extract job description
3. Else if input is Email
    a. Parse email content
    b. Extract text
4. Else
    a. Use direct text input
5. Return ExtractedText

3. Text Preprocessing Module

The Text Preprocessing Module prepares the extracted data for machine learning analysis. It involves cleaning and normalizing the text by converting it to lowercase, removing special characters, tokenizing words, and eliminating stopwords using NLP techniques. This step ensures that the data is structured and noise-free.

Pseudocode for Text Preprocessing

Algorithm TextPreprocessing
Input: ExtractedText
Output: CleanText

1. Convert text to lowercase
2. Remove special characters and punctuation
3. Perform tokenization
4. Remove stopwords
5. Normalize text
6. Return CleanText

4. Feature Extraction Module

The Feature Extraction Module converts the cleaned textual data into numerical form using the TF-IDF (Term Frequency–Inverse Document Frequency) technique. This transformation enables machine learning models to process and analyze the data effectively.

Pseudocode for Feature Extraction

Algorithm FeatureExtraction
Input: CleanText
Output: FeatureVector

1. Load pre-trained TF-IDF vectorizer
2. Transform CleanText into numerical features
3. Return FeatureVector

5. Scam Prediction Module

The Scam Prediction Module is the core component of the system, where the feature vectors are analyzed using trained Machine Learning models such as Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest. Additionally, a deep learning LSTM model may be used to capture contextual patterns. The models generate probabilities indicating whether the job posting is genuine or fraudulent.

Pseudocode for Scam Prediction

Algorithm ScamPrediction
Input: FeatureVector
Output: ScamProbability

1. Load trained ML models
2. Pass FeatureVector to each model
3. Get prediction probabilities
4. Combine results using ensemble method
5. Return ScamProbability

6. Risk Analysis Module

The Risk Analysis Module evaluates the prediction results and identifies suspicious keywords or patterns commonly found in job scams (e.g., "payment required", "urgent hiring"). Based on the scam probability and detected flags, a risk level such as Low, Medium, High, or Critical is assigned.

Pseudocode for Risk Analysis

Algorithm RiskAnalysis
Input: ScamProbability, CleanText
Output: RiskLevel, SuspiciousKeywords

1. Define keyword list for scams
2. Search keywords in CleanText
3. Assign risk level based on probability:
   If probability < 30 → Low
   If 30–60 → Medium
   If 60–80 → High
   If >80 → Critical
4. Return RiskLevel and SuspiciousKeywords

7. Output Generation Module

The Output Generation Module presents the final results to user in an understandable format. It displays the scam probability score, assigned risk level, and highlighted suspicious keywords. The results are also stored in the database for future reference and analysis

Pseudocode for Output Generation

Algorithm OutputGeneration
Input: ScamProbability, RiskLevel, SuspiciousKeywords
Output: ResultDisplay

1. Format output data
2. Display probability score
3. Show risk level (Low/Medium/High/Critical)
4. Highlight suspicious keywords
5. Store results in database
6. Return ResultDisplay

8. Chrome Extension Integration Module

The Chrome Extension Module enables real-time detection directly on job portals. It captures job descriptions from web pages and sends them to the backend API for analysis. The results are displayed instantly on the same page, improving user convenience and workflow.

Pseudocode for Extension Integration

Algorithm ExtensionScan
Input: WebPageContent
Output: ScanResult

1. Capture job description from webpage
2. Send data to backend API
3. Receive prediction result
4. Display risk level on webpage
5. Return ScanResult

This methodology ensures that the system operates efficiently by integrating multiple modules, each handling a specific task in the job scam detection process. The combination of NLP, Machine Learning, and real-time processing enables accurate

and reliable identification of fraudulent job postings, thereby enhancing user safety and trust.

V. RESULTS AND ANALYSIS

A. Test Methodology

To evaluate the performance of the proposed *Real-Time AI Job Scam Detection System,* extensive testing was conducted using different types of job-related inputs, including direct text descriptions, job posting URLs, and email-based job offers. The primary objective of this testing process was to measure the system's ability to accurately classify job postings as genuine or fraudulent and to assess the effectiveness of the machine learning models in detecting scam patterns. The system was tested using a labeled dataset consisting of both legitimate and fake job postings, ensuring that the evaluation reflects real-world scenarios where job seekers encounter a wide variety of job advertisements.

The testing process involved passing each input through the complete system pipeline, including data extraction, preprocessing, feature extraction, and prediction modules. For URL inputs, web scraping techniques were used to extract job descriptions, while email inputs were parsed to retrieve relevant content. The extracted text was then cleaned and transformed using Natural Language Processing techniques before being converted into numerical vectors using the TF-IDF method. These vectors were processed by the ensemble machine learning models and the LSTM model to generate a scam probability score. The predicted output was then compared with the actual label (genuine or fraudulent) present in the dataset to evaluate the system's accuracy and reliability.

The evaluation was performed using the following steps:

1. Collect a dataset containing both genuine and fraudulent job postings.
2. Provide different types of inputs (text, URL, email) to the system.
3. Process the input through data extraction and preprocessing modules.
4. Generate predictions using machine learning and deep learning models.
5. Compare predicted results with actual labels to measure accuracy, precision, and recall.

B. Sample Test Dataset

Table 1 shows a sample dataset used for testing the Real-Time AI Job Scam Detection System. The dataset includes different types of inputs such as text descriptions, job URLs, and email job offers. Each input is labeled as either Genuine or Scam, and the system's detected output is compared with the expected result to evaluate performance.

Table 1: Sample Job Scam Detection Test Data

Input Type	Sample Input	Expected Result	Detected Result
Text	Earn ₹50,000/week from home. No experience needed	Scam	Scam
URL	Verified company job listing with proper description	Genuine	Genuine
Email	Official interview invitation from company domain	Genuine	Genuine

C. Scam Detection Accuracy

To measure the performance of the system, accuracy was calculated using the formula:

Accuracy = (Correct Predictions / Total Predictions) × 100

Based on the testing results, the accuracy of different detection modules is as follows:

Table 2: Scam Detection Accuracy Data

Module	Accuracy
Text-Based Detection	90%
URL-Based Detection	88%
Email-Based Detection	92%

D. Mean Accuracy Calculation

The overall performance of the *Real-Time AI Job Scam Detection System* is evaluated by calculating the Mean Accuracy of all detection modules. Since the system processes different types of inputs such as text, URL, and email, each module produces its own

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715441

accuracy score. To obtain a single performance metric that represents the system's effectiveness, the mean (average) accuracy is calculated by combining the accuracy values of all modules.

Let:

- A_t= Accuracy of Text-Based Detection
- A_s= Accuracy of URL-Based Detection
- A_f= Accuracy of Email-Based Detection

Then the overall system accuracy is calculated as:

$$A_{overall} = \frac{A_t + A_s + A_f}{3}$$

Substituting the experimental values

$$A_{overall} = \frac{90 + 88 + 92}{3}$$

$$A_{overall} = 90\%$$

E. Discussion

The *Real-Time AI Job Scam Detection System* effectively identifies fraudulent job postings across text, URL, and email inputs using NLP and an ensemble of machine learning models. It accurately detects common scam patterns such as fake salary offers, payment requests, and suspicious domains, while also providing transparent outputs like risk levels and highlighted keywords. The Email module shows the highest accuracy, followed by Text and URL modules.

However, the system depends on dataset quality and may struggle with highly sophisticated or multilingual scams. Despite these limitations, it remains a reliable and scalable solution, with future improvements possible through advanced models and multilingual support.

VI. CONCLUSION

The *Real-Time AI Job Scam Detection System* successfully addresses the growing issue of fraudulent job postings by providing an intelligent and automated solution for identifying scams. By integrating Natural Language Processing (NLP) techniques with an ensemble of machine learning models, the system is capable of analyzing job-related data from multiple sources such as text, URLs, and emails with high accuracy. The system not only detects whether a job posting is genuine or fraudulent but also provides detailed insights through risk levels and highlighted suspicious keywords, enabling users to make informed decisions.

The implementation of a multi-module architecture ensures efficient data processing, real-time prediction, and seamless user interaction through both a web application and a Chrome extension. The experimental results demonstrate strong performance across different input types, with an overall high accuracy, proving the effectiveness of the proposed approach. Although certain limitations exist, such as dependency on dataset quality and challenges in detecting highly sophisticated scams, the system provides a strong foundation for further enhancements.

In conclusion, the project serves as a reliable, scalable, and user-friendly solution to combat online job scams. With future improvements such as advanced deep learning models, multilingual support, and domain verification mechanisms, the system can be further enhanced to provide even more robust protection for job seekers in the digital recruitment ecosystem.

VII. FUTURE ENHANCEMENT

The *Real-Time AI Job Scam Detection System* can be further improved by incorporating advanced technologies and additional features to enhance its accuracy, scalability, and usability. One major enhancement is the integration of Transformer-based deep learning models such as BERT or RoBERTa, which can better understand the contextual meaning of job descriptions and detect more sophisticated scam patterns. Additionally, implementing a continuous learning mechanism through user feedback (false positives and false negatives) can help the system dynamically update and improve its prediction performance over time.

Another important enhancement is the introduction of multilingual support, enabling the system to analyze job postings in regional and international languages, which is especially useful in diverse job markets. The system can also be extended with real-time company and domain verification, where APIs such as WHOIS or official company registries are used to validate employer authenticity. Furthermore, integrating phishing detection techniques for email analysis and enhancing URL scanning with advanced web security checks can significantly improve the detection of complex fraud attempts.

From a usability and deployment perspective, the system can be scaled into a cloud-based SaaS platform to support a larger number of users with high availability and performance. Mobile application support can also be introduced to provide on-the-go scam detection for job seekers. Enhancing the Chrome extension to support multiple browsers and job platforms will improve accessibility. Overall, these future enhancements will make the system more intelligent, adaptive, and comprehensive, ensuring stronger protection against evolving job scam threats.

REFERENCES

[1] Salton, G., & Buckley, C. (1988). *Term-weighting approaches in automatic text retrieval*. Information Processing & Management.

[2] Ramos, J. (2003). *Using TF-IDF to determine word relevance in document queries*. Proceedings of the First Instructional Conference on Machine Learning.

[3] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[4] Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing*. Pearson.

[5] Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.

[6] Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

[7] Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation.

[8] McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference.

[9] Richardson, L. (2007). *Beautiful Soup Documentation*.

[10] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*.

[11] Grinberg, M. (2018). *Flask Web Development*. O'Reilly Media.

[12] Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

[13] Kaggle. (2020). *Fake Job Postings Dataset*. Retrieved from https://www.kaggle.com/

[14] Google Developers. (2023). *Chrome Extension Documentation (Manifest V3)*.

[15] OWASP Foundation. (2021). *OWASP Top 10: Web Application Security Risk*