iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81812

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Automated Job Offer Trust Scoring System

Pavithra K[1], Poojalakshmi G[2], Naziya Banu I[3], Serin S[4], Sowndharya M[5]
Department of Computer Science and Engineering Arunai Engineering College (Autonomous), Tamil Nadu, India

Abstract: ***The Automated Job Offer Trust Scoring System (AJOTSS) is developed to help job seekers evaluate the legitimacy of job-related emails and recruitment offers. Scammers often use fake domains, copied company branding, and payment requests to deceive users. The proposed system analyses job-related email content and performs multiple verification checks, including domain name validation, NLP-based content scanning, fraud database lookup, and inspection of embedded links for suspicious payment or redirection behaviour. Based on these verification constraints, the system calculates a Trust Score between 0 and 100 that represents the credibility of the job offer. AJOTSS provides a practical, rule-based solution to reduce recruitment fraud risk and assist job seekers in making informed decisions.***

Keywords: Job fraud detection, recruitment scam, trust score, domain verification, NLP content analysis, phishing, link inspection, Flask, Python.

I. INTRODUCTION

In recent years, the recruitment process has undergone a major transformation due to the rapid growth of digital communication and online job platforms. Companies now rely heavily on emails, career portals, and professional networking platforms to reach potential candidates. This shift has made job searching faster, easier, and more accessible to students and professionals across the world.

One of the major challenges faced by job seekers today is the difficulty in distinguishing between genuine and fake job offers. Fraudulent emails are becoming increasinglysophisticated, making it hard even for educated candidates to identify suspicious patterns. Traditional methods of verification, such as manually checking company websites or searching for reviews, are time-consuming and not always reliable.

To address this issue, there is a need for an automated system that can assist users in evaluating job-related emails quickly and effectively. AJOTSS is designed to fulfil this requirement through a multi-module analysis pipeline.

A. Problem Statement

With the increasing dependency on online recruitment platforms, job seekers are constantly exposed to the risk of fraudulent job offers. Fake recruitment emails are designed in such a way that they closely imitate real company communications, making it difficult for individuals to identify whether the offer is genuine or not. Most existing solutions focus only on phishing detection or domain verification and do not provide a comprehensive multi-level analysis, increasing the chances of users falling victim to scams.

B. Scope of the Project

The scope of this project is focused on developing a system that analyses job-related emails and provides a Trust Score indicating their reliability. Key functionalities include: email content analysis, domain verification, fraud database checking, link inspection, and trust score generation (0–100). The system is implemented using Python and Flask, with a simple web interface accessible to users with minimal technical knowledge.

II. LITERATURE SURVEY

A review of existing literature was conducted to identify prior work relevant to automated fraud detection in recruitment, phishing identification, and trust score computation.

TABLE I Summary of Literature Survey

S.No	Title	Authors	Year	Methodology

©IJRASET: All Rights are Reserved

1	Automatic Detection of Online Recruitment Frauds	Chawla et al.	2017	Data mining on EMSCAD dataset
2	Phishing Detection: A Literature Survey	Khonji et al.	2013	Survey of email, URL, website-based methods
3	Detecting Phishing Websites Using ML Techniques	Masud et al.	2008	URL feature extraction and domain analysis
4	How Experts Detect Phishing Scam Emails	Oliveira et al.	2017	Behavioural study of expert inspection
5	A Survey of Trust Management Systems	Josang et al.	2007	Review of trust computation models

Chawla et al. [1] analysed the EMSCAD dataset using data mining techniques and demonstrated that fraudulent job postings exhibit measurable linguistic and structural patterns. Khonji et al. [2] showed that layered detection strategies significantly outperform single-modality approaches, providing the theoretical basis for AJOTSS's four-channel design. Masud et al. [3] demonstrated that domain inconsistencies and irregular URL redirection patterns are reliable indicators of phishing intent. Oliveira et al. [4] highlighted that human experts rely on linguistic cues such as urgency, unusual formatting, and payment requests, informing AJOTSS's NLP keyword categories. Josang et al. [5] provided a rigorous framework for computing trust by aggregating multiple evidence sources into a single numerical score, which is the direct theoretical ancestor of the AJOTSS Trust Score Engine.

III. OBJECTIVES

1) Develop a system that analyses job-related emails and determines their authenticity by generating a trust score.
2) Design a multi-level verification system that evaluates emails using content analysis, domain verification, web search, and link inspection.
3) Implement NLP-based content analysis to identify suspicious words and patterns such as urgency, payment requests, and misleading instructions.
4) Perform domain verification to check whether the sender's email address matches the official domain of the company.
5) Implement link inspection to analyse all URLs for suspicious behaviour such as redirection, unsecured connections, or payment-related pages.
6) Design a trust score calculation mechanism combining all module results into a score between 0 and 100.
7) Develop a simple and user-friendly web interface accessible to non-technical users.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

IV. SYSTEM ANALYSIS

A. Existing System

In the current scenario, there is no dedicated and unified system specifically designed to verify the authenticity of job-related emails. Job seekers generally rely on manual verification methods. Some existing tools focus on phishing detection or spam filtering, but are not tailored for recruitment fraud detection and fail to consider job-specific scam patterns. The existing system mainly depends on partial detection methods and user awareness, which are not sufficient to handle the growing complexity of recruitment scams.
Limitations include: lack of a unified multi-module system; time-consuming manual verification; limited detection capability for advanced scams that use professional language; no clear trust-score output for users; and dependence on user technical knowledge.

B. Proposed System

The proposed Automated Job Trust Scoring System (AJOTSS) overcomes these limitations by automatically analysing job emails through four independent verification modules and generating a composite Trust Score between 0 and 100. The system is implemented using Python and Flask, with a clean web interface. Advantages include: automated and quick verification; multi-level analysis for better accuracy; clear trust score output; reduced dependency on manual checking; and prevention of financial and data-related risks.

V. SYSTEM REQUIREMENTS

A. Hardware Requirements

S.No	Component	Specification
1	Processor	Intel Core i3 or above
2	RAM	Minimum 4 GB (8 GB recommended)
3	Hard Disk	500 GB or above
4	System Type	64-bit Computer
5	Input Devices	Keyboard and Mouse
6	Output Devices	Monitor

TABLE II HARDWARE REQUIREMENTS

B. Software Requirements

S.No	Software	Specification
1	OS	Windows 10/11 or Linux
2	Language	Python 3.10
3	Framework	Flask
4	IDE	PyCharm / VS Code
5	Frontend	HTML, CSS
6	Browser	Google Chrome / Edge

TABLE III SOFTWARE REQUIREMENTS

VI. SYSTEM DESIGN

A. Architecture Overview

AJOTSS is designed as a modular, web-based application. The system follows a pipeline architecture where the input email passes through four independent verification modules, each contributing a risk score to the final trust computation. The design focuses on simplicity, modularity, scalability, and efficiency.

©IJRASET: All Rights are Reserved

B. Data Flow

The data flow is: User submits job email → System extracts email content → NLP text processing → Verification checks executed (Domain + Fraud Database + NLP Content + Link Inspection) → Trust Score Generated (0–100) → Result displayed to user.

C. Design Considerations

- Modularity: Each module (content analysis, domain verification, fraud lookup, link inspection) is implemented independently, allowing separate testing, updating, and maintenance.
- Scalability: New detection rules, machine learning models, or external fraud APIs can be integrated without affecting the existing structure.
- User-Friendly Interface: Users only need to paste email text and click Analyse to obtain an instant trust score with a detailed breakdown.
- Efficiency: The rule-based approach avoids heavy computation and delivers real-time results on basic hardware.
- Security Awareness: The system highlights specific fraud indicators — fake domains, suspicious links, payment demands, and fraud-listed companies — to guide informed user decisions.

VII. MODULES

A. Email Input Module

The Email Input Module is the entry point of the system. It captures the user-pasted job email, performs basic preprocessing (whitespace removal, length validation), and forwards the cleaned data to all downstream analysis modules.

B. Domain Verification Module

This module extracts the sender's email domain using regular expressions and compares it against a curated list of verified official company domains. If the domain matches an official entry, a risk score of 0 is assigned and the email is immediately flagged as TRUSTED. If the domain impersonates a known brand (e.g., 'tcs-jobs.in'), a maximum domain risk of 30 is applied.

C. NLP Content Analysis Module

This module scans the email text against four keyword categories: (1) Payment keywords (e.g., 'registration fee', 'pay before joining') — a single hit forces the Trust Score to 0 and the verdict to FRAUDULENT; (2) Urgency keywords (e.g., 'within 24 hours', 'act now'); (3) Training trap keywords (e.g., 'training fee', 'internship deposit'); and (4) Credential-harvesting keywords (e.g., 'share your Aadhaar', 'bank details'). Non-payment flags accumulate a risk score of up to 30.

D. Fraud Database Check Module

This module checks the detected company name and domain against a built-in database of over 65 known fraudulent recruiters (BUILTIN_FRAUD_COMPANIES) plus a persistent user-maintained list. The database covers generic fake agencies, South India-specific fraud recruiters, brand-impersonating agencies, WFH/MLM scam operators, and fee-based fake internship providers. Users can dynamically add new entries via the 'Add Fraudulent Company' form.

E. Link Inspection Module

All URLs in the email are extracted using regular expressions. Each link is checked for: IP-address URLs, payment-related path keywords (pay, fee, deposit), credential-harvesting keywords (verify, login, signin), URL shorteners (bit.ly, tinyurl), and excessively long URLs. Each flagged link contributes 20 points to the link risk, capped at 40.

F. Trust Score Calculation Module

The Trust Score is computed as:

Trust Score = 100 – (Domain Risk + Content Risk + Fraud Risk + Link Risk)

Output bands: 80–100 → SAFE; 50–79 → CAUTION; below 50 → SUSPICIOUS; 0 (payment detected) → FRAUDULENT; 100 (official domain) → TRUSTED.

VIII. IMPLEMENTATION

A. Implementation Steps

1) Environment Setup: Install Python 3.10, Flask, and development IDE (PyCharm / VS Code).
2) Frontend Design: Create user interface using HTML and CSS with inline Flask templating.
3) Backend Development: Create Flask application (app.py) with route handlers for / (GET/POST) and /add-fraud (POST).
4) Module Implementation: Code all five analysis modules as Python functions within app.py.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

5) Trust Score Engine: Apply the scoring formula; handle edge cases (payment detected → 0, official domain → 100).
6) Fraud List Persistence: Use JSON file (custom_fraud_companies.json) to persist user-reported fraud entries across server restarts.
7) Testing: Test with real and synthesised genuine and fraudulent job emails; verify edge cases.

B. Key Source Code Segments

The Trust Score calculation function is the core engine:

```
def calculate_trust_score(d_risk, c_risk, w_risk, l_risk):
    score = max(0, 100 - (d_risk + c_risk + w_risk +
l_risk))
    if score >= 80:
        return score, "SAFE"
elif score >= 50:
        return score, "CAUTION"
    else:
        return score, "SUSPICIOUS"
```

IX. OUTPUT AND RESULTS

The system was tested with both fraudulent and genuine job emails. Figure 1 shows the AJOTSS home interface with the 'Add Fraudulent Company' form. Figure 2 shows the Trust Score output for a fraudulent email (score: 40, verdict: SUSPICIOUS). Figure 3 shows the score breakdown table identifying domain and fraud risk. Figure 4 shows the interface with a genuine TCS recruitment email pasted. Figure 5 shows the result for the trusted email (score: 100, verdict: TRUSTED). Figure 6 shows all verification modules passing with 0 risk for the trusted email.

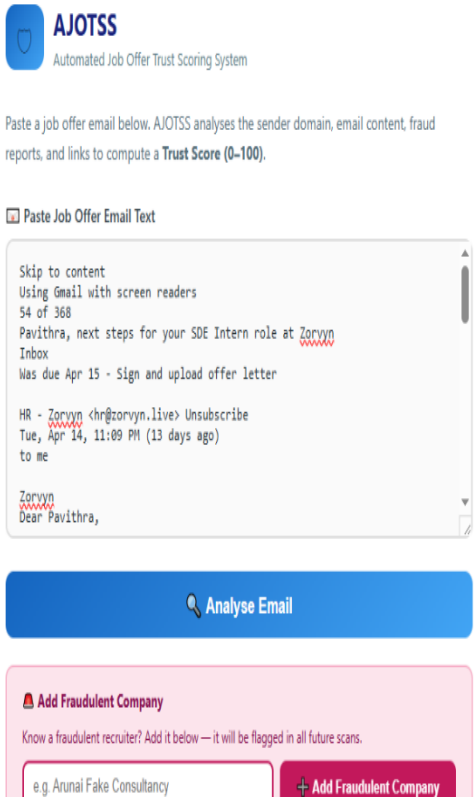


Figure1: AJOTSS home interface

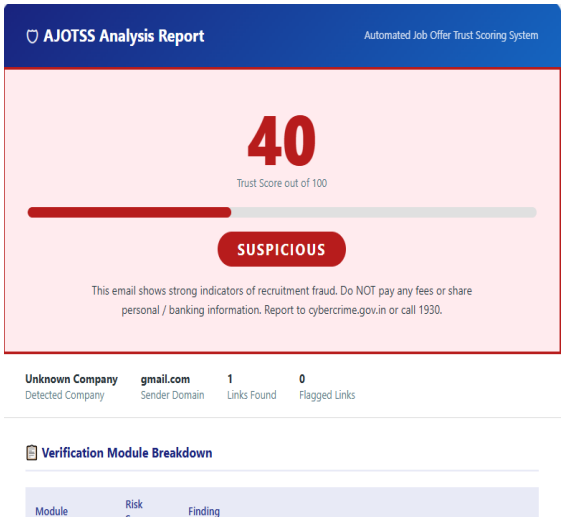


Figure2:Trust Score output fraudulent email

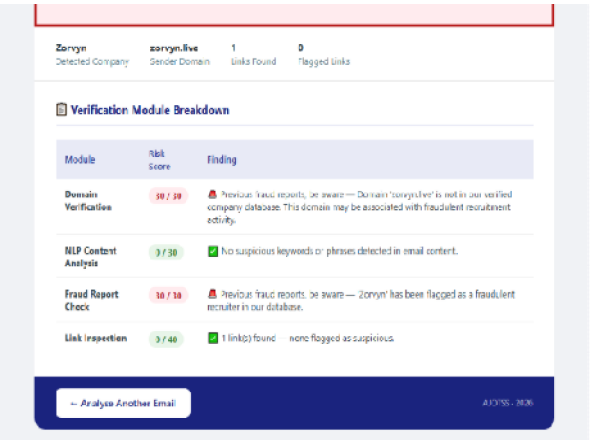


Figure3: Score breakdown

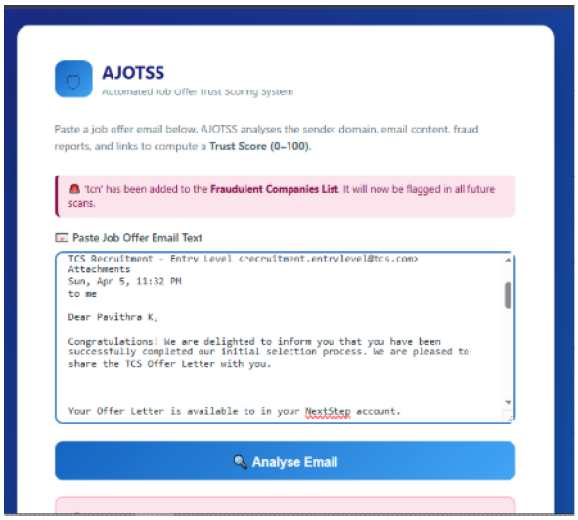


Figure4:Genuine recruitment email pasted

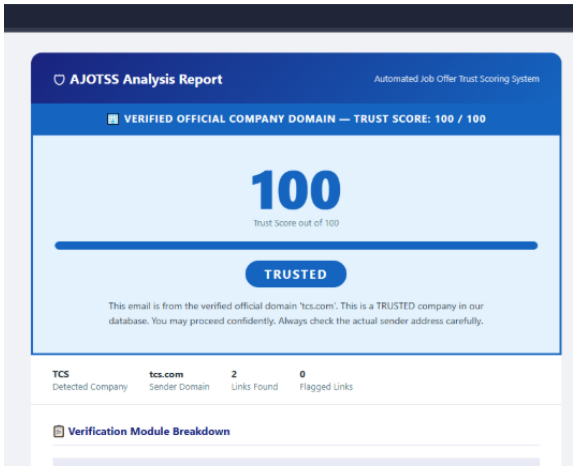


Figure5:Result for the trusted email

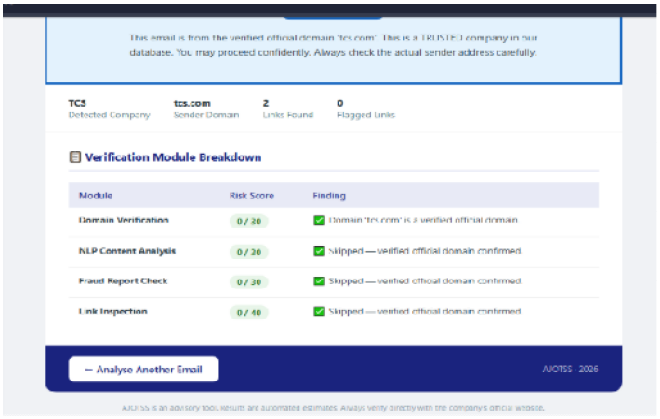


Figure6: All verification modules passing

Key observations from testing: (1) Emails containing any payment keyword are immediately classified FRAUDULENT with Trust Score = 0. (2) Emails from verified official domains (e.g., tcs.com) receive Trust Score = 100 (TRUSTED) with no further checks. (3) Emails using brand-impersonating domains (e.g., tcs-recruitment.live) score 40 (SUSPICIOUS) due to 30-point domain risk and 30-point fraud risk. (4) The dynamic fraud list allows real-time addition of newly identified scam companies.

X. CONCLUSION AND FUTURE ENHANCEMENT

The Automated Job Offer Trust Scoring System provides an effective, rule-based solution for identifying fraudulent job emails. By combining four independent verification modules domain verification, NLP content analysis, fraud database lookup, and link inspectioninto a weighted trust score, AJOTSS enables job seekers to make informed decisions quickly. The system successfully demonstrated the ability to detect payment demands, impersonated domains, and fraudulent recruiters, while correctly validating genuine communications from verified corporate domains.

Future enhancements include: (1) Integration of machine learning models (2) Connection to real-time fraud databases and cybercrime.gov.in APIs for live threat intelligence; (3) Browser extension for inline email scanning; (4) Multilingual support for regional Indian languages; (5) Mobile application deployment for wider accessibility.

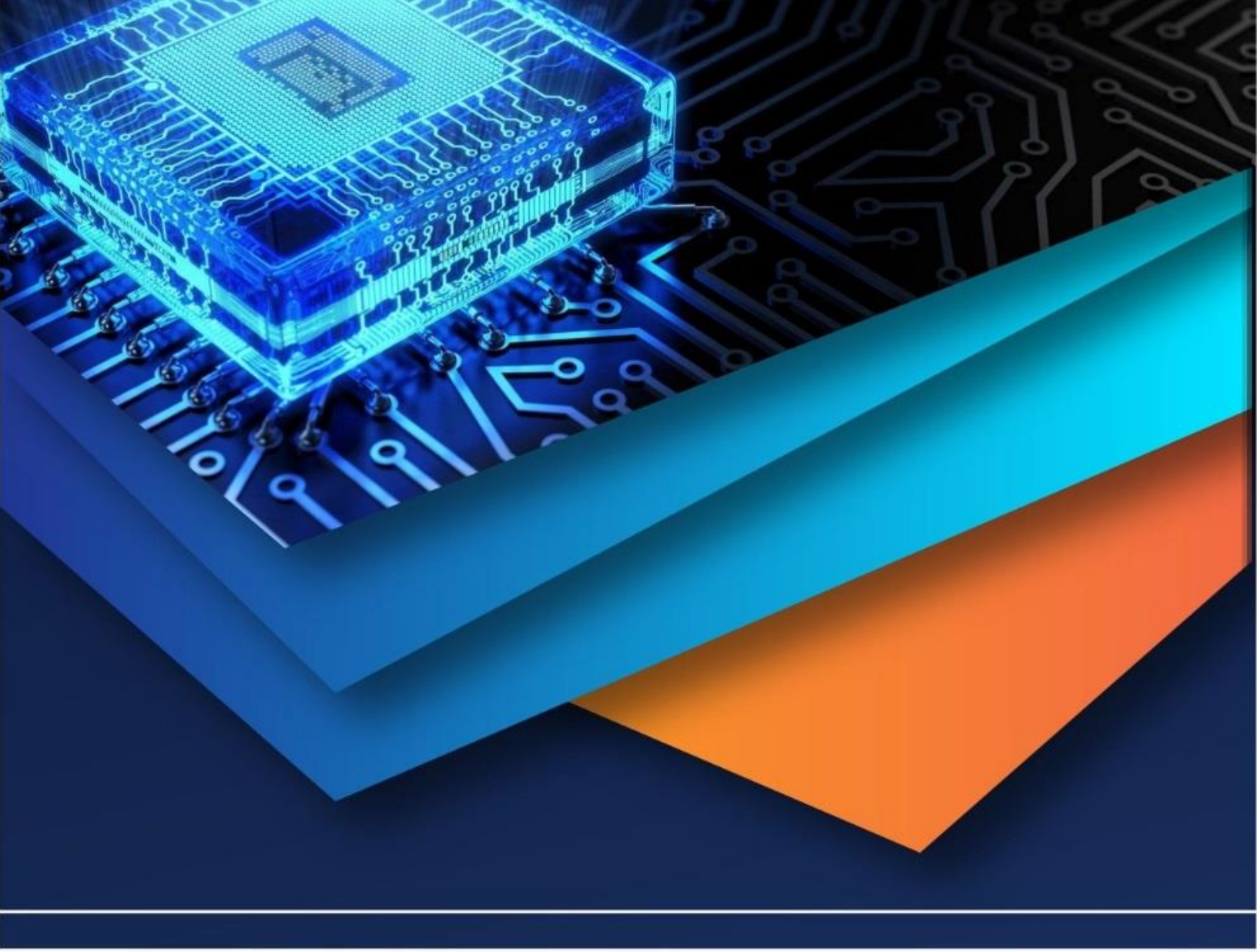
XI. ACKNOWLEDGMENT

The authors express sincere gratitude to Ms. M. Sowndharya M.E., Assistant Professor and Project Supervisor, and Mrs. V. Umadevi M.E., Head of the Department, Department of Computer Science and Engineering, Arunai Engineering College, for their invaluable guidance and support throughout this project. The authors also thank the Management and Principal, Dr. C. Elanchezhian, for providing the necessary facilities.

REFERENCES

[1] S. Chawla, J. Cichy, and T. G. Papaioannou, "Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset," Future Internet (MDPI), vol. 9, no. 6, pp. 1–12, 2017.
[2] M.Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," IEEE Communications Surveys and Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.
[3] M. A. U. Masud, T. Al-Khateeb, L. Khan, B. Thuraisingham, and K. W. Hamlen, "Detecting Phishing Websites Using Machine Learning Techniques," in Proc. IEEE Int. Conf. on Information Reuse and Integration, 2008, pp. 1–6.
[4] D. Oliveira, H. Rocha, and H. Yang, "Dissecting Spear Phishing Emails for Older vs Young Adults," in Proc. ACM CHI Conf. on Human Factors in Computing Systems, 2017, pp. 6412–6424.
[5] A.Josang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," Decision Support Systems (Elsevier), vol. 43, no. 2, pp. 618–644, 2007.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)