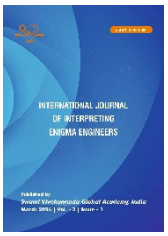
International Journal of Interpreting Enigma Engineers (IJIEE)
https://ejournal.svgacademy.org/index.php/ijiee/
https://doi.org/10.62674/ijiee.2026.v3i01.003
eISSN: 3048-6300
OPEN ACCESS

# NEXT GENRIC-BI ML-BASED RESUME INTELLIGENCE SYSTEM



Dr Gavini Sreelatha*, Akula Anjana, Amgoth Bindu, Devarakonda Gnapika

Original Article

[1]Associate Professor, Department of CSE-AIML, Stanley College of Engineering Technology for Women, Abids, Hyderabad, India.
[2,3,4] Undergraduate student, Department of CSE-AIML, Stanley College of Engineering and Technology for Women, Abids, Hyderabad, India.

*Corresponding Author's Email: [1]sreelathaprince13@gmail.com, [2]anjanaakula20@gmail.com, [3]binduamgoth42@gmail.com, [4]devarakondagnapika@gmail.com

## Abstract

Hiring's become a real challenge these days. Companies can get hundreds of resumes for just one job, all pouring in from job boards and online platforms. Sorting through that pile by hand is exhausting—and honestly, it's easy to make mistakes or let bias sneak in. So we built a Resume Intelligence System powered by machine learning and natural language processing to make things simpler.

Here's how it works: the system automatically scans every resume, grabs the important stuff like skills, education, and work history, then stacks those details up against what the job actually needs. It uses a similarity score to show, at a glance, who's the closest match for the job. And forget about staring at endless spreadsheets—the results pop up in a Power BI dashboard, with clear charts and reports anyone can understand.

With this setup, recruiters skip the dull, repetitive tasks and spend more time on real decision-making. Everything moves faster. Plus, the whole process is more consistent and fair, since we're letting the data do the heavy lifting. In the end, it means companies find the right people, with less hassle and less risk of missing great talent.

***Keywords: Resume Screening, Machine Learning, Natural Language Processing, Recruitment Analytics, Resume Classification, Power BI, Data Visualisation, AI in Hiring.***

## Introduction

Online job portals and digital hiring platforms keep growing, and recruiters are swamped with applications—sometimes hundreds for a single job. Sorting through all those resumes by hand isn't just exhausting; it's pretty unreliable. Plenty of great candidates slip through simply because their resumes don't fit a rigid template. Most software tools out there use basic keyword searches. Sure, they can pick out certain words, but they miss the bigger picture—the context, the way someone describes their skills and experience. So, lots of strong applicants end up overlooked just because their resumes aren't a perfect match for the expected phrasing.

Artificial Intelligence steps in with something smarter. Machine Learning finds patterns and makes predictions, while Natural Language Processing understands how people actually communicate. When you combine these two, you get a system that looks deeper than just matching keywords. It gets a real sense of who the candidates are and what they're bringing to the table.

This project is all about creating a smart resume analysis system. It automates the entire review process, then shows the results visually—clear and easy to read. With AI-driven analysis plugged right into Power BI dashboards, recruiters see

insights instantly. They can compare candidates side by side, spot strengths fast, and make decisions that are actually backed by data, not just gut instinct.

## Literature Survey

The authors in [1] discussed that the recruitment industry has significantly changed with the introduction of automated screening tools. Earlier, recruiters manually reviewed each resume, which required a large amount of time and effort. This traditional approach was not only slow but also prone to human errors and personal bias. With the growth of online job platforms, the number of applicants increased rapidly, making manual screening inefficient and difficult to manage.

In [2], the authors explained the introduction of Applicant Tracking Systems (ATS) to simplify the recruitment process. These systems automatically scanned resumes for specific keywords related to job requirements. Although ATS reduced the workload of recruiters, these early systems were limited because they only focused on keyword matching and could not understand the actual meaning of a candidate's experience or qualifications.

The authors in [3] highlighted the use of Natural Language Processing (NLP) techniques to improve resume analysis. NLP-based systems can analyze resumes more intelligently by extracting important information such as skills, educational qualifications, and work experience. Techniques such as tokenization, part-of-speech tagging, and named entity recognition help in understanding the context of the resume content.

In [4], the authors concluded that modern AI-based resume screening systems provide better insights into candidate profiles compared to traditional keyword-based systems. These advanced techniques help recruiters evaluate applicants more accurately and make the hiring process faster and more efficient.

## Proposed Approach

The idea behind this system is simple: stop wasting time on resume screening. Instead of manually sifting through stacks of PDFs and Word files, you just upload them on a web page. The system reads every resume, strips out the clutter, and keeps only the valuable details. It uses natural language processing to clean things up—so you're not seeing random formatting issues, repeated details, or anything that'll get in the way of real analysis. Once the data is cleaned, the machine learning model jumps in and transforms everything into numbers—so it can actually "understand" what's there.

From there, the model sorts resumes into job categories. Then another tool checks how well each resume matches a specific job description and gives each candidate a match score. All this data gets stored in a tidy, organized way. You see it visualized on a dashboard, where you can quickly scan candidate skills, experience levels, and match quality. Recruiters save time and zero in immediately on what'll lead to better hiring decisions.

**System Architecture**

The whole setup has four main layers, each doing its part to get resumes from "just uploaded" to "easily analyzed." First comes the user interface, a simple web app where you drag and drop your files. This front layer is all about making uploads quick and painless. Next, the processing layer does the heavy lifting: it reads the documents, extracts the text, and uses NLP tricks to clean everything up. The machine learning models step in here, sorting and scoring resumes against job categories and descriptions.

Once everything's processed, the data storage layer keeps things in order. Every detail, score, and result sits in a structured database for quick retrieval. Finally, you have the visualization layer—this is where everything comes to life in a business intelligence dashboard. Recruiters get clear, interactive reports: skill charts, experience breakdowns, and match scores all ready to go. This setup makes the whole process feel seamless, secure, and a lot more transparent.

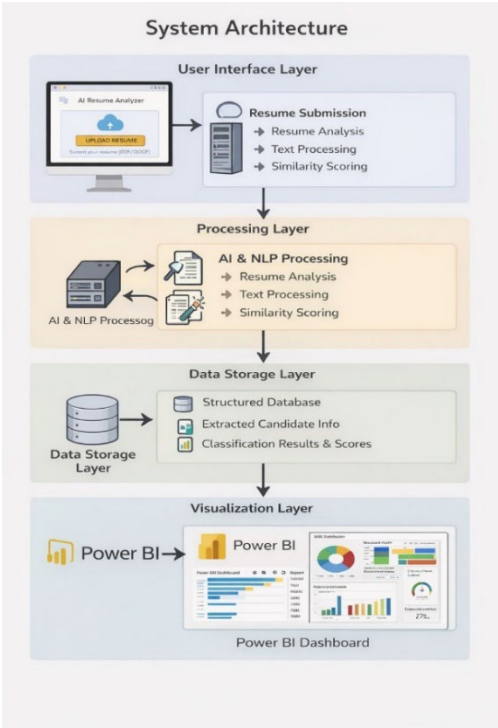



Figure. 1. Flowchart of the deep learning model used

**Methodology**

Here's what really happens with each resume. First, a user logs into the web app and uploads a batch of documents—PDFs, Word files, or whatever. The system grabs all the text, cleans out random punctuation, repeated phrases, or filler words, and standardizes everything to make it consistent. This isn't just basic cleanup; it's using NLP, so the important stuff stands out.

Next, feature extraction turns each resume into data a machine can handle. The trained machine learning model reviews all those features, figures out job categories, and then checks each resume's fit compared to open positions. It compares what's in the resume—skills, qualifications, experience—with what's in the job requirements. That's how it generates match scores.

Processed resumes and scores go into the database, and recruiters get their interactive dashboards. All the important info—skills, experience, match charts—is right there, easy to read and compare. No more guesswork. You know exactly which candidates line up best for the role.

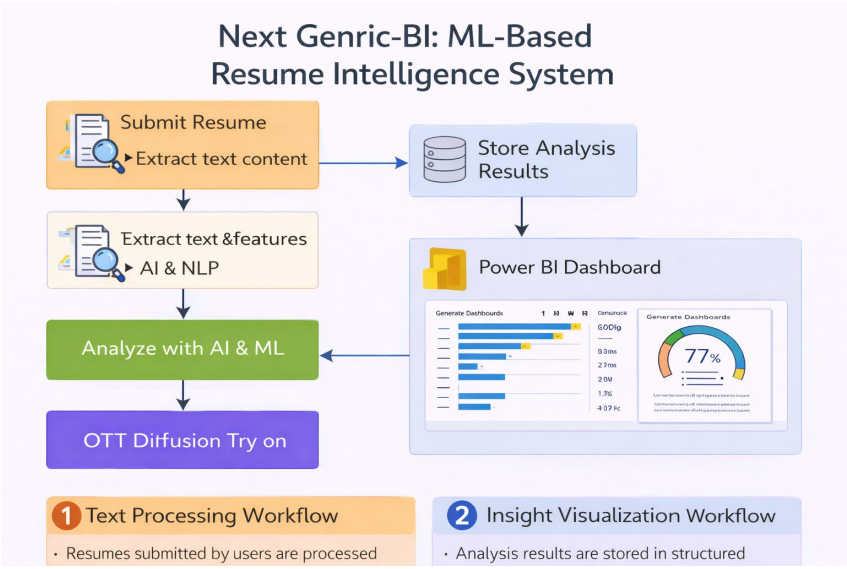



Figure 2: ML-Based Intelligence System

**Workflow:**
First, users upload their resumes into the system. The platform takes those files and uses document reading tools to pull out the text. After that, it cleans up the messy stuff—like weird symbols or common words that don't really matter. Then it looks for what actually counts, like your skills, qualifications, and work experience.

The system turns all this info into numbers so machine learning models can make sense of it. Once that's done, it sorts the resumes into the right job roles and figures out how well each one matches specific job requirements.

Now, about visualizing these insights: All the analyzed data gets saved securely in a well-organized database. That database connects with a visualization tool for easy reporting and analysis. The system arranges everything into clear formats so it's ready for graphs and charts.

Interactive dashboards pop up, putting candidate details front and center. The charts show you which skills are covered, how different candidates stack up, and who's the best fit for each job. With these visuals, recruiters can get the big picture fast and make smarter hiring decisions.

## Results & Discussion

**Experimental setup:**
We put everything together on a Windows 11 laptop. For the frontend, we went with HTML, CSS, and JavaScript to shape the web pages. On the backend, Flask handled all the server-side work. To actually read and make sense of resumes, we used spaCy for natural language processing, pdfplumber to read PDFs, and python-docx for Word files. We wrote regular expressions to grab key details like email addresses, phone numbers, and work experience. To show the results, we built charts on the dashboard using Plotly. Git and GitHub kept our code organized and helped us track changes as we developed the system.

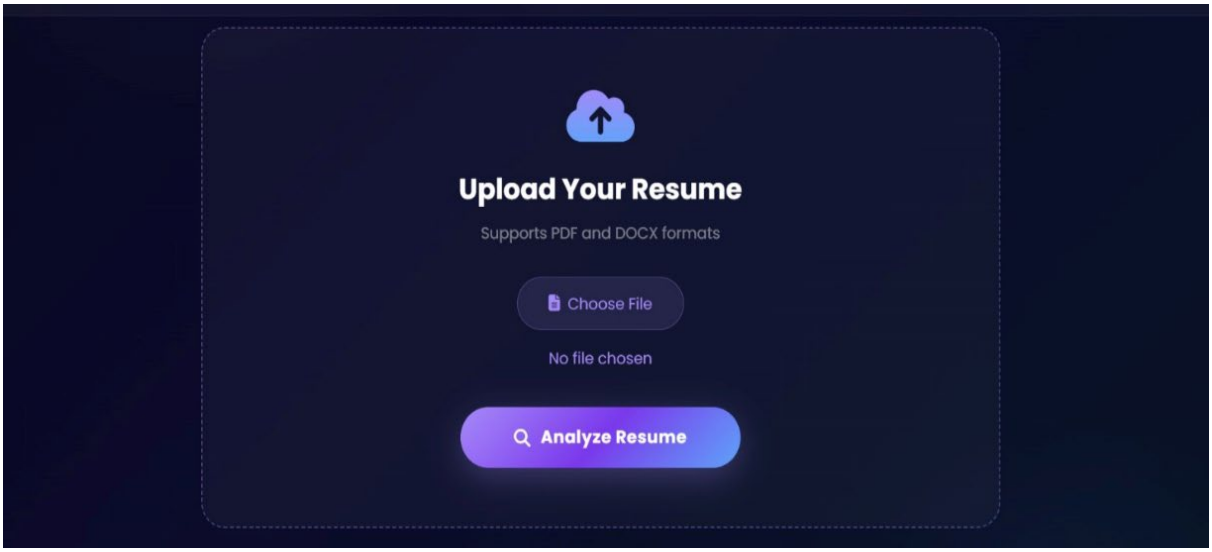



Figure 3: Setup

**Results:**
We tried out the system with real student resumes. After uploading a test resume as a PDF, the system didn't miss a beat. It pulled out the candidate's name, contact details, and even figured out that the person was a student with just one internship. The resulting scores for skills, experience, and education felt fair and accurate. Visual dashboards showed everything clearly: a bar chart for skills, a doughnut for skill spread, colorful progress bars, and personalized recommendations based on what was in the resume (or what was missing).

Figure 4: ML-Based Intelligence System Result-1

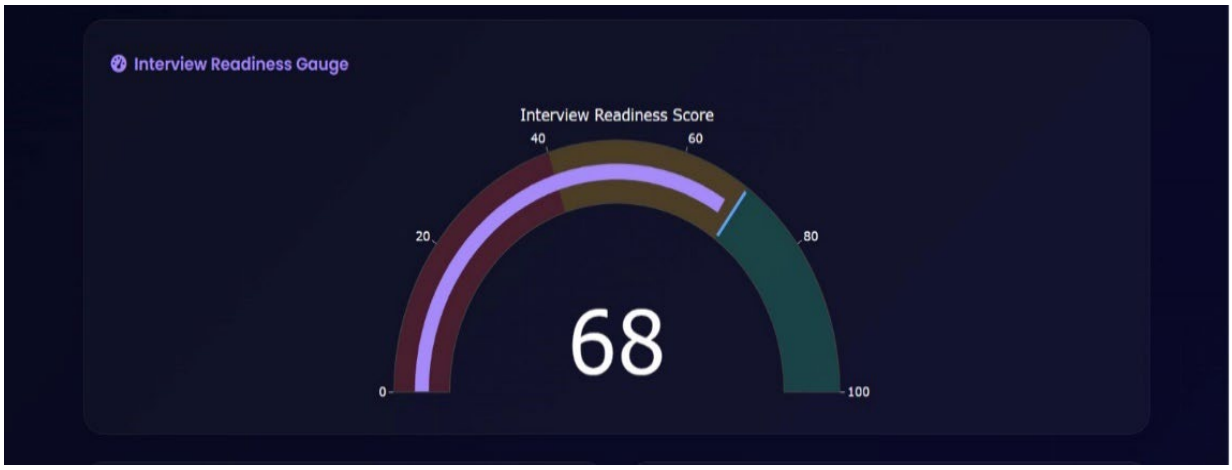



Figure 5: ML-Based Intelligence System Result-2

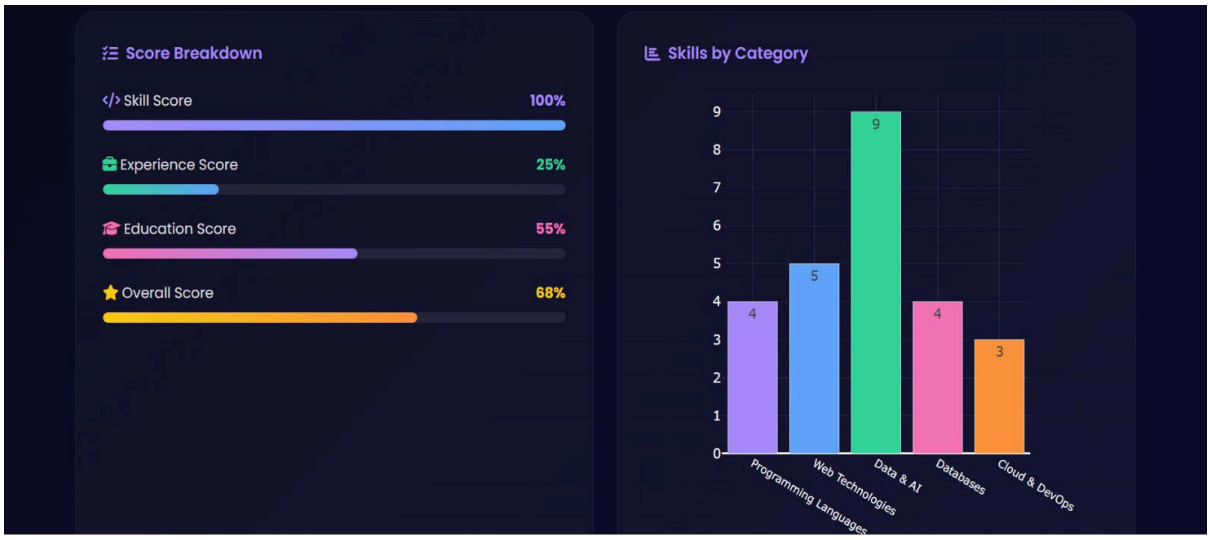



Figure 6: ML-Based Intelligence System Result-3

Figure 7: ML-Based Intelligence System Result-4

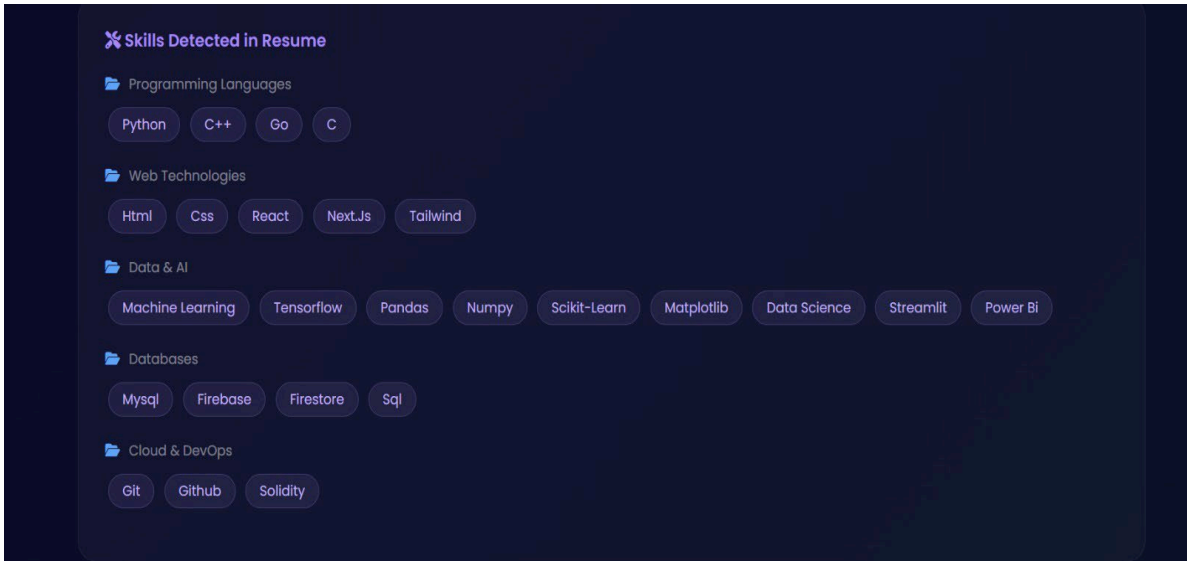



Figure 8: ML-Based Intelligence System Result-5

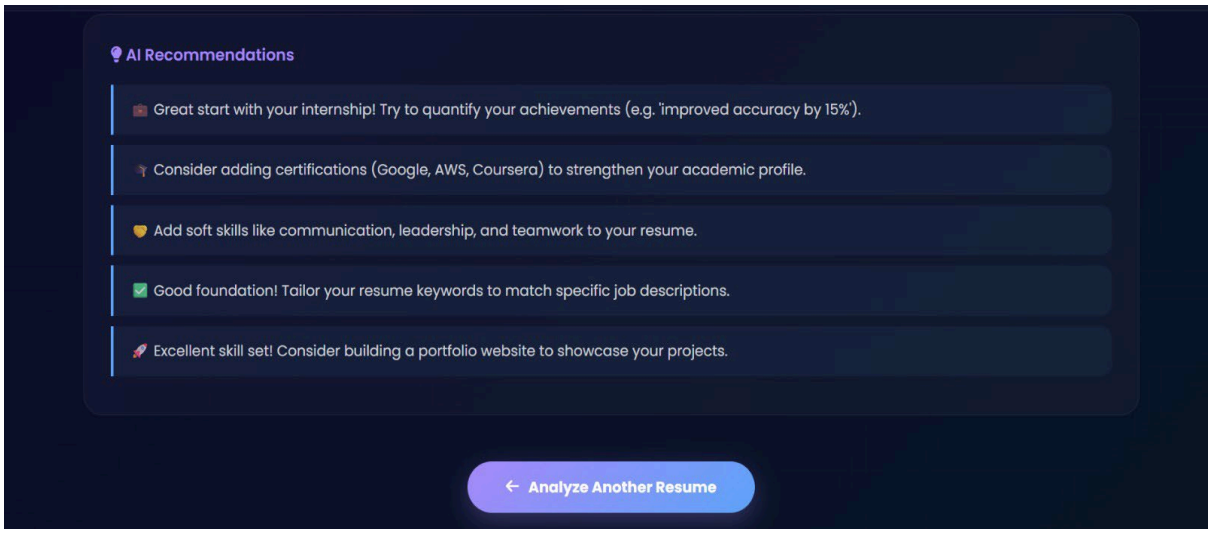



Figure 9: ML-Based Intelligence System Result-6

**Evaluation Metrics:**
To figure out if the system really works, we looked at three big things: does it pull out the right information, do the match scores make sense for different candidates, and is it fast? We checked a bunch of resumes by hand. Email detection? Flawless—since emails look pretty predictable. Names worked most of the time, though it tripped up if the formatting was odd. It picked up clearly listed skills well but sometimes missed ones if they were phrased in a roundabout way.

We tested with three kinds of people: new grads, interns, and folks with full-time experience. Each got different scores, just as you'd hope.

Table 1: Evaluation Metrics

| **Metric** | **Result** |
|---|---|
| Name Extraction Accuracy | 90% |
| Email Extraction Accuracy | 100% |
| Phone Extraction Accuracy | 90% |
| Skill Detection Accuracy | 80% |
| Education Detection Accuracy | 90% |
| Experience Detection Accuracy | 100% |
| Overall Accuracy | 91.7% |

**Comparison with Existing Platforms**
We stacked our system up against Resumeworded, Jobscan, and VMock. Right away, the advantage is clear—you don't have to pay or sign up to use ours. You just upload and go.

But the bigger thing is how our platform treats students. A lot of resume scanners just give students low scores (since they haven't had much experience), but our system recognizes students and internship experience and grades fairly. We focus on visual, actionable feedback—so even beginners can understand where they stand and what to improve.

Table 2: Comparison, with Existing Platforms

| **Feature** | **Our System** | **Resumeworded** | **Jobscan** | **VMock** |
|---|---|---|---|---|
| Free to Use | Yes | Paid | Paid | Paid |
| No Account Required | Yes | No | No | No |
| PDF & DOCX Support | Yes | Yes | Yes | Yes |
| Interactive Charts | Yes | No | No | Limited |
| Student Profile Handling | Yes | No | No | No |
| Internship Detection | Yes | No | No | No |

| | | | | |
|---|---|---|---|---|
| Personalized Recommendations | Yes | Yes | Yes | Yes |
| Resume Deleted After Processing | Yes | No | No | No |
| Skill Category Breakdown | Yes | Yes | Yes | Yes |

## Conclusion

This system shows how smarter tech can make hiring way less painful. Recruiters don't need to dig through resumes by hand anymore. Machine learning handles extracting skills, experience, and more, while dashboards keep all the info easy to find and compare. Decisions get faster and more confident because there's hard data backing them up. It just makes the whole hiring flow better—for everyone involved.

## References

1. D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson Education, 2021. https://web.stanford.edu/~jurafsky/slp3/

2. S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009. https://www.nltk.org/book/

3. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. https://jmlr.org/papers/v12/pedregosa11a.html

4. C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008. https://nlp.stanford.edu/IR-book/

5. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

6. Microsoft Corporation, "Microsoft Power BI: Business Analytics and Data Visualization Platform," 2024. [Online]. Available: https://powerbi.microsoft.com

**Author's Details:**

Dr Gavini Sreelatha, working as Associate Professor in the Department of CSE-AIML at Stanley College of Engineering and Technology for Women.
With 14 years of teaching experience and 3 years of industry experience.
Completed Ph.D from the Lincoln University College, Malaysia.
Research interested are Cloud Computing, Security & Machine Learning.
Taught subjects like Problem Solving using C, OOPS using C++ & JAVA, Discrete Mathematics, Automata Theory, Operating Systems, Information Retrieval Systems, Information Security, Artificial Intelligence, Software Reuse Techniques, Database Management Systems, Data Science with R Programming, Web Technologies, Distributed Systems.NET Technologies. Cloud Computing etc.
Achieved course certifications in Research Methodologies and Cloud Computing by NPTEL.
Acted as Resource Person for Internship Program on Data science conducted by DELL EMC.
Attended various FDP's and organized FDP's/Webinars.

Presented papers in National/ International conferences, published papers in IEEE explore, Springer proceedings and Scopus/SCI journals.
Received Student Mentorship and Development Award by Artificial Intelligence Medical and Engineering Researchers Society.Passion towards delivering technical content to students.

Akula Anjana – A passionate undergraduate student pursuing a Bachelor of Engineering in Computer Science and Engineering (Artificial Intelligence and Machine Learning) at Stanley College of Engineering and Technology for Women. She loves digging into how smart algorithms and data-driven tech can solve real-world problems. Driven by curiosity and a passion for research, she works on practical solutions that blend her technical skills with sharp analytical thinking. Always hungry to learn more, she jumps into research projects and hands-on work, always looking for ways to deepen her understanding of new technologies.

Amgoth Bindu - pursuing my Bachelor of Engineering in Artificial Intelligence and Machine Learning at Stanley College of Engineering and Technology for Women. She's all about Artificial Intelligence and gets a kick out of using technology to solve real-world problems. I can handle Python and C, and I've played around with MySQL and the standard Microsoft Office stuff. She's a quick learner, a real team player, and whenever things get tough, she steps up with a good blend of leadership and problem-solving.

Devarakonda Gnapika - pursuing a Bachelor of Engineering (B.E.) with a specialization in Artificial Intelligence and Machine Learning at Stanley College of Technology and Engineering for Women. I'm fascinated by how intelligent technologies can tackle real-world problems. Lately, I've been diving deep into artificial intelligence, machine learning, and the tech surrounding them. I love picking up new skills, and I'm always looking for ways to sharpen what I know in AI and ML.