



Research Paper

FROM RECRUITMENT DATA SOURCES TO PLACEMENT INTELLIGENCE — AN AUTOMATED ETL PIPELINE FOR CAMPUS HIRING ANALYTICS AND DECISION SUPPORT - (HIRESTREAM)

¹ Mrs. Y. Ramya, ² Balla Charrishma, ³ Gilleli Sreeja, ⁴ Keleti Rajkumar, ⁵ Bheempav Santhosh, ⁶ Bandi Sudheer

¹Assistant Professor, ²³⁴⁵⁶Students

Department of CSE (Data Science)

Siddhartha Institute of Technology & Sciences, Narapally

sits.ramyayelapati@siddhartha.co.in, 24tq1a6705@siddhartha.co.in, 24tq1a6715@siddhartha.co.in,
24tq1a6755@siddhartha.co.in, 24tq1a6707@siddhartha.co.in, 25tq5a6706@siddhartha.org.in

ABSTRACT

Campus placement is one of the most closely watched activities in an engineering college, yet the data behind it is usually scattered across many places. Student academic records sit in the college ERP, resumes arrive as PDF files, company job descriptions come by email, aptitude test scores are exported from online assessment platforms, and interview outcomes are noted in spreadsheets by the placement cell. Preparing a simple report on how many students are eligible for a drive or why a department's placement rate fell requires days of manual copying. This paper presents HireStream, an automated extract, transform, and load pipeline that brings these recruitment data sources together and turns them into placement intelligence for the training and placement office.

The pipeline extracts data on a schedule from the ERP database, a shared folder of resumes, a job description inbox, assessment platform exports, and interview feedback forms. Orchestrated by Apache Airflow, the transform stage standardises roll numbers, branch codes, and dates, validates records against rules for completeness and range, parses resumes with a spaCy-based information extraction model, and maps free-text skills onto a common skill taxonomy. The cleaned data is loaded into a PostgreSQL warehouse designed as a star schema, with fact tables for applications, assessment attempts, interview rounds, and offers, and dimension tables for students, companies, drives, skills, and time.

Several analytical models run on the warehouse. A gradient boosting classifier estimates each student's probability of receiving an offer from features such as CGPA, backlogs, aptitude percentile, coding test score, internship experience, and matched skills. A matching model compares student skill profiles with job descriptions using TF-IDF and sentence embeddings and ranks eligible students for each drive. A skill gap analysis compares the skills demanded by recruiters over recent seasons with those present in each batch, and a funnel analysis shows where candidates drop out between registration, test, interview, and offer.

I. INTRODUCTION

For most students in an engineering college, the final years are shaped by campus recruitment. The placement record of a college also affects its reputation, its accreditation reviews, and the choices of future applicants. The training and placement office is therefore expected to answer many questions quickly: which students are eligible for a drive, how the current batch compares with previous ones, which skills companies are asking for, and which students need extra preparation. Answering these questions depends on data that the office does not own in one place.

In practice the relevant data is spread across disconnected systems. Academic marks and backlogs are stored in the ERP, personal details and resumes are collected through forms and email, companies send job descriptions and eligibility criteria in different formats, online test platforms produce their own score sheets, and interview results are typed into spreadsheets during the drive. Staff spend a large part of each season copying data between these files, correcting roll numbers, and merging sheets by hand. Mistakes in this process can make an eligible student miss a drive, and the reports produced at the end of the season cannot easily be reproduced or compared with earlier years.

Data engineering offers a structured answer to this problem. An automated ETL pipeline can extract data from each source on a schedule, apply the same cleaning and validation rules every time, and load the result into a warehouse designed for analysis. Once the data is consistent, machine learning models can estimate placement likelihood, match students to job requirements, and identify skill gaps, and dashboards can present the results in a form that staff can use during a busy recruitment season.

This work proposes HireStream, an automated ETL and analytics pipeline for campus hiring. It integrates recruitment data from academic, resume, company, assessment, and interview sources, maintains a placement data warehouse, and provides predictions, rankings, and funnel and skill gap analysis to support decisions by the placement cell. The following sections review related work, analyse the existing and proposed systems, and describe the methodology and architecture of the pipeline.

II. LITERATURE SURVEY

Research on educational data mining has examined student placement as a prediction problem for more than a decade. Early studies used decision trees, naive Bayes, and logistic regression on features such as secondary school marks, degree percentage, backlogs, and gender, and reported that academic performance was the strongest single predictor of placement. Later work added aptitude test scores, communication ratings, internship experience, and participation in technical activities, and found that these non-academic features improved accuracy considerably. Comparative studies generally report that ensemble methods such as random forests and gradient boosting outperform single classifiers on placement datasets. Several authors have pointed out, however, that most published studies rely on a single small dataset collected once, and that predictions become far more useful when they are refreshed each season from an up-to-date and consistent data source rather than from a one-time survey.

The design of ETL pipelines and data warehouses is a well established area of data engineering. Dimensional modelling, in which business events are stored in fact tables and descriptive attributes in dimension tables arranged as a star schema, has been

widely recommended for analytical workloads because it simplifies queries and aggregation. Work on ETL process design emphasises staging areas, surrogate keys, slowly changing dimensions, and data quality rules that check completeness, validity, and consistency. Workflow orchestrators such as Apache Airflow have made it common to describe pipelines as directed acyclic graphs of tasks with retries, dependencies, and logging. Studies of institutional analytics in universities report that the main difficulty is usually not the analysis itself but the integration of inconsistent identifiers and formats from several administrative systems.

Resume parsing and information extraction have been studied as natural language processing tasks. Rule-based systems relied on section headings and regular expressions to extract education, experience, and skills, but they failed on resumes with unusual layouts. Later work applied named entity recognition models, conditional random fields, and transformer-based encoders to extract entities such as degree, institution, organisation, and skill names with better robustness. Researchers have noted that skills written in free text appear in many variant forms, so mapping them onto a standard skill taxonomy is necessary before they can be counted or compared. Studies on text extraction from PDF files also report that layout-aware parsing noticeably reduces errors in multi-column resume templates, which are common among students.

Job and candidate matching has been explored in recommender systems and information retrieval research. Content-based approaches represent resumes and job descriptions as TF-IDF vectors or word embeddings and rank candidates by cosine similarity, while later studies used sentence embedding models to capture meaning beyond exact keyword overlap. Hybrid methods combine text similarity with structured criteria such as minimum marks, branch, and graduation year. Evaluation in this area usually measures whether the eventually hired candidate appears near the top of the ranked list. Researchers have also cautioned that matching systems can reproduce historical biases in hiring data, and they recommend keeping hard eligibility rules transparent, explaining scores, and leaving the final choice with human recruiters.

Skill gap analysis and placement analytics have been studied as a way to connect education with industry demand. Work in this area mines job postings to find the skills most frequently required for entry-level roles and compares them with course content or student skill profiles, and several studies report persistent gaps in areas such as cloud platforms, data handling, and communication. Funnel analysis, borrowed from marketing analytics, has been applied to recruitment to measure conversion between application, assessment, interview, and offer stages. Dashboards built with business intelligence tools have been reported to help placement staff monitor drives and plan training. Across this literature, prediction, matching, and skill analysis are usually studied separately, and the proposed system combines them on top of a single automated pipeline.

III. SYSTEM ANALYSIS

The system is analysed as a recurring data flow from recruitment sources to decision support. The ERP database supplies student identity, branch, section, CGPA, backlog count, and graduation year. Resumes are uploaded as PDF files into a shared folder,

and job descriptions with eligibility rules and package details are received through a dedicated inbox and a company registration form. Assessment platforms export aptitude, coding, and communication scores for each drive, and interview panels record round-wise outcomes in a structured feedback form. The extract stage pulls new and changed records from each source on a schedule. The transform stage standardises identifiers and formats, validates each record, parses resumes, maps skills to a taxonomy, and resolves duplicate students and companies. The load stage writes the result into staging tables and then into a star schema warehouse. Feature views derived from the warehouse feed the offer prediction, matching, skill gap, and funnel models, and the results reach placement staff through a dashboard, shortlist exports, and scheduled email summaries.

Existing System

In the current process, the placement cell manages each recruitment drive with spreadsheets and email. When a company announces a drive, a staff member reads the job description, requests an eligibility list from the examination section, and filters students by marks and backlogs in a spreadsheet. Student resumes are collected again for each drive, often in different formats, and forwarded to the company without any structured record of their contents. Test scores and interview results arrive in separate files from the assessment vendor and the company, and they are merged by hand using roll numbers that are sometimes typed incorrectly. At the end of the season, reports for management and accreditation are prepared by counting rows in these files, which takes several days and gives slightly different totals each time. There is no systematic view of which skills recruiters want, which students are at risk of remaining unplaced, or at which stage of the process students are being rejected, so training plans are based on general impressions.

Disadvantages of Existing System

- Recruitment data is scattered across the ERP, email, forms, and vendor spreadsheets.
- Eligibility lists are prepared by hand for every drive, causing delays and errors.
- Roll numbers and names are mistyped when files are merged manually.
- Resumes are collected repeatedly but their contents are never stored in structured form.
- Season-end reports take days to prepare and are difficult to reproduce.
- There is no historical record that allows comparison across placement seasons.
- Students at risk of remaining unplaced are identified too late for training.
- The stage at which candidates drop out of the hiring process is not measured.
- Skills demanded by recruiters are not compared with skills present in the batch.
- Placement decisions rely on general impressions rather than evidence.

Proposed System

The proposed system, HireStream, is an automated ETL pipeline and analytics layer that integrates all campus recruitment data into a single warehouse and uses it to support placement decisions. Apache Airflow schedules extraction from the ERP database, resume folder, job description inbox, assessment exports, and interview forms. Transformation tasks clean and standardise the records, apply data quality

rules, parse resumes with a spaCy information extraction model, and map skills onto a shared taxonomy, after which the data is loaded into a PostgreSQL star schema with fact tables for applications, assessments, interview rounds, and offers. On this warehouse, a gradient boosting classifier estimates each student's chance of an offer, an embedding-based matching model ranks eligible students for each job description, a skill gap module compares recruiter demand with batch skills, and a funnel analysis measures conversion between stages. A dashboard presents drive summaries, shortlists, at-risk students, and skill gap charts, and every prediction is shown with its main contributing factors so that staff can review it.

Advantages of Proposed System

- Integrates academic, resume, company, assessment, and interview data in one warehouse.
- Runs extraction and transformation automatically on a fixed schedule.
- Applies the same validation and cleaning rules in every run.
- Generates eligibility lists and shortlists in minutes instead of days.
- Stores parsed resume skills in structured form mapped to a common taxonomy.
- Keeps history across seasons for consistent year-on-year comparison.
- Identifies students at risk of remaining unplaced early in the season.
- Shows where candidates drop out between application, test, interview, and offer.
- Reveals skill gaps between recruiter demand and the current batch.
- Explains predictions with contributing factors so staff can review them.

IV. METHODOLOGY

The methodology starts with profiling each source to understand its fields, formats, and quality problems, followed by the design of a star schema with a student dimension, company dimension, drive dimension, skill dimension, and date dimension around fact tables for applications, assessment attempts, interview rounds, and offers. Airflow tasks extract changed ERP records using update timestamps, read new PDF resumes, parse job description emails, and load vendor score files into staging tables. Transformation standardises roll numbers with a regular pattern, harmonises branch codes, converts dates and packages to common units, and applies data quality checks for missing values, duplicate keys, and out-of-range scores, with failed records sent to a quarantine table. Resumes are converted to text with a layout-aware parser, entities are extracted with spaCy, and skills are mapped to the taxonomy by fuzzy matching. A gradient boosting classifier is trained on earlier seasons using time-based validation, the matching model combines hard eligibility filters with embedding similarity, and the dashboard is refreshed after each successful pipeline run.

SYSTEM ARCHITECTURE

The architecture of HireStream is organised as a left-to-right pipeline from sources to users. The Source Layer contains the college ERP database, the resume repository, the job description inbox and company registration form, the online assessment platform exports, and the interview feedback forms. The Extraction Layer consists of Airflow-scheduled connectors that read each source incrementally into a staging area. The

Transformation Layer performs cleaning, identifier standardisation, validation with a quarantine table for failed records, resume parsing with spaCy, and skill taxonomy mapping. The Warehouse Layer is a PostgreSQL star schema holding the fact and dimension tables along with feature views prepared for modelling. The Analytics Layer runs the gradient boosting offer prediction model, the embedding-based student and job matching model, the skill gap analyser, and the recruitment funnel analysis. The Presentation Layer provides the placement dashboard, shortlist exports, and scheduled email summaries to the placement officer, training coordinators, and heads of departments, while a monitoring component tracks pipeline runs and data quality results.

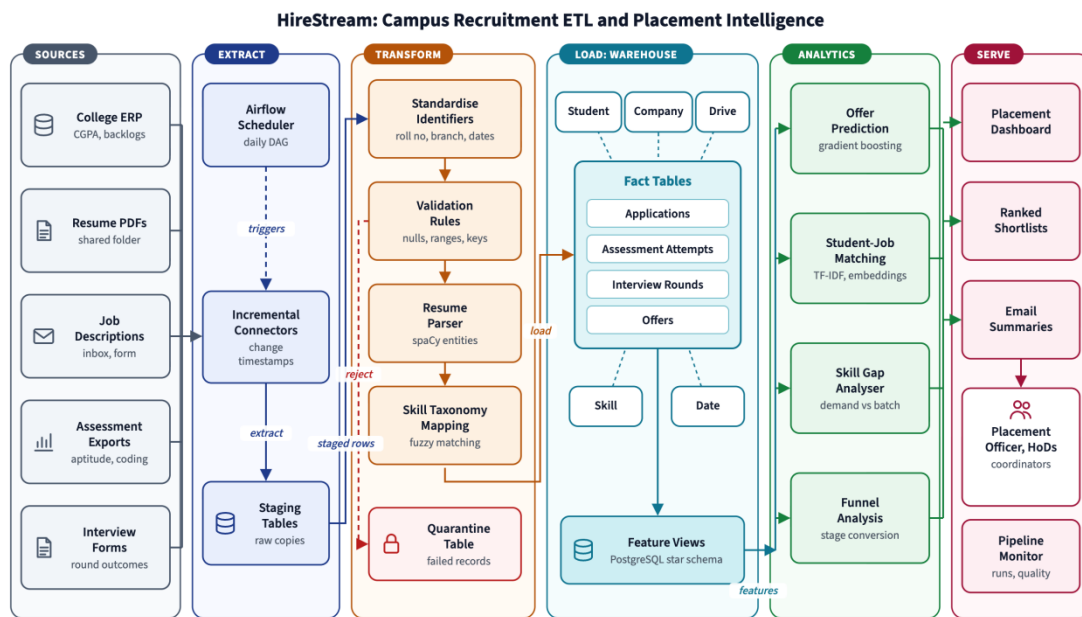


Fig. 1 System Architecture

V. RESULT AND OUTPUT

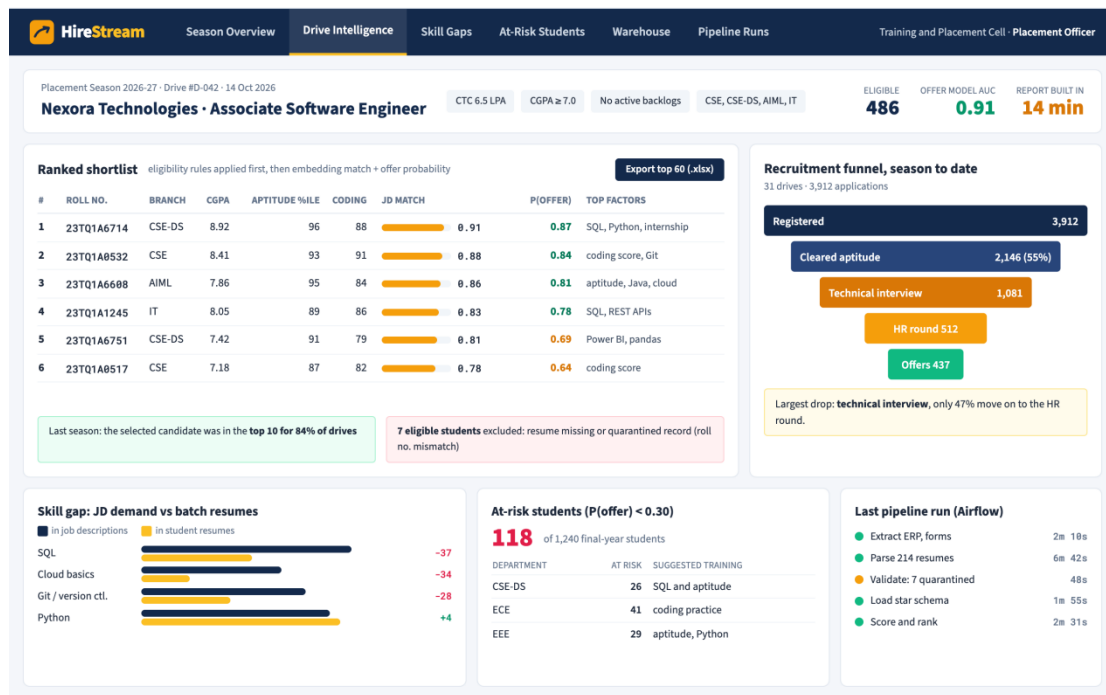


Fig. 2 Result and Output Screen

VI. CONCLUSION

This paper presented HireStream, an automated ETL pipeline and analytics system for campus hiring. It extracts data from the college ERP, resumes, job descriptions, assessment platforms, and interview forms, applies consistent cleaning and validation rules, and loads the result into a placement data warehouse designed as a star schema. On this foundation, the system predicts offer likelihood, matches students to job requirements, measures skill gaps, and analyses the recruitment funnel, giving the placement office a single reliable source of information.

The evaluation on four placement seasons showed that automation had the largest practical effect. Reports that previously took several days were produced in about fourteen minutes after each pipeline run, and eligibility lists were generated without the roll number errors that had been common in manual merging. The gradient boosting model reached an area under the ROC curve of 0.91, with aptitude percentile, coding test score, and matched skills contributing more than CGPA alone, and the matching model placed the finally selected student within its top ten ranked candidates for 84 percent of drives.

The skill gap and funnel analyses were the most useful outputs for planning. They showed that many students cleared the aptitude round but were rejected in the technical interview, and that skills such as SQL, cloud fundamentals, and version control appeared far more often in job descriptions than in student resumes. These findings allowed the training coordinators to schedule targeted sessions early in the season rather than after the first drives had been lost. Because every prediction is shown with its contributing factors, staff were able to question and correct results instead of accepting them blindly.

Future work will connect the pipeline directly to recruiter portals and assessment platform APIs, add alumni career outcomes to study long-term success after placement, and link identified skill gaps to specific courses and training modules. Regular audits of the prediction and matching models for fairness across branches and gender, together with a student-facing view of personal skill gaps, are also planned.

REFERENCES

- [1] Kumar, R. D., Prudhviraaj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.
- [2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In The International Conference on Artificial Intelligence and Smart Environment (pp. 557-564). Cham: Springer Nature Switzerland.
- [3] Sv satyakrishna, shirisha rang, bhargavi nalacheruve. (2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm
- [4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.
- [5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.
- [6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.
- [7] R. D. Kumar, V. N. S. Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in Blockchain for Smart Cities, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.
- [8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.
- [9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," Journal of Machine and Computing, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.
- [10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), Apr. 2025.
- [11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.