From Prediction to Action: A Review of Machine Learning for Employee Attrition and AI-Driven Workflow Automation in HR Follow-Up

Vedant Keshavrao Deshmukh[1], Ajinkya Manoj Deshmukh[2], Kartik Nilesh Raghuwanshi[3], Anurag Sandeep Shrikhandkar[4], Dr Kalyani Hemant Deshmukh[5]

[1–4]B. Tech Final-Year Students, Department of Computer Science and Engineering
[5]Department of Computer Science and Engineering
[1–5]Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati, Maharashtra, India

vedantdeshmukh975@gmail.com, ajinkya71005@gmail.com, raghuwanshikartik521@gmail.com, anuragshrikhandkar@gmail.com, khdeshmukh@mitra.ac.in

***Abstract*—Employee attrition prediction is a well-studied application of machine learning (ML), and the public IBM HR Analytics Employee Attrition and Performance dataset has become its most common benchmark. Most studies, however, stop at classification performance and leave unanswered what an organisation should do with a risk score. This review examines three connected areas: (i) ML approaches to attrition prediction on the IBM dataset, including gradient boosting, ensemble methods and deep learning; (ii) evaluation practice, including class imbalance, resampling order, and precision-recall analysis; and (iii) the emerging use of workflow automation platforms such as n8n, together with calendar services, to turn predictions into HR follow-up actions under human review. We also discuss explainability, fairness, privacy and human oversight, and we use a recent prototype that links a gradient boosting classifier to a Google Calendar scheduling workflow as an illustrative case. The literature we examined shows that results across studies are not directly comparable, that evaluation of the prediction-to-action step is largely missing in the sources examined, and that responsible-use safeguards need explicit testing. We close with a research agenda. The review is non-systematic and covers a limited set of sources; its conclusions describe those sources and should not be read as a claim about all existing work.**

***Keywords*—employee attrition prediction, gradient boosting, HR analytics, workflow automation, n8n, human-in-the-loop.**

I. INTRODUCTION

Employee attrition is the departure of employees from an organisation, either voluntarily (the employee decides to leave) or involuntarily (the employer ends the contract) [1], [2]. Replacing a departed employee costs time and accumulated organisational knowledge, which gives human resource (HR) teams a reason to identify employees at elevated risk of leaving before the exit happens. ML has been applied to this problem with a range of data sources, and the public IBM HR Analytics Employee Attrition and Performance dataset has become a widely used benchmark [1]-[4].

Prediction alone has an organisational limit. A risk score in a table does not decide whom to contact, does not arrange a conversation, and does not record an outcome. Studies on this dataset usually report classification performance [1]-[3], and the step that turns a score into an HR action typically remains outside the experiment. At the same time, practitioners have begun to combine attrition analysis with low-code workflow tools [12], [13], but without reported evaluation.

This review therefore asks four questions:

- **RQ1.** Which ML methods have been applied to attrition prediction on the IBM dataset, and what do their results actually allow us to conclude?
- **RQ2.** How is evaluation designed, and which methodological pitfalls recur?
- **RQ3.** How have workflow automation and calendar services been combined with attrition analysis, and how is this step evaluated?
- **RQ4.** What ethical, privacy and fairness requirements apply when a score leads to contact with a real person?

The contributions of this review are (i) a structured comparison of the sources examined; (ii) a synthesis of evaluation pitfalls; (iii) a framework separating prediction quality from workflow reliability; and (iv) a research agenda. The paper follows the IMRaD structure required by the journal. Section II reviews the literature, Section III describes the review method and analytical framework, Section

IV presents the synthesised findings, Section V discusses implications, gaps and future directions, and Section VI sets out limitations and conclusions.

II. LITERATURE REVIEW

A. Attrition Prediction on the IBM Dataset

The dataset is distributed through Kaggle and is described by its distributors as fictional data created by IBM data scientists [4]. Other studies describe it as 1,470 records with 35 attributes, of which 237 (16.1%) carry the label Attrition = Yes [1], [2]. Because the data are fictional, results describe behaviour on this benchmark and do not establish performance in a real organisation. Its popularity makes cross-study comparison tempting, but, as Sec. II-C shows, differences in protocol make such comparison unreliable.

Raza et al. compared four classifiers on the dataset and found that a tuned Extra Trees classifier reached 93% accuracy, with monthly income, hourly rate, job level and age identified as the most influential factors [1]. This kind of result is typical of studies that emphasise accuracy. As discussed below, the way resampling was ordered in that study may inflate the held-out score.

Li compared a Transformer-based network for tabular data with gradient boosting, XGBoost, random forest and extremely randomised trees, using a stratified 85:15 split of 1,470 records, of which 237 (16.1%) were leavers; the Transformer obtained higher ROC-AUC and PR-AUC values [2]. In that comparison, gradient boosting obtained a ROC-AUC of 0.6929 and a PR-AUC of 0.3590 on one test split of 221 records containing 36 leavers. These values belong to that particular split and should not be treated as a general benchmark. Nandal et al. applied ML, ensemble and deep learning methods to the same data and found that a feedforward network gave the best accuracy, recall and F1-score [3].

B. Gradient Boosting

Friedman described boosting as stagewise gradient descent in function space and developed gradient boosting of regression trees for regression and classification [5]. Shallow trees are added sequentially, each fitted to the residual errors (pseudo-residuals) of the ensemble built so far. After encoding, the method handles a mixture of numerical and categorical attributes, which suits the tabular structure of HR data, and mature implementations exist in libraries such as scikit-learn [6]. Its role in the attrition literature is mostly as a conventional baseline [2].

In its simplest form the ensemble is built by the update in Eq. (1), where F_m is the model after m trees, h_m is the m-th regression tree fitted to the negative gradient of the loss with respect to the current predictions, γ_m is a step size and ν is the learning rate [5]. For binary classification the logistic loss is used, and the final model output is converted into a score between 0 and 1 by the logistic function σ, as in Eq. (2).

$$F_m(x) = F_{m-1}(x) + \nu\gamma_m h_m(x) \quad (1)$$

$$p(x) = \sigma(F_M(x)) = 1 / (1 + e^{-FM(x)}) \quad (2)$$

Two properties matter for the present review. First, the learning rate, the number of trees and the tree depth jointly control model complexity, so tuning has to be done inside training folds to avoid optimistic estimates. Second, the score in Eq. (2) is a ranking signal; it should not be read as a calibrated probability unless calibration has been evaluated.

C. Evaluation Practice and Methodological Pitfalls

1) Resampling order and leakage

Oversampling techniques such as SMOTE create synthetic records from existing ones. If oversampling is applied before the train-test split, synthetic samples derived from records similar to test data can enter the training set and inflate held-out scores. Study [1] describes oversampling as taking place before the 85:15 split. Resampling should be confined to training folds, and all preprocessing should be fitted on training data only, for example inside a single pipeline applied identically to training and test records.

2) Class imbalance and metric choice

With roughly one leaver in six [2], a model that predicts no attrition for everyone is right most of the time yet detects no leavers. Accuracy must therefore be read against the majority-class baseline. Precision-recall analysis can be more informative than ROC analysis when the positive class is a minority [7], and it is sensitive to the prevalence of that class. A useful reporting set includes precision, recall, F1-score, the confusion matrix, ROC-AUC and PR-AUC, all computed on a held-out test set only.

3) Single splits, small test sets and thresholds

A single stratified split with a few dozen positive test records yields unstable estimates. Cross-validated comparison with confidence intervals is preferable. In addition, the threshold at which a score becomes an action is a design decision that changes how many employees are flagged and how many leavers are missed; reporting metrics at only one default cutoff (commonly 0.50) hides this trade-off.

4) Calibration

A boosted-tree score is a ranking signal and should not be described as a calibrated probability unless

calibration has been evaluated. This matters when scores are mapped onto risk tiers or shown to decision-makers.

D. Explainability and Fairness

Feature attribution methods such as SHapley Additive exPlanations (SHAP) can explain individual predictions [8]. For attrition, explanations matter in two ways: they help HR reviewers judge whether a flag is plausible, and they expose reliance on attributes that may act as proxies for protected characteristics. Surveys of bias in ML describe how choices about data, modelling and deployment can lead to unfair outcomes for particular groups [9]. False positives expose employees to unwarranted attention, whereas false negatives leave at-risk employees unnoticed [9]. These concerns carry more weight when a score leads to contact with a real person. The attrition studies summarised in Table I focus on predictive performance; we did not identify a fairness assessment across groups or a test of how reviewers use explanations among their reported elements, and this should be verified in a broader search.

E. Workflow Automation and HR Decision Support

A classifier becomes useful to an organisation only when its output is connected to a decision process: who reviews the flag, what action follows, how duplicates and delays are handled, and how the outcome is recorded. Workflow automation platforms such as n8n can link a model output to calendar services, which allows this step to be described and tested explicitly.

The Google Calendar node of n8n supports checking availability and creating, retrieving, updating and deleting events [10]. The Google Calendar API can create an event and, if the conference data version parameter is set, can also request a Google Meet conference for that event [11]. These capabilities are sufficient to automate the scheduling of a follow-up conversation once a recommendation has been approved.

Practitioner material that has not been peer reviewed shows that these elements are already being combined with attrition analysis. One community n8n template scores attrition risk with a language model and e-mails HR [12], and a public repository uses n8n to generate HR reports from an attrition analysis [13]. In the material reviewed, neither source reports an evaluation of scheduling reliability.

Human oversight is the common thread in these concerns. In the design discussed later in this review, flagged recommendations pass through an HR dashboard where a reviewer can confirm or snooze them before any external action, and access to scores is limited by role. These are procedural controls; the literature we examined does not evaluate how well they work in practice, which is a gap for future studies [9].

F. Comparison and Research Gap

Table I summarises the sources discussed above. It rests on a limited, non-systematic web and database search carried out in October 2026 (Sec. III-A). It therefore describes the sources we examined and does not show that no other work exists.

TABLE I. COMPARISON OF SELECTED RELATED WORK

Ref.	Data and method	Main finding	Limitation
[1]	IBM data (1,470 records, 35 features); SVM, LR, DTC, tuned Extra Trees; SMOTE; 10-fold CV	Extra Trees reported 93% accuracy; income, hourly rate, job level, age highlighted	Oversampling described before the split; accuracy focus; prediction only
[2]	IBM data; stratified 85:15 split; Transformer vs. GB, XGB, RF, ERT	Transformer obtained higher ROC-AUC and PR-AUC than the baselines	Small test set (221 records, 36 leavers); no workflow
[3]	IBM Watson HR data; ML, ensemble and deep learning	Feedforward network reported best accuracy, recall and F1	Prediction only; protocol not comparable across studies
[12]	n8n template; LLM risk scoring, threshold routing, e-mail alert	Automated alert with recommended actions	Not peer reviewed; no scheduling; no reported evaluation
[13]	Employee CSV; Pandas analysis, n8n report generation	Automated HR report	Not peer reviewed; report only; no scheduling

SVM: support vector machine; LR: logistic regression; DTC: decision tree classifier; SMOTE: synthetic minority oversampling technique; CV: cross-validation; GB: gradient boosting; XGB: XGBoost; RF: random forest; ERT: extremely randomised trees; CSV: comma-separated values; LLM: large language model.

Among the sources examined, peer-reviewed attrition studies report prediction performance, while practitioner n8n material reports alerts or reports. We found no source that uses defined metrics to evaluate the path from a classifier output, through human review, to a calendar event. This review addresses that narrower gap. It does not claim that attrition prediction or HR workflow automation is unexplored.

III. METHODOLOGY

A. Review Design and Scope

This is a narrative review based on a limited, non-systematic web and database search carried out in October 2026. It covers peer-reviewed studies using the IBM dataset, foundational methodological references (gradient boosting, scikit-learn, precision-recall analysis, SHAP, bias and fairness),

official documentation for the n8n Google Calendar node and the Google Calendar API, and two practitioner sources (a community n8n template and a public code repository). Practitioner sources were not peer reviewed and are treated as precedent, not as evidence of effectiveness.

Because the search was not systematic, the review describes the sources examined and does not show that no other work exists. Reported figures are taken from the cited sources as stated; they are not recomputed.

No personal data were collected for this review. The illustrative case in Sec. IV-E used the public, fictional IBM dataset, and all company names, employee titles and e-mail addresses in its interfaces, payloads and invitations are synthetic placeholders [14].

For every source, the following items were extracted where available: the data and setting, the model or method, the split and resampling procedure, the metrics reported, whether any step after the prediction was evaluated, and any stated safeguards for privacy, fairness or human oversight. Extracted items were compared in tabular form (Table I) and then synthesised narratively by research question. Because the sources differ in protocol, no pooled estimate was attempted.

B. Analytical Framework

To answer RQ1 to RQ4 consistently, each source was read against a two-part framework that keeps prediction quality and workflow reliability separate, together with a set of responsible-use criteria. Fig. 1 shows the pipeline and the two evaluation tracks.

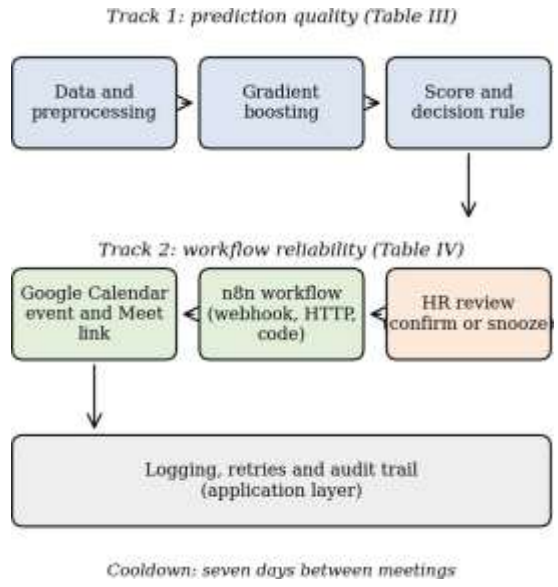


Fig. 1. Attrition-to-action pipeline with two separate evaluation tracks

Table II frames the attrition-to-action pipeline as seven stages and links each stage to the evaluation concern discussed in the literature. The stages are the unit of analysis for the rest of the review.

TABLE II. STAGES OF THE ATTRITION-TO-ACTION PIPELINE

Stage	Evaluation concern	Basis
Data and preprocessing	Leakage; consistent transformation of training and test records	[1], Sec. II-C
Model training	Documented split; tuning and resampling inside training folds	[1], [2]
Scoring and thresholds	Calibration; precision and recall at each cutoff	[7]
Explanation and fairness	Feature attribution; group-wise outcomes	[8], [9]
Human review	Confirm or snooze before contact; reviewer workload	Sec. III-B3
Workflow and calendar	Event creation, Meet link, latency, duplicates	[10], [11]
Logging and retries	Audit trail; retention limit; error handling	Sec. III-B3

1) Prediction evaluation criteria

Table III summarises the minimum protocol used as the yardstick for the prediction studies. It reflects the pitfalls discussed in Sec. II-C.

TABLE III. RECOMMENDED MINIMUM EVALUATION PROTOCOL FOR ATTRITION STUDIES

Element	Recommendation
:---	:----------
Splitting	Stratified on the target; fixed random seed; test set untouched until final evaluation
Resampling	Only within training folds, never before the split
Baselines	Report the majority-class baseline and at least one simple model
Metrics	Precision, recall, F1, confusion matrix, ROC-AUC, PR-AUC with prevalence stated
Uncertainty	Cross-validation with confidence intervals; significance tests for model comparisons
Thresholds	Metrics at each operating threshold used by downstream actions
Calibration	Evaluate before describing scores as probabilities

The classification metrics for the positive (leaver) class are defined in Eqs. (3) to (5), where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives and false negatives.

$$\text{Precision} = TP / (TP + FP) \quad (3)$$

$$\text{Recall} = TP / (TP + FN) \quad (4)$$

$$F1 = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (5)$$

2) Workflow evaluation criteria

Prediction quality and workflow execution answer different questions and need different metrics. Suggested workflow metrics, each with an explicit denominator, are shown in Table IV.

TABLE IV. SUGGESTED WORKFLOW METRICS

Metric	Definition	Denominator
:--------	:----------	:----------
Event-creation success	Cases ending with a calendar event with expected attendees and time	All test cases
Meet-link success	Created events that contain a conference link	Created events
Scheduling latency	Time from workflow trigger to event creation	Successful cases
Duplicate-event rate	Events created more than once for the same recommendation	Created events
Failure and recovery	Handling of injected failures (unavailable slot, API error)	Injected failures
Integration errors	Count and type of error responses	All API calls

3) Responsible-use criteria

Employee-level risk scores are sensitive. A prediction is a risk indicator derived from historical patterns; it is not proof that an employee plans to resign, and approaching an employee because of a score raises ethical and privacy concerns. Design principles that follow from the sources and from the case in Sec. IV-E include:

- **Human review before action.** Each flagged recommendation should be confirmed or snoozed by an HR reviewer before any external action.
- **Role-based access.** Scores should be visible only to authorised HR users and, where appropriate, direct managers, and hidden from employees.
- **Data minimisation.** Webhook payloads, calendar titles and event descriptions should contain only what scheduling needs; neutral titles and no risk drivers in invitations.
- **Auditability.** Scans, inferences, triggers and notifications should be logged, with a defined retention limit.
- **Delegated access.** Calendar access should use standard authorisation such as OAuth 2.0.
- **Duplicate control.** A cooldown period prevents repeated contact for the same recommendation.
- **No unsupported compliance claims.** Compliance with a regulation or security standard should be claimed only after it has been assessed.

IV. RESULTS

This section synthesises the findings of the sources against the framework in Sec. III-B and answers RQ1 to RQ4.

A. Prediction Methods (RQ1)

The studies disagree on which model family is best: an ensemble tree method in [1], a Transformer in [2] and a feedforward network in [3]. The most defensible reading is that the best model depends on the protocol, namely the split, resampling choices, tuning procedure and metrics, and not that one family is superior in general. Test sets can also contain few positive cases (for example 36 leavers in [2]), so differences between models may not be stable.

B. Evaluation Practice (RQ2)

Pitfalls identified in the sources examined are resampling before splitting [1], accuracy-centred reporting [1], small test sets [2], and non-comparable protocols across studies [1]-[3]. Calibration and threshold analysis are further elements that a rigorous protocol should include, as set out in Table III.

C. Workflow Automation (RQ3)

In the sources examined, peer-reviewed attrition studies report prediction performance, while practitioner n8n material reports alerts or reports. We found no source that uses defined metrics to evaluate the path from a classifier output, through human review, to a calendar event. This is a narrower gap than attrition prediction or HR automation in general, both of which are active areas.

D. Responsible Use (RQ4)

Against the criteria in Sec. III-B3, the answer to RQ4 is that safeguards such as human review, role-based access, data minimisation, auditability and duplicate control are design requirements, and that in the sources examined they are described much more often than they are tested.

The case in Sec. IV-E illustrates both the safeguards and the gaps. Its webhook payload contained the employee identifier, name, e-mail address, department, manager e-mail address, the attrition probability, the risk level, a list of top reasons, a suggested action and a timestamp, and the event description carried the top reasons, so invitees could see the risk drivers [14]. The event title was neutral and showed no risk score. The authors of that case state that both the payload and the description should be minimised, that no retention limit for logs had been defined, and that no compliance with any regulation or security standard was claimed because none had been assessed [14].

E. Illustrative Case: A Prediction-to-Calendar Prototype

To make the discussion concrete, we summarise a recent prototype by the authors of this review [14]. It is presented as an illustration of the design space and not as external validation.

1) Prediction component

A gradient boosting classifier (scikit-learn 1.6.1; 200 trees, learning rate 0.05, depth 3, selected by grid search with 5-fold cross-validation) was trained on the IBM dataset using a stratified 80/20 split (1,176 training and 294 test records; 47 positive test records), a single preprocessing pipeline, a fixed seed of 42 and no resampling. Against logistic regression and a random forest under the same split, gradient boosting obtained an accuracy of 0.898, precision of 0.743, recall of 0.553, F1-score of 0.634, ROC-AUC of 0.864 and PR-AUC of 0.682, compared with a majority-class accuracy of 0.840. At the 0.50 threshold, 26 of 47 leavers were identified and 21 were missed.

For reproducibility, the configuration was as follows. The target is the binary attribute Attrition, with 1,233 records (83.9%) labelled No and 237 (16.1%) labelled Yes (Fig. 2). Four columns were dropped before modelling (EmployeeCount, StandardHours, Over18 and EmployeeNumber), leaving 30 input features: 23 numerical attributes, which were median-imputed and standardised, and seven categorical attributes, which were mode-imputed and one-hot encoded. All transformations were combined in a single scikit-learn pipeline applied identically to training and test records. The split was stratified on the target (190 positives in training and 47 in test), the random seed was 42, and no resampling or class weights were used. The software stack was Python 3.10, pandas 2.2.3, NumPy 2.2.3, SciPy 1.15.3 and scikit-learn 1.6.1; unlisted gradient boosting settings kept the library defaults [14].

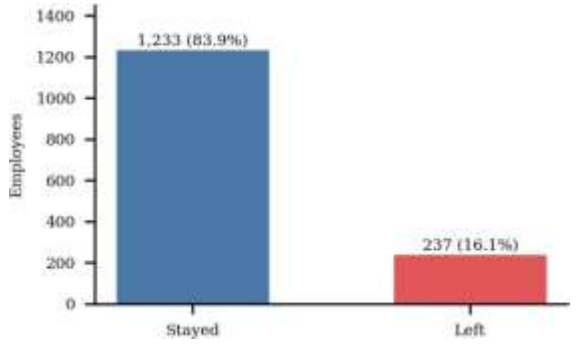


Fig. 2. Target-class distribution of the dataset [14]

Table V gives the test-set metrics at the 0.50 threshold. For gradient boosting the confusion matrix was TN = 238, FP = 9, FN = 21 and TP = 26.

TABLE V. HELD-OUT TEST-SET METRICS (N = 294, 47 POSITIVES) [14]

Model	Acc.	Prec.	Rec.	F1	ROC-AUC	PR-AUC
Gradient boosting	0.898	0.743	0.553	0.634	0.864	0.682
Logistic regression	0.871	0.636	0.447	0.525	0.812	0.584
Random forest	0.881	0.688	0.468	0.557	0.829	0.621
Majority class	0.840	n/a	0.000	0.000	n/a	n/a

Gradient boosting scored highest on every reported metric, but the differences come from a single split with 47 positive records and no significance test or confidence interval was computed, so they do not establish that it is better in general. Its accuracy is 5.8 percentage points above the majority-class baseline, and 35 employees (11.9% of the test set) were flagged, of whom 26 (74.3%) were genuine leavers. Fig. 3 compares the three classifiers.

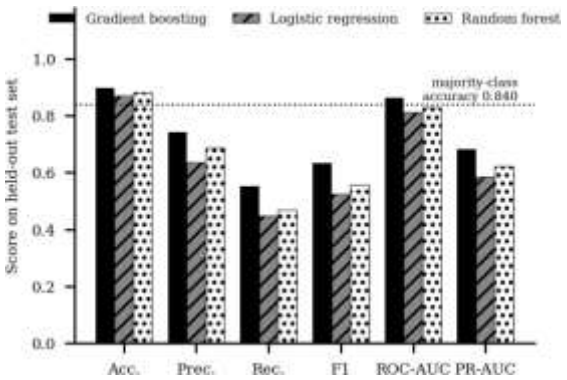


Fig. 3. Test-set metrics of the three classifiers [14]

2) Workflow component

A decision rule maps scores to four risk tiers (low below 0.40; medium from 0.40; high from 0.65; critical from 0.80). Scheduling starts only at the critical tier, subject to a seven-day cooldown, and after confirmation on an HR dashboard. An n8n workflow (webhook, two HTTP Request nodes, a Code node and a respond-to-webhook node) queries free/busy information and creates a 30-minute Google Calendar event with a Google Meet link. The cooldown, HR review and retry logic (three attempts with exponential backoff) run in the application layer, not as n8n nodes.

The workflow acts on a score and not only on a class label. For an employee i with score p_i and eligibility indicator e_i (for example, an active employee with no open follow-up), a recommendation a_i is raised according to Eq. (6), where the decision threshold is $\tau = 0.80$. For the classification metrics in Table V, a score of 0.50 or higher counts as a positive prediction. A recommendation is passed to an HR reviewer, who decides whether any contact is appropriate.

$$a_i = 1 \text{ if } p_i \geq \tau \text{ and } e_i = 1; \text{ otherwise } a_i = 0 \qquad (6)$$

In the implemented workflow, a request reaches n8n through a webhook node that accepts Hypertext Transfer Protocol (HTTP) POST requests. The

nodes, in order, are the webhook; an HTTP Request node that posts to the Google Calendar free/busy endpoint to obtain the availability of HR and the manager; a Code node (JavaScript) that picks the earliest free 30-minute window between 10:00 and 17:00 on the next business day and formats the times as Request for Comments (RFC) 3339 timestamps; a second HTTP Request node that posts to the events endpoint with the conference data version set to 1 and a Google Meet conference of type hangoutsMeet requested [11]; and a respond-to-webhook node. The canvas contains no condition node or error branch, because the tier, cooldown and review logic run in the application before the webhook is called. The request is sent by the application back end, which dispatches it asynchronously when an employee submits feedback, and by a Python alert agent that also performs a scheduled nightly scan. Each dispatch makes up to three attempts with exponential backoff.

The message sequence in the case has seven steps. The alert agent flags a recommendation to the HR dashboard; the reviewer confirms or snoozes it; for a confirmed recommendation the payload is posted to the n8n webhook; n8n queries free/busy information and creates the event; Google Calendar returns an event identifier or an error; the webhook returns a response to the caller; and the outcome is recorded in an audit log. Separating these steps is what allows each to be given its own denominator in Table IV.

Table VI lists the components and the status that the available evidence supports.

TABLE VI. IMPLEMENTATION COMPONENTS AND STATUS IN THE CASE [14]

Component	Where	Status
Data and preprocessing	Python pipeline	Implemented
Classifiers	scikit-learn; FastAPI service	Implemented
Decision rule, cooldown	Application layer	Implemented
Webhook interface	Back end; Python alert agent	Implemented
Workflow engine	n8n, five nodes	Implemented
Calendar, invite, Meet link	Google API via HTTP nodes	Implemented
HR review stage	Dashboard in application	Implemented; not an n8n node
Duplicates, retries, logging	Application layer	Implemented
Input schema validation	n/a	Proposed
Deployment environment	n8n instance	Not documented

Table VII lists the tiers and the eligibility condition: an employee is eligible only if no meeting was created during the previous seven days, unless the risk level has risen from high to critical.

TABLE VII. RISK TIERS AND ACTIONS IN THE CASE [14]

Tier	Score	Action
Low	below 0.40	Standard review cycle
Medium	0.40 to below 0.65	Department watchlist
High	0.65 to below 0.80	Flag on HR dashboard; notify manager
Critical	0.80 or higher	Start scheduling workflow after HR review

3) Test bench

In a 20-run test bench with synthetic records, 18 normal dispatches created events with Meet links, two edge cases (an invalid address and a busy slot) were handled, all five duplicate triggers were blocked by the cooldown, and mean latency was 1.42 s.

Table VIII reports the observed values against the metrics in Table IV.

TABLE VIII. OBSERVED WORKFLOW TEST-BENCH VALUES [14]

Metric	Observed value
Test cases run	20 (synthetic records)
Event-creation success	18 normal dispatches created an event; 2 edge cases handled
Meet-link success	Present in all created events
Scheduling latency	1.42 s (mean)
Duplicate-event rate	0 duplicates; 5 of 5 duplicate triggers blocked
Failure and recovery	3 of 3 busy slots moved to next free slot; invalid address handled

4) What the case does and does not show

It shows that the path from a classifier to a calendar event can be specified and measured with explicit denominators. It does not show reliability at scale, because the bench is small; it does not report performance at the 0.80 scheduling cutoff; it relies on one fictional dataset and one split; it has no fairness or explainability analysis; and it does not measure any effect on attrition. The calendar event description also included top risk reasons, which the prototype itself identifies as a privacy concern requiring minimisation.

F. Summary of Findings

Table IX summarises the answer to each research question together with the strength of the evidence behind it.

TABLE IX. SUMMARY OF FINDINGS BY RESEARCH QUESTION

RQ	Finding	Evidence
RQ1	The best model depends on the protocol; no ranking across studies is supported	Three studies, different protocols
RQ2	Recurring pitfalls: resampling order, accuracy focus, small test sets, non-comparable protocols	One study with leakage risk [1]; methodological references [7]
RQ3	No source evaluates the path from score through review to calendar event with defined metrics	Two practitioner sources [12], [13]; documentation [10], [11]; one case [14]
RQ4	Safeguards are specified as design requirements and are rarely tested	Methodological references [8], [9]; one case [14]

V. DISCUSSION

A. Interpretation

The sources reviewed do not support a ranking of model families. Gradient boosting served as a baseline in [2], an Extra Trees classifier led in [1], and a feedforward network led in [3], under different splits, resampling choices and metrics. The defensible conclusion is that the best model depends on the protocol, and that new studies should report a fully documented protocol (Table III) before comparing against published figures.

The case in Sec. IV-E shows how the prediction and scheduling components interact. Scheduling starts only at a score of 0.80 or higher, which is stricter than the 0.50 threshold used for the headline classification metrics. Fewer employees therefore reach the scheduling stage, and the recall at that operating point cannot exceed the recall of 0.553 reported at 0.50. The tier cutoffs thus directly determine HR workload and missed risk, and studies should report precision and recall at each cutoff. With no observed failures, the one-sided 95% lower confidence bound on a success rate is only about 0.85 for 18 dispatches and about 0.55 for five duplicate tests, so the test bench does not demonstrate reliability at scale. The prediction provides a ranking signal and the workflow provides the surrounding process, but neither validates the other.

Several assumptions remain. The dataset is fictional. The eligibility rule and threshold are design choices that influence how many employees are flagged. HR review is a procedural control whose effectiveness has not been tested. The present evidence supports claims about how a pipeline can be specified and measured on test data; it does not support claims about retention or employee satisfaction.

External validity is a further concern. The IBM dataset has a positive-class share of 16.1% [2], and the precision of a flagging rule at a given cutoff depends on how common leavers are in the population being scored [7]. A threshold tuned on this benchmark therefore cannot be transferred to another organisation without re-evaluation, and the cost of a false positive, which is unwarranted attention to an employee, has to be weighed against the cost of a false negative, an at-risk employee left unnoticed [9]. This asymmetry is one reason why a human reviewer decides whether any contact is appropriate, and why a score is treated as a risk indicator and not as evidence that an employee intends to resign.

B. Implications for Researchers and Practitioners

For researchers, the main implication is to treat the prediction and the action as two evaluated components, using leakage-free splits, imbalance-aware metrics and explicit workflow denominators. For practitioners building automations of the kind described in [12] and [13], the implication is to define what counts as a successful, duplicate or failed action before deployment, to keep a human reviewer between the score and the contact, and to minimise the data carried in payloads and invitations.

Governance matters as much as technique. The owner of the tier cutoffs and the cooldown should be named, the cutoffs should be reviewed against observed precision and recall at each tier, and the audit log should be sampled regularly to check that actions were taken only after review. These are organisational practices that the reviewed sources do not evaluate, and they are where a prototype is most likely to differ from a production system.

C. Research Gaps and Challenges

1. **Non-comparable protocols.** Splits, resampling, tuning and metrics differ across studies, so reported rankings of models cannot be compared directly [1]-[3].
2. **Evaluation rigour.** Test sets can contain few positive cases (36 leavers in [2]); cross-validated confidence intervals, calibration and threshold analysis should be reported.
3. **Reliance on one fictional dataset.** Generalisation to real organisations is unestablished.
4. **Missing evaluation of the action step.** Practitioner workflows [12], [13] and academic studies [1]-[3] do not report scheduling reliability, duplicate control or failure handling with defined metrics.
5. **Limited fairness and explainability evidence.** Bias and explanation quality should be evaluated alongside predictive performance [8],

[9]; we did not identify such evaluations in the sources examined.

6. **Privacy by design is described, not tested.** Payload minimisation, retention limits and access controls need verification.
7. **No outcome evidence.** No source we examined shows that predicting attrition and arranging meetings reduces attrition or improves employee wellbeing.

D. Future Research Directions

- Validation on appropriately authorised organisational data, with ethical approval and consent processes.
- Cross-validated comparison of classifiers under one documented protocol, with confidence intervals and significance tests.
- Threshold tuning and class-imbalance treatment inside training folds, with precision and recall reported at each risk-tier cutoff.
- Calibration analysis before scores are described as probabilities.
- Explainability (for example SHAP [8]) and group-wise fairness evaluation [9].
- Load testing and failure injection for workflows, including exhausted retries and API errors.
- Payload and invitation minimisation, and defined log retention.
- Controlled studies of whether human-reviewed workflows help HR staff, and whether interventions affect retention or employee satisfaction.

E. Recommended Reporting Checklist

Drawing the two evaluation tracks together, we suggest that studies linking an attrition classifier to an action report at least the following items.

1. The dataset, its provenance and whether it is real or synthetic, with class counts and prevalence.
2. The split, the random seed, and where resampling and tuning take place relative to the split.
3. The majority-class baseline and at least one simple comparison model.
4. Precision, recall, F1-score, the confusion matrix, ROC-AUC and PR-AUC, with confidence intervals where possible.
5. The decision threshold or tier cutoffs that trigger an action, with precision and recall at each cutoff.
6. The eligibility and cooldown rules, and who reviews a recommendation before any contact.
7. Workflow metrics with explicit denominators: success, link success, latency, duplicates and handled failures.
8. The data carried in payloads and invitations, access controls, and log retention.

VI. CONCLUSION

A. Limitations of the Review

This review has several limitations. The search was limited and non-systematic, and the number of sources is small, so the conclusions describe the sources examined and not the whole field. Practitioner sources were not peer reviewed and are used only as precedent. Figures from the cited studies are reported as stated and were not recomputed. The comparison focuses on the IBM dataset, and other data sources and settings are outside its scope. The illustrative case comes from the authors' own prototype and is not independent validation; it rests on one fictional dataset, a 20-run synthetic test bench, and no measurement of effects on attrition.

B. Concluding Remarks

The attrition literature we examined concentrates on prediction accuracy on a single fictional benchmark, with conclusions about the best model that depend heavily on protocol. Practitioner work has started to connect predictions to automated actions, but without reported evaluation. We argue that attrition systems should be assessed as two linked components, a classifier evaluated with leakage-free, imbalance-aware and uncertainty-aware metrics, and a workflow evaluated with explicit reliability metrics, wrapped in human review and privacy safeguards. Every score should be treated as a risk indicator and not as evidence that someone intends to resign. The review is limited by its non-systematic search and its small set of sources, and its conclusions should be read accordingly.

ACKNOWLEDGEMENT

Parts of the manuscript text were drafted with the assistance of an artificial intelligence (AI) language model (Claude, Anthropic) and were reviewed and edited by the authors, who take responsibility for the content.

REFERENCES

[1] A. Raza, K. Munir, M. Almutairi, F. Younas, and M. M. S. Fareed, "Predicting employee attrition using machine learning approaches," *Appl. Sci.*, vol. 12, no. 13, Art. no. 6424, 2022, doi: 10.3390/app12136424.

[2] W. Li, "A transformer-based deep learning framework to predict employee attrition," *PeerJ Comput. Sci.*, 2023, doi: 10.7717/peerj-cs.1570.

[3] M. Nandal, V. Grover, D. Sahu, and M. Dogra, "Employee attrition: Analysis of data-driven models," *EAI Endorsed*

Trans. Internet Things, vol. 10, no. 1, 2024, doi: 10.4108/eetiot.4762.

[4] "IBM HR analytics employee attrition & performance," Kaggle dataset (contributor: pavansubhasht). [Online]. Available: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset (accessed Oct. 11, 2026).

[5] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189-1232, 2001, doi: 10.1214/aos/1013203451.

[6] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825-2830, 2011.

[7] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765-4774.

[9] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1-35, 2021, doi: 10.1145/3457607.

[10] n8n, "Google Calendar node," n8n Documentation. [Online]. Available: https://docs.n8n.io/integrations/builtin/app-nodes/n8n-nodes-base.googlecalendar/ (accessed Oct. 11, 2026).

[11] Google, "Events: insert," Google Calendar API v3 Reference. [Online]. Available: https://developers.google.com/calendar/v3/reference/events/insert (accessed Oct. 11, 2026).

[12] "Employee attrition risk detection & HR alerts using Azure OpenAI GPT-4o-mini & Gmail," n8n workflow template 9309. [Online]. Available: https://n8n.io/workflows/9309-employee-attrition-risk-detection-and-hr-alerts-using-azure-openai-gpt-4o-mini-and-gmail/ (accessed Oct. 11, 2026).

[13] shwetamauryak, "HR-AI-Attrition-Workflow: HR employee attrition analysis using Python, Pandas, n8n automation, and AI-style reporting," GitHub repository. [Online]. Available: https://github.com/shwetamauryak/HR-AI-Attrition-Workflow (accessed Oct. 11, 2026).

[14] V. K. Deshmukh, A. M. Deshmukh, K. N. Raghuwanshi, A. S. Shrikhandkar, and K. H. Deshmukh, "Integrating gradient boosting and artificial intelligence agent-based workflow automation for employee attrition prediction and human resource meeting scheduling," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 10, no. 10, Oct. 2026.