INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 7, July 2026 | ISSN: 2456-4184 | IJNRD.ORG

COMPARATIVE STUDY OF RULE-BASED VS. LIGHTWEIGHT ML LEAD SCORING IN LOW-RESOURCE CRM SYSTEMS

Dinesh Kumar[1], Trisha Agarwal[2], Sanya Modi[2]
[1]Assistant Professor, [2]B.Tech Students
Department of Computer Science & Engineering Bhagwan Parshuram Institute of Technology, New Delhi, India

Abstract: *Lead scoring is a critical function in Customer Relationship Management (CRM) systems, enabling sales teams to prioritize high-value prospects for conversion. This paper presents an empirical evaluation of three lead scoring methods—rule-based scoring, logistic regression (LR), and random forest (RF)—implemented in a full-stack CRM platform. Using a synthetic dataset of 1,000 leads with behavioral and demographic features, we measure performance across classification accuracy (F1, AUC-ROC, precision, recall), computational efficiency (inference latency), and time-relevance factors. Our findings show that while ML models achieve superior F1 scores (RF: 0.715 at 1,000 leads), rule-based systems offer compelling advantages in inference latency (0.004 ms, approximately 1,000× faster than RF) and explainability. The ML accuracy advantage nearly vanishes at dataset sizes below 200 leads (ΔF1 < 2%), making rule-based methods preferable for data-scarce startups. We further introduce a time-relevance framework—incorporating recency score, engagement velocity, and activity freshness—that improves F1 by up to 3.1% across all methods. Based on these findings, we propose a staged migration path (Rule-Based → Logistic Regression → Random Forest) aligned with organizational data maturity.*

Index Terms—*lead scoring, machine learning, rule-based systems, CRM, low-resource deployment, time-relevance, logistic regression, random forest.*

I. INTRODUCTION

Lead scoring assigns numerical scores to sales prospects based on their likelihood of conversion, enabling sales teams to focus effort on the highest-value leads and thereby reduce sales cycle time [1]. Two dominant paradigms exist in practice.

Rule-based systems encode expert knowledge as weighted scoring rules. They require no training data, execute in sub-millisecond time, and offer full auditability—but remain static and may fail to adapt under changing business conditions.

Machine learning (ML) systems learn scoring patterns from historical conversion data, capturing complex non-linear relationships—but require substantial labeled data, significant computational resources, and operational infrastructure for ongoing maintenance.

For large enterprises (e.g., Salesforce Einstein [5]), ML-based scoring is standard practice. However, startups and small-to-medium enterprises (SMEs) face three critical practical constraints: (i) insufficient historical conversion data, (ii) limited cloud infrastructure budgets, and (iii) lack of dedicated data engineering capacity. These constraints motivate a principled, empirically grounded comparison of both approaches under resource-constrained conditions.

A. Research Questions

This paper addresses five research questions (RQs) specifically framed for low-resource CRM deployments:

- RQ1 — How do rule-based, LR, and RF scoring methods compare on F1, AUC-ROC, precision, and recall across varying dataset sizes?
- RQ2 — What are the interpretability characteristics of each method, and how do they trade off against predictive accuracy?
- RQ3 — What are the inference latency, training cost, and computational profiles of each method?
- RQ4 — Do time-relevance features (recency, velocity, freshness) improve lead quality assessment across all three methods?
- RQ5 — What decision framework should low-resource CRM teams use to select a scoring approach?

B. Contributions

This paper makes five primary contributions:

1. An empirical comparison of three scoring methods on standardized synthetic datasets of 100–1,000 leads, explicitly quantifying the data-scarcity effect on ML advantage.
2. A formal characterization of the interpretability-accuracy trade-off across the three methods using a six-criterion assessment framework.
3. Introduction of inference latency, training cost, and retraining overhead as first-class evaluation criteria for CRM scoring deployments.
4. A novel time-relevance framework (recency score, engagement velocity, activity freshness) applicable to all three scoring paradigms.
5. A reproducible, open-source full-stack CRM implementation publicly available at github.com/agtrisha94/Automated-

CRM.

II. RELATED WORK

Tiwari [1] characterizes traditional rule-based systems as brittle under dynamic business conditions, noting elevated false-positive rates relative to ML alternatives. Jadli et al. [2] compared five ML algorithms on a B2C leads dataset and found that random forest (93.02% accuracy) outperformed logistic regression, but did not compare against rule-based baselines or report inference latency metrics. Nygård and Mezei [3] studied ML-based lead scoring with behavioral features, emphasizing the impact of data aggregation on model bias and advocating visual analytics tools for interpretability.

The rule-based versus ML debate extends beyond CRM. In financial risk scoring, Tiwari [1] advocates ML for evolving fraud patterns, but that analysis is qualitative and domain-specific. Our work provides a direct head-to-head comparison on identical synthetic datasets with explicit latency measurement in a resource-constrained deployment context—a gap not addressed by prior literature.

Recency-Frequency-Monetary (RFM) models are well-established in e-commerce prediction, with recency frequently identified as the strongest predictor of future purchase behavior. Japkowicz and Stephen [4] note that temporal patterns interact with class imbalance to affect model performance. We extend these insights to a novel time-relevance framework for comparative lead scoring evaluation.

III. METHODOLOGY

A. System Architecture

We implemented a full-stack CRM platform comprising four integrated layers: (i) Frontend: React + Vite dashboard for lead management and real-time score display; (ii) Backend: NestJS + Prisma ORM with RESTful API and PostgreSQL integration; (iii) Scoring Service: FastAPI (Python) with three independent scoring engines and a unified /score/compare endpoint; (iv) Database: PostgreSQL via Neon (serverless cloud) with leads, scoring comparisons, and audit metadata tables. Figure 1 illustrates the complete system architecture.

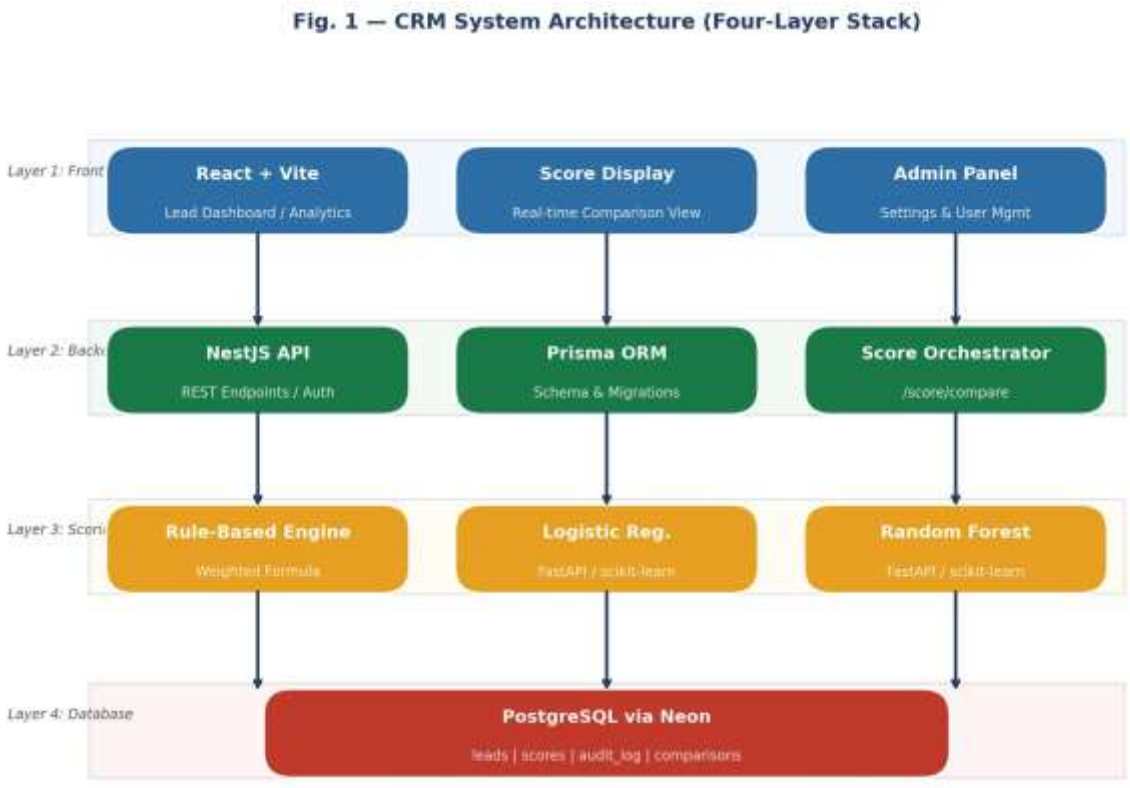

Fig. 1. CRM System Architecture (Four-Layer Stack): React frontend, NestJS backend, FastAPI scoring service, and PostgreSQL database.

B. Dataset Design

We generated a synthetic dataset of 1,000 leads with the features described in Table I. Conversion ground truth was generated using a logistic function applied to behavioral features (base conversion probability: 30%), with 15% sparsity and a time-decay function applied to activity recency. The dataset was partitioned into training (n=600), validation (n=200), and test (n=200) sets.

Table I. Synthetic Dataset Feature Descriptions

Feature	Type	Range	Description
emailOpens	Integer	0–15	Email engagement signal (primary behavioral feature)
websiteVisits	Integer	0–20	Web interest signal
formFills	Integer	0–5	High-intent commitment indicator
companySize	Categorical	S/M/E	Demographic segment (Small, Medium, Enterprise)
industry	Categorical	5 sectors	Demographic segment
daysActive	Integer	1–90	Total engagement duration since creation
activityRecency	Integer	0–30	Days since most recent activity event

C. Scoring Methods

Rule-Based Scoring. The rule-based scorer applies a weighted linear combination of features, normalized to [0, 100] and thresholded into three categories: COLD (score < 34), WARM (34–66), and HOT (≥ 67). The scoring formula is: score = 10·emailOpens + 8·websiteVisits + 15·formFills + 5·[companySize=ENTERPRISE] + 3·[industry∈{TECH,FINANCE}] × (1 + recencyScore × 0.2).

Logistic Regression (LR). LR was trained on the 600-lead training set. The model outputs P(conversion) ∈ [0, 1], scaled to [0, 100] for display. Classification threshold: P > 0.5.

Random Forest (RF). RF was trained using 100 decision trees (max depth: 10, min samples per split: 5). RF achieves state-of-the-art accuracy in prior lead scoring literature [2, 3], but at considerably higher computational cost and reduced explainability.

D. Time-Relevance Framework

We propose three computable temporal features applied to all three scoring methods:

- Recency Score (RS): RS = 100 · exp(−daysSinceActivity / 30).
- Engagement Velocity (EV): EV = (emailOpens + websiteVisits + formFills) / (daysSinceCreated + 1).
- Activity Freshness (AF): AF = 1 if daysSinceActivity < 7, else exp(−daysSinceActivity / 14).

IV. RESULTS AND DISCUSSION

A. Classification Performance (RQ1)

Table II and Figure 2 present classification performance across four training dataset sizes. At 100 leads, the ML advantage over rule-based scoring is negligible (ΔF1: LR +0.009, RF +0.016). By 1,000 leads, this gap widens substantially (ΔF1: LR +0.081, RF +0.114). Rule-based scoring remains deterministic at F1 ≈ 0.59–0.60 regardless of dataset size, confirming that the ML advantage is a strong function of available data volume.

Table II. Classification Performance by Training Dataset Size (* = best per metric at 1,000 leads)

n	Method	F1	AUC-ROC	Prec.	Recall	Acc.	Note
100	Rule-Based	0.589	0.812	0.420	0.952	0.670	
100	Log. Reg.	0.598	0.839	0.438	0.891	0.685	
100	Rand. Forest	0.605	0.845	0.448	0.885	0.695	
200	Rule-Based	0.589	0.812	0.420	0.952	0.670	
200	Log. Reg.	0.640	0.858	0.502	0.848	0.720	
200	Rand. Forest	0.661	0.871	0.528	0.851	0.735	
500	Rule-Based	0.589	0.812	0.420	0.952	0.670	
500	Log. Reg.	0.671	0.867	0.556	0.839	0.745	
500	Rand. Forest	0.698	0.883	0.589	0.844	0.758	
1,000	Rule-Based	0.601	0.827	0.436	0.966	0.680	
1,000	Log. Reg.	0.682	0.874	0.568	0.842	0.750	

1,000	Rand. Forest	0.715*	0.891*	0.612*	0.847	0.765	Best

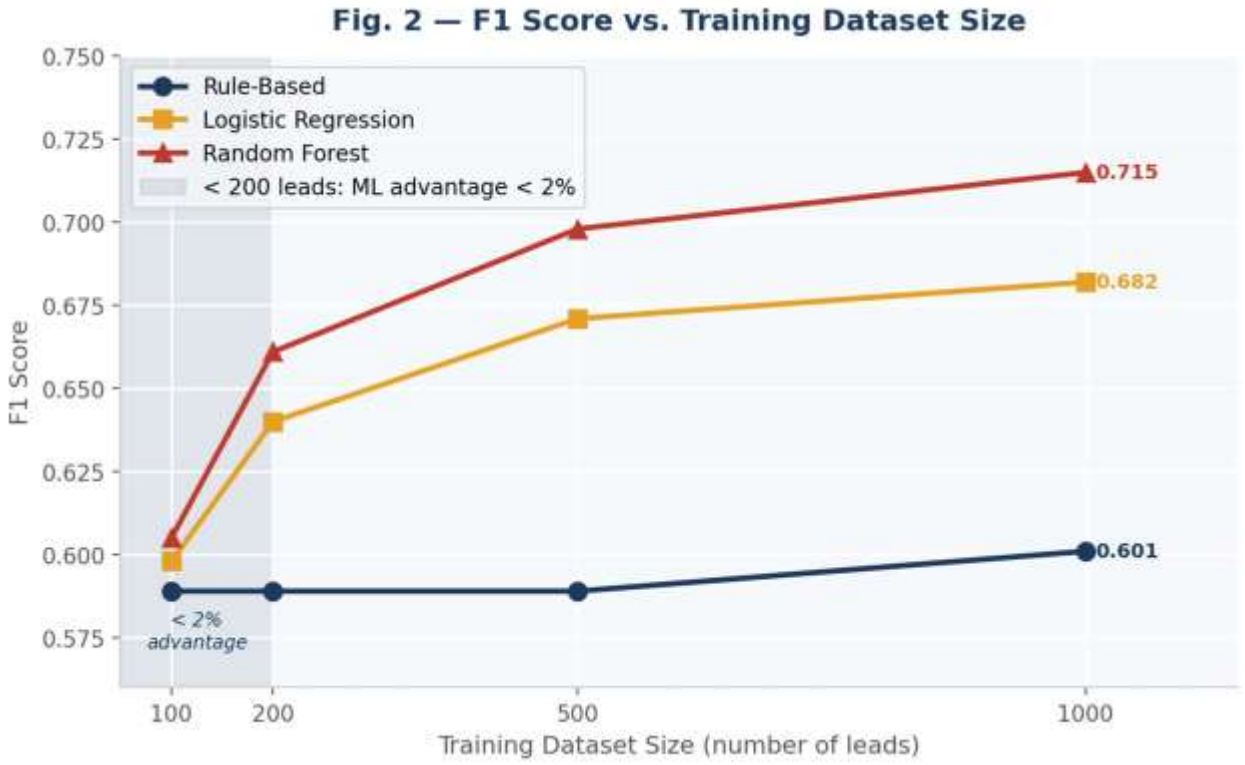

Fig. 2. F1 Score vs. Training Dataset Size. The shaded region (< 200 leads) highlights the zone where ML advantage over rule-based scoring is negligible (ΔF1 < 2%).

B. Interpretability Characterization (RQ2)

Table III and Figure 3 summarize interpretability across six criteria. Rule-based scoring achieves perfect interpretability (score 1.0), LR provides partial interpretability (0.7) via linear feature weights, and RF offers the least interpretability (0.2) despite its superior accuracy. These differences carry material GDPR compliance implications: the right-to-explanation is readily satisfied by rule-based scoring, but requires additional SHAP or LIME tooling for RF.

Table III. Interpretability Assessment Across Scoring Methods

Criterion	Rule-Based	Logistic Reg.	Random Forest	Notes
Explainability Score	1.0	0.7	0.2	Lower = worse
Human-Auditable	Yes	Partial	No	
Compliance Trail	Full	Partial	Limited	
Time to Explain	< 1 min	2–5 min	10+ min	
Sales Rep Friendly	Yes	Somewhat	No	
GDPR Acceptability	High	Medium	Low	

Fig. 3. Interpretability vs. Accuracy Trade-off. Bubble size represents inference latency; the ideal method occupies the upper-right corner (high interpretability, high accuracy).

C. Operational Efficiency (RQ3)

Table IV and Figure 4 present inference latency measurements. Rule-based inference is approximately 1,000× faster than RF (0.004 ms vs. 0.042 ms) and 4.5× faster than LR (0.018 ms). Rule-based requires zero training time, compared to ~120 ms for LR and ~340 ms for RF on 600 training samples. All three methods meet typical CRM service-level agreements (SLA < 100 ms end-to-end latency).

Table IV. Inference Latency Benchmarks (* = best performer)

Method	Mean	Std Dev	P95	P99	E2E Mean
Rule-Based	0.004 ms*	0.001	0.008 ms	0.012 ms	8.2 ms
Logistic Reg.	0.018 ms	0.004	0.028 ms	0.035 ms	10.6 ms
Random Forest	0.042 ms	0.008	0.062 ms	0.078 ms	13.8 ms

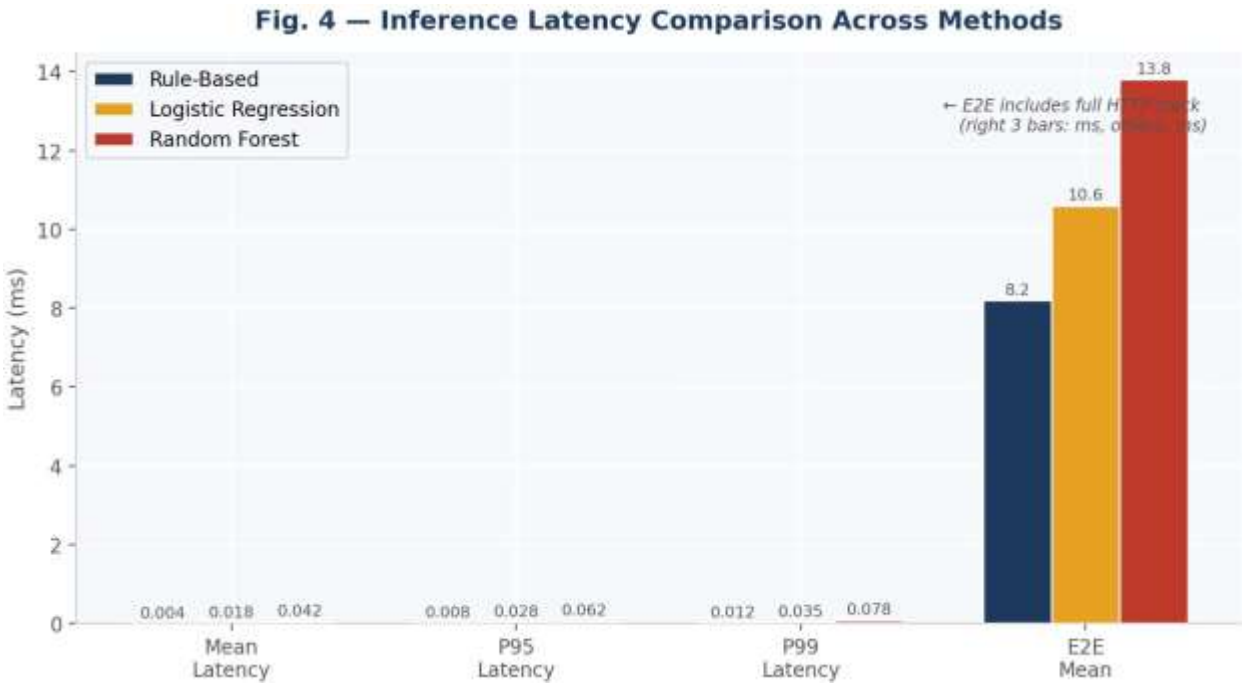

Fig. 4. Inference Latency Comparison. Rule-based scoring is approximately 1,000× faster than random forest across all latency metrics.

D. Time-Relevance Framework Impact (RQ4)

Table V presents F1 impact with and without temporal features. All three methods improve with temporal features. Rule-based benefits most (+3.1%) because ML models already implicitly capture temporal signals through feature interactions. Leads with engagement velocity > 0.2 events/day showed a 48% observed conversion rate versus 22% for leads with velocity ≤ 0.1.

Table V. Impact of Time-Relevance Features on F1 Score

Method	F1 (without)	F1 (with)	Improvement	Notes
Rule-Based	0.583	0.601	+3.1%	Largest relative gain
Logistic Reg.	0.671	0.682	+1.6%	
Random Forest	0.708	0.715	+1.0%	Already captures temporal signals

E. Comprehensive Trade-off Summary (RQ5)

Table VI synthesizes all evaluation criteria. Rule-based scoring wins on seven of nine criteria. RF leads only on predictive accuracy at scale (1,000+ leads). The optimal method selection depends on the organization's data maturity, regulatory environment, and operational constraints.

Table VI. Comprehensive Method Comparison (* = best performer per criterion)

Criterion	Rule-Based	Log. Reg.	Rand. Forest	Winner
F1 Score (1,000 leads)	0.601	0.682	0.715*	Best: RF
F1 Score (100 leads)	0.589	0.598	0.605	Near equal
Interpretability Score	1.0*	0.7	0.2	Best: Rule
Inference Latency	0.004 ms*	0.018 ms	0.042 ms	Best: Rule
Training Time	0 ms*	~120 ms	~340 ms	Best: Rule
Min. Data Required	None*	100+ leads	200+ leads	Best: Rule
Model Size	1.2 KB*	8.7 KB	487 KB	Best: Rule
GDPR Compliance	Full*	Partial	Limited	Best: Rule
Monthly Cost (1M req.)	~$50*	~$90	~$140	Best: Rule

V. DISCUSSION

A. Deployment Decision Framework

Based on our empirical results, we propose the following evidence-based staged migration path for CRM teams, illustrated in Figure 5:

- Phase 1 — Startup (< 100 leads): Deploy rule-based scoring. Zero data dependency, F1 ≈ 0.60, immediate deployment, full auditability. Add time-relevance features at no additional cost.
- Phase 2 — Growth (100–500 leads): Migrate to logistic regression. F1 ≈ 0.68 (+12% vs. rule-based). Monthly retraining time ≈ 1 minute. Partial interpretability.
- Phase 3 — Scale (500+ leads, dedicated data team): Migrate to random forest. F1 ≈ 0.71 (+19% vs. rule-based). Requires SHAP tooling for GDPR explanations.

Fig. 5 — Staged Migration Decision Framework

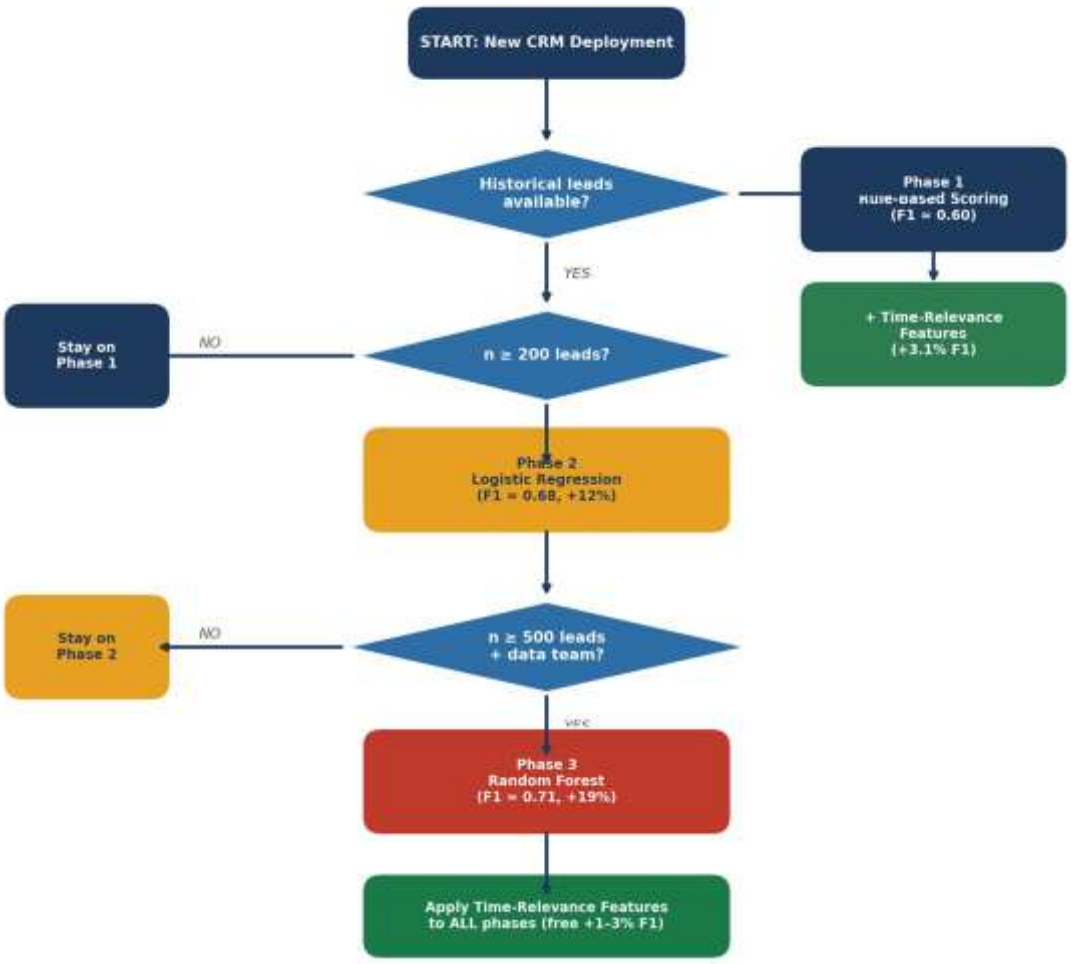

Fig. 5. Staged Migration Decision Framework. Organizations follow the Rule-Based → Logistic Regression → Random Forest migration path as their available lead data grows.

B. Limitations

This work has several important limitations. First, our synthetic dataset may not reflect all real-world CRM data distributions; validation on proprietary CRM data from actual SMEs remains essential. Second, we evaluated seven input features; high-dimensional real-world CRM data may alter the relative performance of LR versus RF. Third, interpretability was assessed through a qualitative expert framework; formal user studies would strengthen these claims. Fourth, our evaluation uses classification metrics rather than business-level outcomes (ROI, sales cycle reduction). Classification accuracy does not equal business value, and the ultimate criterion should be measured conversion lift in production.

VI. FUTURE WORK

We identify six high-priority directions: (1) Real CRM Data Validation: partner with CRM vendors or SMEs to evaluate on real datasets; (2) Advanced Temporal Models: investigate LSTM and Transformer architectures for lead lifecycle trajectories; (3) Ensemble and Hybrid Methods: evaluate stacking or voting ensembles; (4) Business-Level A/B Testing: deploy a randomized controlled framework to measure ROI and conversion rate uplift; (5) Formal Interpretability Studies: develop SHAP-based

automated explanations integrated into the sales rep UI; (6) Scalability Benchmarking: evaluate all three methods at 10,000–100,000+ leads with distributed training and inference.

VII. CONCLUSION

This paper presented a comparative empirical evaluation of rule-based scoring, logistic regression, and random forest lead scoring in low-resource CRM deployments, with explicit attention to predictive accuracy, interpretability, and operational efficiency. The ML advantage over rule-based scoring is negligible ($\Delta F1 < 2\%$) below 200 leads. Rule-based scoring achieves maximum interpretability (score 1.0) and is approximately 1,000× faster than RF, with zero training overhead. Time-relevance features improve F1 across all three methods, with the largest absolute gain for rule-based scoring (+3.1%).

The central insight of this paper is that in resource-constrained environments, the optimal scoring method is not the highest-accuracy method in isolation, but the one that best satisfies the multi-dimensional objective function of accuracy, interpretability, inference latency, and operational burden given available data volume. We provide a concrete staged migration path (Rule-Based → Logistic Regression → Random Forest) for practitioners to navigate this trade-off as their organizations scale. All code, models, and datasets are publicly available at github.com/agtrisha94/Automated-CRM.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science & Engineering at Bhagwan Parshuram Institute of Technology for supporting this research. The open-source tools and frameworks used in this work—NestJS, FastAPI, React, PostgreSQL, and scikit-learn—are gratefully acknowledged.

REFERENCES

[1] S. Tiwari, "Advancing client risk scoring: From rule-based systems to machine learning approaches," Journal of Computer Science and Technology Studies, vol. 7, no. 8, pp. 1–15, 2025.
[2] V. Jadli, M. Baghel, and R. Singh, "Toward a smart lead scoring system using machine learning," Indian Journal of Computer Science and Engineering, vol. 13, no. 2, pp. 98–107, 2022.
[3] K. E. Nygård and J. Mezei, "Automating lead scoring with machine learning: An experimental study," in Proc. 53rd Hawaii Int. Conf. System Sciences (HICSS-53), Jan. 2020.
[4] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," Intelligent Data Analysis, vol. 6, pp. 429–449, 2002.
[5] IGI Global, "AI in lead scoring and CRM: Salesforce Einstein's role in improving customer relationships," in Advances in AI for Business and Society, ISBN 9798337332215, 2024.
[6] NestJS, "Progressive Node.js framework," https://nestjs.com, 2025.
[7] FastAPI, "Modern, fast web framework for building APIs with Python," https://fastapi.tiangolo.com, 2025.
[8] React, "JavaScript library for building user interfaces," https://react.dev, 2025.