International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Advibe: An AI-Assisted Investor-Discovery and Outreach Prototype for Founders

Tejaswini K[1], Sameer Mokhasi[2], Laksh G[3] Rajat Y[4]
[1,2,3,4] Department Of Computer Science And Business Systems, B.M.S. College Of Engineering, Bengaluru, Karnataka, India
{Tejaswinik.Cbs, Sameervenkatesh.Bs24, Lakshgajre.Bs23, Rajaty.Bs23}@Bmsce.Ac.In

Abstract. For founders to fundraise, they need to find investors that fit the same stage, sector, and geography as their company, contact them via an email, and track responses in multiple disconnected tools. Advibe is a prototype that unifies all these tasks in a unified workflow. The system takes a pitch deck or company description, creates a company profile, finds suitable investors in a catalog, scores them using an explainable score, drafts emails to be sent to the investors, and classifies the responses received. The Advibe system has a React and Vite client with a FastAPI backend, using PostgreSQL database on Supabase and email delivery through Resend. All language models are used to create company profiles, score investors and give an explanation, draft emails, and classify responses. The main model used is Llama 3.3 70B versatile which is served by Groq, and the fallback model is DeepSeek-R1 served through OpenRouter while the fallback for keyword-based use cases for company profile and response classification is also used. Ranking involves two stages: the rule score R, based on stage, sector, and geography matches, plus recency, ranks up to 50 candidates, who in turn get ranked by language model score L, with final score S = 0.40R + 0.60L. In our analysis of this score in closed form we enumerate all seven possible rule scores and compute break-even language model score for each of them. The investor catalogue contains 969 investor entries and 1,242 contact entries, of which the first sheet of the workbook supplied for analysis lists 114 investors, 84.2% of them residing in India. Intake, matching, drafting, approval gate for sending, and replies handling activities are related through 17 backend routes modules and 24 client components. For 20 test profiles, LLM-judged rating puts the rule stage significantly above random ranking (precision@10 = 0.91 vs 0.42). Human judgements and evaluation of LLM stage require future work.

Keywords: investor discovery · LLM ranking · human-in-the-loop outreach · startup fundraising · retrieve-then-rank

I. INTRODUCTION

When a founder already has a product and is looking to raise money within just a few months' runway, fundraising starts as a discovery challenge. There are many many funds out there, but not even a fraction of them invest in founders' stages, founders' sectors and founders' regions. Finding them requires studying their websites, compiling lists, contacting people and writing emails one by one. Huge investor databases help with the first part, but most of them are created for the American market and ignore others [1]. A founder in India, for instance, should have a list of investors who actively write checks in the region, who rank themselves according to some logic, and whom one has total control over all outgoing emails.

This is the gap Advibe is built for. Advibe takes a pitch deck or a short company description, works out the stage, sector, geography, cheque size and thesis, and returns a ranked list of investors that fit, each with a short reason. It then drafts a personalised email for each investor, holds it for the founder's approval, sends it only after approval, and sorts the replies into categories so the founder can see where the round stands. The fit score is deliberately simple, so a founder can see why an investor ranked where it did, and the rationale is stored with the score.

The main contributions of this paper are:

1. the design and implementation of an end-to-end pipeline that connects pitch-deck intake, investor retrieval, two-stage ranking, outreach drafting, approval-gated sending and reply labelling (Sections 3 and 6);

2. a two-stage score that combines a transparent rule score with a language-model score, together with a closed-form analysis of every reachable rule score and its break-even model score (Section 5);

3. a catalogue of 969 investors and 1,242 contacts across venture, real-estate and limited-partner (LP) tracks, with a profile of its coverage (Section 7);

© 2026 IJSRET

4. a working web interface for intake, agent chat and a command centre (Section 8); and

5. an offline evaluation of the rule stage on 20 test profiles and a protocol for human ratings of the full ranking (Sections 9 and 10).

Section 2 reviews existing applications and related work. Section 3 describes the system architecture and Section 4 the methodology. Section 5 explains how a match is scored, Section 6 how the system works end to end, Section 7 the dataset and Section 8 the interface. Section 9 discusses the results, Section 10 concludes, and Section 11 states the limitations and future work.

1. Background And Related Work

a. Existing Applications

Founders already have tools for parts of this problem. Table 1 lists four existing tools [2]–[5]. Their descriptions follow each tool's own pages and listings as they appeared in October 2026, and we did not test the tools ourselves. Taken together, they centre on investor databases, filters and pipeline tracking; the right-hand column of Table 1 states what Advibe intends to add.

Table 1. Existing applications related to Advibe, as described on their own pages. The right-hand column records design intent and is not a tested comparison.

Application	What it offers	Advibe intended difference
Foundersuite [2]	Investor database of 230,000+ investors with a fundraising CRM, pitch-deck sharing and email follow-ups	Profile built from the deck, a stored fit rationale and an approval step before sending
OpenVC [3]	Free database of 20,000+ investors filterable by stage, sector and geography; founders can submit a pitch deck to investors	A ranked shortlist and drafted outreach
Signal by NFX [4]	Free investor search by sector and stage, fundraising tracking, and Gmail-based discovery of existing investor contacts	Drafted outreach and reply labelling
Tracxn [5]	Market-research platform for tracking startups and deals, aimed mainly at VC, PE and corporate teams; Bengaluru-based	The founder side of the process rather than investor research

However, from the texts that we have studied, there is no single software mentioned that has the capacity to build the profile from a pitch deck, stores the reasoning behind every investor being ranked in that particular way and requires the founder's approval for sending out the drafts. This is due to the restrictions in what we have read and not an indication of the market. The design intention of Table 1 is, therefore, documented below.

This involves retrieval followed by ranking of the candidates. A cheap rule-based scoring initially filters out the candidates, and then a language model scores the few filtered candidates. The second step has something in common with the research work of Sun et al. [6] on the application of LLMs as re-ranking systems for passage search.. Their results concern passages, so for Advibe they motivate the design but do not support its quality. Model output is parsed as JSON. The engineering rules require JSON with one retry, and Groq documents that its JSON mode guarantees valid JSON but not a particular structure [7]. A profile or score that parses is therefore not necessarily a correct one, which matters for the default values discussed in Section 6.1. LLM judges also show position, verbosity and self-enhancement biases, though strong models agree with human preferences over 80% of the time [8]; this argues for calibrating L against human ratings before it carries 60% of the final score.

b. Requirements Traceability

Gotel and Finkelstein [9] analysed why following a requirement from its source into code is hard in practice, and Mäder and Egyed [10] measured the benefit that requirement-to-code traces bring to developers who maintain a system. Our methodology applies the same idea at small scale: each stated behaviour of Advibe is traced to the code that implements it (Section 4.2).

2. Outreach Safety And Email Rules

A tool that sends email on a founder's behalf carries two obligations that a ranking prototype does not. First, inbound events must be authenticated: the project documents call for a keyed-hash (HMAC) signature on the reply webhook [11], and Resend documents the specific headers its signatures use [12]. Second, commercial email must meet sender rules: the FTC's CAN-SPAM guidance asks for accurate sender information, a postal address and an opt-out [13]. We do not give legal advice.

3. System Architecture

1. Overview

Figure 1 shows the system layout and Table 2 lists the technology stack. A React and Vite client communicates with a FastAPI backend, which calls three service classes (AIService, DataService and OutreachService) and three external services: PostgreSQL on Supabase for storage,

© 2026 IJSRET

Groq for language models with OpenRouter as the fallback, and Resend for email, which also posts replies back to the webhook. The client comprises 24 component files and about 11,700 lines of JavaScript, and the backend comprises about 4,900 lines of Python, excluding tests.

Table 2. Technology stack of Advibe

Layer	Technology	Role in Advibe
Client	React 18 + Vite	Agent Chat and Command Center screens
Backend	FastAPI (Python)	17 route modules and 42 handlers: intake, match, outreach, webhook
Services	AIService, DataService, OutreachService	Extraction, scoring, drafts and reply labels; rule scoring and investor records; sending and credits
Database	PostgreSQL on Supabase	Profiles, investor catalogue, scores, campaigns, messages, outcomes; row-level security
LLM	Llama 3.3 70B Versatile on Groq [7], [14]; DeepSeek-R1 on OpenRouter as fallback [15]	Profile extraction, LLM score and rationale, drafts, reply labels; JSON output with one retry
PDF intake	PyMuPDF	Text extraction from an uploaded pitch deck
Email	Resend [12]	Sends approved drafts; posts reply events to the webhook
Tests	16 test functions	Mocked model calls, logging, one end-to-end flow

.

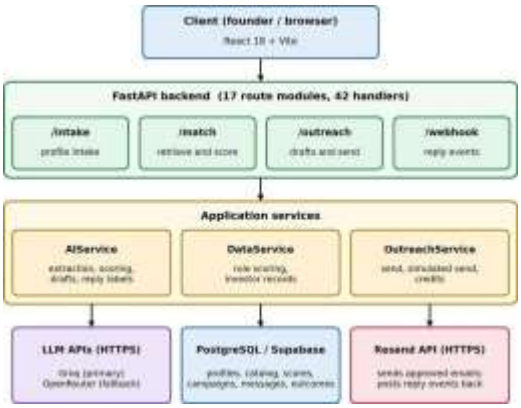


Fig. 1. System architecture of Advibe. The React client calls a FastAPI backend (17 route modules, 42 handlers), which uses three application services and three external services: LLM APIs, PostgreSQL on Supabase and Resend.

b. Main Workflow

Step-by-step main flow is depicted in Figure 2. The founder inserts text or uploads a PDF deck, and an LLM completes a profile in stage, sector, geography, cheque size, and thesis. The default value is used for any field not filled by the model: Seed stage, B2B SaaS sector, Global geography, and a cheque from $0.5M to $2M. The match flow then finds up to 50 investors that match the profile in at least one of stage, sector, or geography and evaluates them using the rule-based score R (Section 5). The language model evaluates the top 10 among them. One draft is generated for each contact, but only approved drafts are sent using Resend. Responses come into the webhook and get one of the five labels: four reply categories as per product requirements plus general reply..

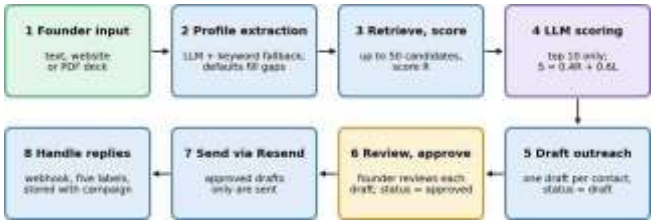


Figure 2. Primary process of Advibe comprising eight major steps from founder feedback till reply management.

II. METHODOLOGY

Resources

1. System Description

Description of the system is only possible through the project archives. This includes the README, product requirements document (PRD), Engineering rules, and the DATA_SOURCES Provenance Note; Backend, Frontend, Migrations, and Tests (17 Route modules, 24 Client component files, and 16 Test functions); Investor and Person CSV files (969 Investor rows and 1,242 Person Rows, Section 7); and Investor list.xlsx, a four sheet workbook created using an ETL script. No external dataset and model benchmark was necessary in describing the system. Rule-Stage Evaluation involves 20 Test Profiles and an LLM Judge (Section 9.1).

© 2026 IJSRET

2. Procedure

This process comprises four steps. **Extract:** list each behaviour stated in the README, PRD, rules or provenance note. **Trace:** locate the code in the backend, migrations or client that implements each behaviour. **Describe:** record how the behaviour works in the current code and which code paths use it. **Recompute:** repeat the scoring arithmetic and the data counts without relying on the code's own output. Behavioural descriptions therefore rest on reading the code and on one recorded session of the running web client (Section 8).

III. HOW A MATCH IS SCORED

1. The Two-Stage Score

The match route scores retrieved investors in two stages, beginning with a rule score *R*. A candidate receives 40 points if the investor's stage list contains the company's stage, 35 points for a sector match, 15 points for a geography match, and a flat 10-point recency term that a comment in the source code calls a "recency mock":

$$R = 40 \cdot I_{stage} + 35 \cdot I_{sector} + 15 \cdot I_{geo} + 10, \quad (1)$$

where each *I* is 1 for a match and 0 otherwise. The recency term is applied to every candidate, and cheque size is not part of this route's score. The route sorts by *R* (ties stay in database order), keeps 50 candidates and sends the top 10 to the LLM, which returns a score *L* between 0 and 100 and a short rationale. The two scores are blended as

$$S = 0.40 \cdot R + 0.60 \cdot L, \quad (2)$$

and the remaining 40 candidates, which receive no *L*, keep *S* = *R*. DataService contains a second rule-scoring function that uses the PRD weights (30, 35, 20 and 15 for stage, sector, geography and cheque size), but the match route does not call it. If the first attempt to invoke an LLM returns an error, then AIService will use a default value of L = 88.

2. A Worked Example

Investor A matches the company's stage and sector but not its geography, so $R = 40 + 35 + 0 + 10 = 85$. If the LLM gives A a 70, then $S = 0.40 \times 85 + 0.60 \times 70 = 76$. Investor B has the same matches but ranks 11th, so it is not sent to the LLM and keeps $S = R = 85$.

3. Enumeration And Break-Even Points

Because *R* takes only eight values, we enumerated every case (Table 3). The value 10 cannot occur among retrieved candidates, which leaves seven reachable values; Figure 3 plots three of them.

Table 3. Every reachable rule score *R* (Equation 1), its break-even LLM score *L*, and the final score *S* that a failed LLM call receives ($S = 0.4R + 52.8$).

R	Matches (plus the flat 10)	Break-even L	Failed-call S	Above rule-only candidate with same R?
25	Geography	25	62.8	Yes
45	Sector	45	70.8	Yes
50	Stage	50	72.8	Yes
60	Sector, geography	60	76.8	Yes
65	Stage, geography	65	78.8	Yes
85	Stage, sector	85	86.8	Yes
100	Stage, sector, geography	100	92.8	No

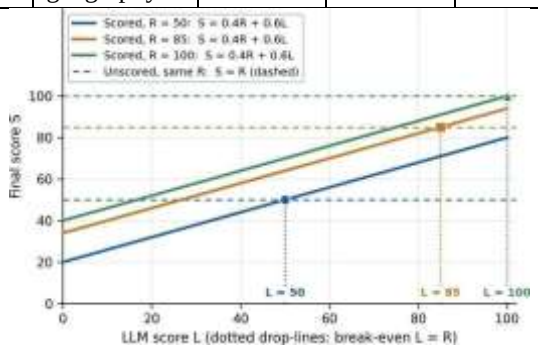


Fig. 3. Final score of LLM-scored candidates (solid) against rule-only candidates with the same *R* (dashed). Dots mark the break-even

L, where the two are equal.

Three observations follow from the formulas, as Figure 3 and Table 3 show.

Break-even equals *R*. An LLM-scored candidate ranks below a rule-only one with the same *R* whenever its *L* is below

R. For $R = 85$, any *L* under 85 does so.

© 2026 IJSRET

The recency term shifts the list. It adds 4 points to every scored candidate and 10 to every rule-only one, so it favours rule-only candidates by 6. Without it the break-even would be R - 10. It is also the only way to reach R = 100, since the three real matches total 90.

The fallback value of 88. A candidate whose LLM call failed receives $S = 0.4R + 52.8$, anywhere from 56.8 to 92.8. That is above a genuine candidate with the same R unless the genuine L exceeds 88, and above a rule-only candidate for every R below 88. The fallback therefore behaves like a strong model score.

The results are purely based on the arithmetic of the scoring function. The results illustrate that the score given uses two criteria, but it does not provide evidence of which ranking is more effective for investors, as there is no labelled data in the archive for testing..

IV. HOW THE SYSTEM WORKS

1. Profile Extraction

The input channel sends the text, either pasted or extracted from the PDF deck using PyMuPDF, to the language model and gets the profile back in JSON format. In case of failure to execute the model, a keyword-based fallback will fill the profile. Since a valid JSON profile does not guarantee correctness [7], the entrepreneur can revise it before matching; additionally, any empty fields left by the model will be filled with the defaults discussed in Section 3.2.

2. Drafting, Approval And Sending

The system generates one draft per contact and associates it with the status draft. The send route and OutreachService do not send any message if its status is not approved, enforcing the PRD's approval policy. While the approval is a client button in the provided archive, there is no backend route that assigns the status. Hence, an approval route could be present in the archive. The sending process happens via Resend, where each delivered email consumes one credit..

3. Reply Handling

Resend posts reply events to the webhook. Each reply is stored and assigned one of five labels by the language model, with a keyword fallback if the model call fails. The webhook reads an x-signature or x-resend-signature header, whereas Resend documents svix-id, svix-timestamp and svix-signature headers [12]. The signature check therefore needs to be aligned with the provider's scheme before real emails are sent (Section 11). Stored replies are not yet used by the ranking step.

4. Dataset

Although the README file and the provenance information state that the data set contains a list of 100 companies and 100 partners gathered from publicly available resources, there are actually more records (Table 4).Of the 969 investor rows, 100 cite public sources such as SEC Form ADV filings and firm websites. The remaining 869 cite Investor list.xlsx, a four-sheet workbook whose rows an ETL script merges and deduplicates (1,151 rows in, 869 investors out).

Table 4. Profile of the Advibe investor dataset and of Sheet1, the first sheet of the supplied workbook.

Measure	Value
Investor rows / person rows	969 / 1,242
Investor rows from public sources / from the workbook	100 / 869
Types: venture capital / angel syndicate / real-estate PE or REIT / family office or LP	770 / 159 / 14 / 10 (the other 16: micro VC 6, accelerator 5, growth equity 3, corporate VC 2)
Workbook rows with identical stage and sector tags	845 of 869 (97.2%)
Person rows marked verified	1,128 of 1,242 (90.8%)
Workbook Sheet1: records / with email / phone / website / LinkedIn	114 / 114 / 76 / 72 / 112
Sheet1 email type: corporate only / mixed / personal only	79 / 18 / 17 (14.9%)
Sheet1 location in India	96 of 114 (84.2%)

Sheet1's contact fields are largely complete. Email is present in all 114 records and LinkedIn in 112 (98.2%, counting only valid-looking links), and company and location are also present in nearly every record. Phone and website are present in about two thirds of records (66.7% and 63.2%).

Two properties of the data affect the scoring in Section 5. First, the ETL assigns every venture-track row the same stage list (Seed, Series A, Series B) and the same sector list (AI/ML, B2B SaaS, Fintech, Healthcare), while real-estate and LP rows receive lists of their own. For 845 of the 869 workbook rows the 40 stage points and 35 sector points are therefore the same, and geography is the signal that separates investors. Sheet1 suits this: if a founder is in India, 84% of its records match. Sheet1 has no stage, sector, cheque-size, source or verified column, so these tags come from the script, not from the workbook. Second, the ETL marks a contact as verified whenever an email address is present, so the 90.8% verified rate means that an email is on file.

5. The Interface

We ran the web client on a recorded session and captured two of its screens (Figure 4). After the founder uploads a

© 2026 IJSRET

pitch deck, the agent shows the extracted profile with an Edit details button, and a Find matching investors button runs the match route. In the recorded run the profile was an HR-tech company raising $1.5M, and the agent returned six matched investors, each with a fit percentage, role, firm, region tags, a masked email and an external link to the investor's profile (Fig. 4a). A masked email is unlocked in the Leads screen for one credit, which the interface calls a Spark. The Enrich a List screen takes pasted firm names or a CSV file and returns, for each firm, a partner, a masked email, a verified status and a LinkedIn profile link (Fig. 4b), and following a link opens the person's LinkedIn page. The lookalike search (Twin Finder), the campaign manager and the Command Center also load; on our new account the last two showed no data.

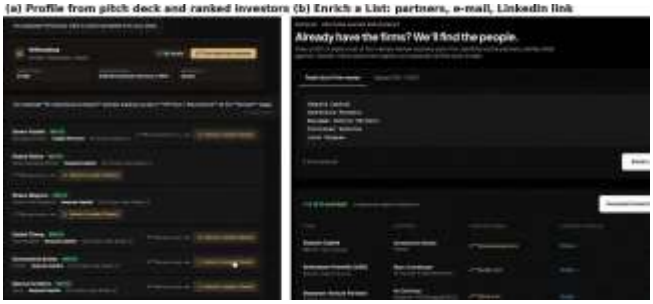


Fig. 4. Advibe web client in operation: (a) profile extracted from an uploaded pitch deck with ranked investor matches; (b) Enrich a List results with partner, masked email and LinkedIn profile link. Account details removed.

6 Evaluation And Discussion

1. Evaluation Of The Rule Stage

The LLM stage needs model access that this study did not have, so we evaluated the stage that decides which investors reach the model: retrieval and the rule score *R*. We wrote 20 test company profiles covering five stages, ten sectors and four regions, each with a cheque range, and pooled the top 10 of three rankers over the 969-investor catalogue: random investors; *R* (Equation 1); and the PRD rule score in DataService, which adds cheque-size fit and fuzzy matching. Ties were broken at random with a fixed seed, because the route's SQL has no ORDER BY. The 536 distinct pooled entries were shuffled with method labels removed and rated by an LLM judge (Claude) with a written rubric: 2 for a plausible investment firm whose stage, sector, region (or Global) and cheque range all fit the profile; 1 for a plausible firm with one soft or missed criterion, or an identity that could not be confirmed; 0 for an entry that is not an investment firm, or that misses two or more criteria. We report precision at 10 (rating of at least 1, and strictly 2) and NDCG at 10 with the ideal ranking taken from the pooled entries, with 95% confidence intervals over profiles. The judge reads the same catalogue fields as the rankers and uses general knowledge to spot non-investors, and no human has yet rated the lists, so the numbers are indicative (Section 11).

Table 5. Precision at 10 and NDCG at 10 from LLM-judged ratings, mean ± 95% confidence interval over 20 test profiles.

Ranker (top 10)	P@10 (rating 1 or 2)	P@10 strict (rating 2)	NDCG@10
Random	0.42 ± 0.12	0.11 ± 0.07	0.26 ± 0.09
Advibe rule score R	0.91 ± 0.07	0.72 ± 0.13	0.84 ± 0.08
PRD rule score	0.93 ± 0.05	0.79 ± 0.10	0.91 ± 0.06

As Table 5 shows, both rule scores rank far above random (precision at 10 of 0.91 and 0.93 against 0.42; NDCG of 0.84 and 0.91 against 0.26), and paired bootstrap intervals for both gaps exclude zero. The difference between *R* and the PRD score is small at this sample size (precision +0.02, interval -0.05 to +0.11; NDCG +0.07, interval -0.01 to +0.15), although the PRD score is ahead on the strict measure (0.79 against 0.72). Random lists drew 58% zero ratings against 9% and 7% for the two rule scores; many of these were catalogue entries that are not investment firms, such as operating companies and banks recorded as venture capital. Two properties of *R* follow from the catalogue. First, scores tie: in 12 of the 20 profiles all ten investors at the top of *R* share one value, and a median of 11.5 candidates (up to 148) reach $R = 100$, so which of them reach the LLM depends on database order, not on the score. Second, geography is matched as an exact string, and the catalogue holds 301 distinct region strings. For a company in India, 35 investors list “India” exactly, whereas 271 mention India in some form, such as a city name, so exact-string matching awards geography points to only some of the investors who mention India. The LLM stage and the blend were not evaluated. The evaluation scripts, ratings and rubric accompany this paper.

2. Discussion

Small scoring choices have large ranking effects. The recency term, the fallback value of 88 and the handling of ties are each a line of code, yet their effects on the ranking are large. Tie handling determines which of the tied top candidates reach the model, while the recency term and the fallback value determine whether an unreviewed candidate can outrank a reviewed one (Sections 5 and 9.1).

The data, not the formula, limits discrimination. However well its weights are calibrated, the rule score cannot separate investors whose stage and sector tags are identical (Table 4). Improving the tags and normalising region names would let all three matching signals contribute.

© 2026 IJSRET

V. CONCLUSION

Advibe treats fundraising outreach as one continuous pipeline, from a company description to a labelled reply, and the prototype shows that such a pipeline can be assembled quickly: profile extraction, retrieval, two-stage ranking with a stored rationale, drafting, an approval-gated send and reply labelling, all behind a working web interface. Its fit score is a short formula that can be recomputed by hand, and the analysis in Section 5 gives the break-even point for every reachable rule score. We would call Advibe an implementation prototype rather than a validated product. The PRD's targets of a 35% reply rate, a match latency under 1.2 seconds and 95% inbox placement have no measurement behind them yet.

The evaluation covers the rule stage only. The next step is human ratings of the full ranking, checked against the LLM judge on a sample: independent relevance ratings for each test profile's candidates, precision at 10 and normalised discounted cumulative gain (NDCG) against those ratings for the rule-only, LLM-only and blended rankers, latency per request, and a hand-labelled sample of replies to score the five-way classifier.

1. Limitations And Future Work

Four limits apply to this study. We did not run the backend, call a model or send an email ourselves, so behavioural statements rest on the code and one recorded demonstration session. The evaluation in Section 9.1 covers the rule stage, uses 20 profiles written by the authors and ratings from an LLM judge that reads the same catalogue fields as the rankers, with no human raters yet, so its numbers are indicative. We examined one archive snapshot, and the deployed database may differ. The comparison in Table 1 rests on public pages and one outside review [1], with no testing with founders.

The prototype also has known gaps, and these set the agenda for future work:

1. Scoring. Since the match route and DataService function have their own respective weights and both candidates scored using rules and the model score are sorted together, S will become an absolute and consistent measurement, if there is one scoring function for the system based on the PRD formula, where region names are normalized, tie breaking is done using the rule, and recency is calculated using data.

2. Data. The stage and sectors tags remain consistent within each track, a contact will be deemed as verified if the email address is available, while an investor without a corresponding person will be assigned a placeholder contact. The tags drawn from actual investor records, a legitimate verification process, and elimination of placeholder contacts will ensure the data conforms to the provenance statement.

3. Approval and sending. There must be a backend route for approving this status, and a send request without a Resend flag must not be counted as sent..

4. Webhook and email rules. Signature verification should follow the provider's scheme [12], and drafts should carry a sender address and an unsubscribe link [13].

5. Learning from replies. Stored replies should feed back into ranking, and the rationale should reach the draft prompt.

6. Placeholder modules. The lookalike search, the dashboard figures and scheduled runs are currently placeholders and need server-side implementations.

7. Evaluation. Human ratings should be collected so that the protocol described in Section 10 can be applied to the full ranking.

REFERENCES

1. Verabro reviews, pricing and demos. Software Advice (accessed 9 Oct 2026). https://www.softwareadvice.com.au/software/552579/Verabro
2. Foundersuite: fundraising CRM and investor database. Foundersuite. https://www.foundersuite.com
3. OpenVC: open investor database for founders. OpenVC. https://www.openvc.app
4. Signal: investor search and fundraising tracking. NFX. https://signal.nfx.com
5. Tracxn: startup and market intelligence platform. Tracxn Technologies. https://tracxn.com
6. Is ChatGPT good at search? Investigating large language models as re-ranking agents. Sun, W., Yan, L., Ma, X., et al. In: Proceedings of EMNLP 2023, pp. 14918–14937 (2023). https://arxiv.org/abs/2304.09542
7. Structured Outputs. GroqDocs (accessed 8 Oct 2026). https://console.groq.com/docs/structured-outputs
8. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. Zheng, L., Chiang, W.-L., Sheng, Y., et al. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track (2023). https://arxiv.org/abs/ 2306.05685
9. An analysis of the requirements traceability problem. Gotel, O.C.Z., Finkelstein, A.C.W. In: Proceedings of the First International Conference on Requirements Engineering, pp. 94–101. IEEE (1994). https://doi.org/10.1109/ICRE.1994.292398
10. Do developers benefit from requirements traceability when evolving and maintaining a software system? Mäder, P., Egyed, A. Empirical Software Engineering

© 2026 IJSRET

20(2), 413–441 (2015). https://doi.org/10.1007/s10664-014-9314-z

11. HMAC: keyed-hashing for message authentication. Krawczyk, H., Bellare, M., Canetti, R. RFC 2104 (1997). https://www.rfc-editor.org/info/rfc2104
12. Verify webhooks requests. Resend (accessed 8 Oct 2026). https://resend.com/docs/dashboard/webhooks/verify-webhooks-requests
13. CAN-SPAM Act: a compliance guide for business. U.S. Federal Trade Commission (accessed 8 Oct 2026). https://www.ftc.gov/business-guidance/resources/can-spam-act-compliance-guide-business
14. The Llama 3 herd of models. Llama Team, AI @ Meta. arXiv preprint arXiv:2407.21783 (2024). https://arxiv.org/abs/2407.21783
15. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. DeepSeek-AI. arXiv preprint arXiv:2501.12948 (2025). https://arxiv.org/abs/2501.12948

© 2026 IJSRET