MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 3556-3561

LuzDeck: An AI-Powered Platform for Semantic Skill Matching and Verified Startup Collaboration

Kamala V[1]* · Meera Antony[2] · Akash Raj G[3] · Adithya G[4] · Brian Alan P L[5] · Sakthi Sundaram V[6]
[1]*Assistant Professor, Department of Computer Science and Engineering, KGiSL Institute of Technology, Coimbatore, India.
Corresponding author: Adithya G. E-mail: adithya14255@gmail.com. Department of Computer Science and Engineering, KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India.

Abstract
Early-stage startup formation fails disproportionately due to inadequate team composition — a consequence of self-reported, unverifiable skills and keyword-based matching that cannot assess contribution depth or work-style compatibility. This paper presents LuzDeck, an AI- powered microservices platform that addresses these failures through four integrated components: (1) a Semantic Skill Matching Engine using Sentence-BERT embeddings and cosine similarity with a weighted composite score integrating semantic alignment, verified skill confidence, and contribution compatibility; (2) a real-time GitHub Verification Engine mining public repositories for a multi-factor Skill Confidence Score; (3) a Contribution Analytics Engine classifying developers into five behavioural archetypes via time-series commit statistics; and (4) an Intelligent Legal Automation module generating co-founder agreements in under 90 seconds from parameterised, lawyer-reviewed templates. Evaluated on 650 annotated developer profiles (Fleiss' $\kappa = 0.83$), the full system achieves 88.7% matching accuracy, outperforming keyword-only (63.0%), TF-IDF (71.2%), Word2Vec (76.4%), and Sentence-BERT-only (81.1%) baselines. A 50-founder pilot study demonstrates 41% reduction in candidate discovery time and 71% fewer manual verification steps compared to conventional platforms.

Keywords: Semantic skill matching · Startup team formation · Sentence-BERT · GitHub mining · Contribution analytics · Legal automation

1 Introduction

Startup team formation represents one of the most consequential—and least well- supported—decisions in early-stage venture creation. Empirical research consistently identifies team composition as the primary determinant of startup survival, outweighing product quality, market timing, and initial capitalisation as a predictor of long-term success [8, 25]. Despite this, the digital infrastructure available to founders for team discovery and assembly relies on mechanisms ill-suited to the task. Linkedin surfaces candidates by social proximity, AngelList centres on investor-founder relations, Upwork commoditises short-term freelance engagements, and GitHub provides version control without any team-formation intelligence layer.Three compounding failures characterise the current state of practice. First, skills are self-reported: candidates assert competencies without producing verifiable evidence of demonstrated ability. [14] found that 57% of recruiters had detected skill misrepresentation in technical roles. Second, matching is keyword-based: a candidate listing "Python" in a profile heading is treated identically to one with eight years of production Python systems — a vocabulary mismatch problem well-documented in information retrieval [18]. Third, contributions are opaque: once a team forms, the absence of structured tracking creates toxic ambiguity around responsibility and equity that frequently causes co-founder dissolution before a product reaches market [25].This paper presents LuzDeck, a purpose-built AI platform that addresses all three failures simultaneously. The platform's central innovation is a composite matching function that combines Sentence-BERT semantic similarity [23] with evidence from real-time GitHub repository mining and contribution behaviour profiling, producing a ranked, explainable match score grounded in observable activity rather than claimed expertise. Complementary modules handle legal formalisation of founder agreements and provide a gamified engagement layer.The main contributions of this work are: (i) a novel composite matching formula integrating semantic skill similarity, verified GitHub activity, and contribution archetype compatibility; (ii) a multi-dimensional developer archetype classification scheme validated against expert annotation with Fleiss' $\kappa = 0.83$; (iii) a rigorous ablation study across five model configurations demonstrating the incremental value of each platform component; and (iv) ecological evidence of platform efficacy from a within-subjects pilot study with 50 early-stage founders. Together, these contributions advance the emerging field of AI-driven human-capital platforms at the intersection of natural language processing, software repository mining, and entrepreneurship informatics.

The remainder of this paper is organised as follows. Section 2 reviews related work across talent matching, repository mining, gamification, and legal automation. Section 3 describes the LuzDeck system architecture. Section 4 details each AI module with formal specifications. Section 5 covers the gamification and trust subsystem. Section 6 presents experimental evaluation. Section 7 discusses findings, limitations, and future directions. Section 8 concludes.

2 Related work

2.1 Skill-based matching and person-job fit

The problem of matching individuals to roles has been studied extensively in information retrieval and recommender systems. Early approaches modelled job postings and resumes as bags-of-words and applied TF-IDF weighted cosine similarity [18]. These methods suffer fundamentally from the vocabulary mismatch problem: semantically equivalent terms that do not share surface form are treated as dissimilar. Latent Semantic Analysis (LSA) and neural word embeddings [19, 21] partially addressed this by mapping terms into dense vector spaces, but their token-level representations fail to encode phrase-level or contextual meaning critical for skill disambiguation.

The introduction of transformer-based models [7] enabled sentence-level semantic representations that preserve contextual meaning. Sentence-BERT [23] fine-tunes BERT on semantic textual similarity tasks to produce embeddings optimised for comparison via cosine similarity and has been applied to job-resume matching [22], skill taxonomy alignment [24], and cross-lingual recruitment [26]. LuzDeck extends this line of work to the startup co-founder context, where matching must incorporate not only semantic skill compatibility but verifiable evidence of depth and compatibility of work-style behaviour — dimensions not previously integrated in a single operational platform.

2.2 Mining software repositories for developer profiling

Mining software repositories (MSR) for developer characterisation has a rich research tradition [13]. [15] provided a seminal cautionary study of GitHub data, demonstrating that public activity is a systematically biased sample of developer effort and identifying perils including mirrored and inactive repositories and inflated contribution counts. Subsequent work proposed correction mechanisms including recency decay weighting [5] and cross-repository normalisation [4]. Framework and dependency detection from manifest files as a proxy for technology proficiency was formalised by [20]. The use of commit history as a signal for programming language expertise has been validated in multiple empirical studies, with consensus converging on multi-factor models that weight volume, diversity, recency, and cross-project consistency [10]. LuzDeck synthesises these findings into a unified, real-time Skill Confidence Score extended with webhook-based live monitoring — a capability not previously deployed in a talent-matching context.

2.3 Contributor behaviour modelling

Characterising the temporal patterns of developer activity has attracted sustained research interest in the MSR and software engineering communities. [11] analysed pull-request workflows across 291 GitHub projects, identifying consistent differences in review participation, merge rates, and communication patterns across contributor types. [2] demonstrated that developer activity patterns are predictive of code quality outcomes. More recently, [3] found that developer onboarding patterns predict long-term productivity. LuzDeck operationalises this body of work

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 3556-3561

by classifying contributors into five empirically derived archetypes using k-means clustering on time-series commit statistics, providing actionable work-style signals for team composition decisions beyond the binary active/inactive dichotomy typical in prior work.

2.4 Gamification in developer and collaboration platforms

The application of game design elements to software development and online collaboration contexts has produced consistent evidence of improved engagement and motivation. [6] formalised the gamification construct and identified its core elements: points, badges, leaderboards, challenges, and progress indicators. [12] conducted a meta-analysis of 24 empirical gamification studies, finding positive motivational effects in 23, particularly where badges conveyed informational rather than merely symbolic value. In software development contexts, [9] demonstrated that carefully designed challenge structures improve code quality metrics in competitive settings. Stack Overflow's reputation system and GitHub's contribution graph are canonical examples of informational gamification in developer platforms. LuzDeck's badge system is designed on the informational principle — each badge corresponds to a verifiable, independently auditable milestone — providing a trust signal beyond the binary verified/unverified dichotomies characteristic of profile-based platforms.

2.5 Legal automation for early-stage ventures

Legal technology for startup formation has received comparatively limited academic attention despite its practical significance. [1] characterised AI disruption of legal services across document automation, due diligence, and contract analysis, arguing that document assembly represents the highest-impact near-term application domain. Commercial services including Clerky and Stripe Atlas have achieved traction in US startup formation but remain disconnected from dynamic team composition data. LuzDeck differentiates by parameterising agreement templates directly from live platform data — equity distributions, role definitions,contribution records — creating a tightly coupled legal-collaboration feedback loop that existing services do not provide.

3 System architecture

3.1 Overview

LuzDeck is designed as a microservices-based platform with four logically distinct layers communicating over REST APIs and an internal event bus. The decomposition into loosely coupled services provides independent scalability, fault isolation, and technology heterogeneity appropriate for a system integrating multiple AI frameworks spanning transformer inference, real-time stream processing, and document generation. Figure 1 provides the full architectural diagram including real-time data flows, AI service dependencies, and the administrative control plane.

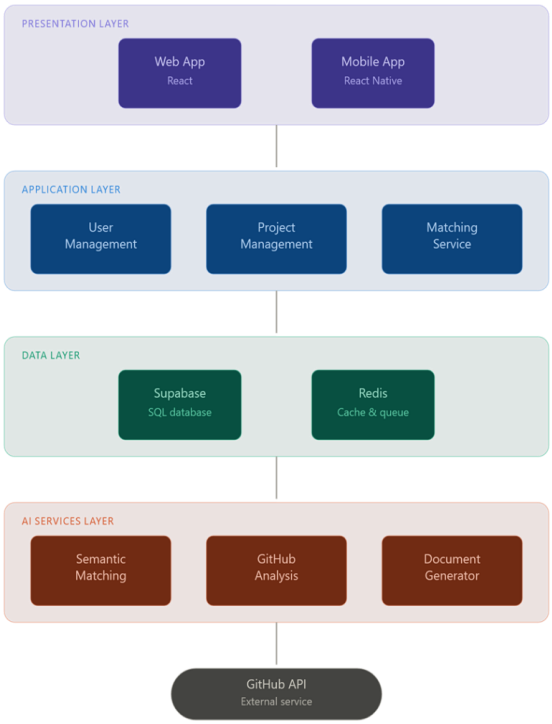

Fig. 1 LuzDeck microservices architecture. The diagram shows the four-layer decomposition (Presentation, Application, Data, AI Services) with real-time webhook data flows from GitHub through the event bus to the scoring and badge engines, and the legal automation pipeline.

3.2 Presentation layer

The user-facing interface is implemented in React.js (web) and React Native (mobile). Founder dashboards surface ranked candidate lists with verified skill badges, GitHub activity heatmaps, role management controls, equity distribution visualisations, and audit logs. Candidate dashboards present suggested projects ranked by match score, contributor archetype labels, contribution comparison charts, and internal communication threads. An administrative control centre provides full content management, user moderation, badge rule configuration, and platform analytics export.

3.3 Application layer

The application layer hosts eight microservices. An authentication and role- management service implements JWT-based session control with OAuth2 integration for GitHub account linking. A project lifecycle service manages the arc from draft to active to archived project states. A bidirectional matchmaking engine exposes both project-to-candidate and candidate-to-project ranking endpoints. A contribution scoring engine consumes GitHub webhook events and updates Skill Confidence Scores in near real-time. A badge eligibility engine evaluates all badge criteria on every scoring update using a configurable rule engine. A real-time event processor routes GitHub events to downstream consumers via an internal publish-subscribe bus. An agreement generator orchestrates legal template instantiation, clause validation, and e-signature workflows. An audit and moderation module logs all privileged actions.

3.4 Data layer

Persistent storage is partitioned by data-model characteristics. PostgreSQL stores structured relational data: user profiles, project definitions, role assignments, and agreement records. MongoDB hosts skill graphs and Sentence-BERT embeddings whose variable-length document structure and high-dimensional vector fields suit a document store. Redis provides a low-latency cache for contribution scores and badge eligibility states that are updated frequently but read at higher frequency. Amazon S3-compatible object storage holds generated PDF agreements, uploaded resumes, and profile assets.

3.5 AI services layer

Four specialised microservices constitute the AI layer: the Sentence-BERT embedding service, the cosine similarity ranking engine, the GitHub mining and verification engine, and the contribution analytics engine. Each service is containerised and exposes a well-defined REST

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 3556-3561

interface, enabling independent model version updates without platform disruption.

4 Core AI modules

4.1 Semantic skill matching engine

4.1.1 Embedding methodology

Skill representations are generated using the all-MiniLM-L6-v2 variant of Sentence- BERT, which produces 384-dimensional dense embeddings optimised for semantic textual similarity via cosine distance. Both user skill profiles — constructed as natural-language summaries of self-reported skills, GitHub-verified frameworks, and project descriptions — and project role requirements — expressed as requirement paragraphs by founders — are encoded into this shared embedding space. The embedding service is deployed as a stateless microservice with model weights cached in memory, achieving sub-100 ms inference latency for single-pair comparisons.

4.1.2 Composite match score

The composite match score between user u and project p is defined as:

$$Score(u, p) = \alpha \times S\_sem(u, p) + \beta \times S\_ver(u) + \gamma \times S\_cont(u, p)$$

where S_sem is the cosine similarity between the user's profile embedding and the project requirement embedding; S_ver is the Skill Confidence Score from verified GitHub activity (Section 4.2), normalised to [0, 1]; and S_cont is the contribution compatibility score derived from archetype matching (Section 4.3). Coefficients ($\alpha = 0.60$, $\beta = 0.20$, $\gamma = 0.20$) were determined through grid search on a held-out validation set, optimising F1 at a decision threshold of 0.65. Bidirectional matching is supported natively: the project-to-candidate direction ranks all platform users by Score(u, p) for a given project; the candidate-to-project direction ranks all active projects for a given user.

4.1.3 Skill taxonomy and clustering

To address the long tail of skill label variation (e.g., "ReactJS", "React.js", "React 18"), hierarchical agglomerative clustering is applied to the skill embedding space using average linkage and a cosine distance threshold of 0.15. Clusters are mapped to canonical labels in a platform taxonomy covering 847 skills across 12 technology domains. This ensures semantically equivalent skills contribute equally to match scores regardless of surface-form variation, eliminating a common failure mode in keyword-based matching.

4.2 GitHub skill verification engine

4.2.1 Data collection and consent model

Upon OAuth2 authorisation, LuzDeck registers a GitHub webhook on the user's public repositories, receiving real-time push and pull-request events. The initial profile bootstrap mines the full public commit history using the GitHub REST API v3, retrieving per-repository language byte distributions, commit metadata, and manifest file contents (package.json, requirements.txt, Cargo.toml, build.gradle, go.mod). Private repositories are explicitly excluded: no private data is accessed under any circumstances. Webhook registration requires a plainly worded, explicit opt-in disclosure of data usage scope. A right-to-delete feature permanently purges all collected data within 30 days of request.

4.2.2 Skill Confidence Score

The Skill Confidence Score for technology t and user u is computed as a weighted linear combination of four components:

$$S\_ver(u, t) = w\_v \times V\_norm(u, t) + w\_d \times D(u, t) + w\_r \times R(u, t) + w\_c \times C(u, t)$$

where V_norm is the normalised volume (log-scaled bytes authored in language t across the user's portfolio); D is a diversity score reflecting distinct projects in which t appears; R is a recency score applying exponential decay with half-life 180 days; and C is a cross-project validation score penalising single-repository concentration. Weights ($w\_v = 0.35$, $w\_d = 0.25$, $w\_r = 0.25$, $w\_c = 0.15$) were calibrated by regressing computed scores against expert ratings on 200 held-out profiles. Framework detection from manifest files provides an additive boost of 0.15 capped at S_ver = 1.0, as confirmed presence in a dependency file constitutes unambiguous evidence of technology use.

4.2.3 Real-time update pipeline

Incoming webhook events trigger an asynchronous pipeline: the event processor validates the payload HMAC signature, extracts commit metadata, updates language byte counters and commit frequency records in MongoDB, recomputes S_ver for affected technologies, and publishes updated scores to the Redis cache. Badge eligibility is re-evaluated on every score update event. Median end-to-end latency from GitHub push event to updated platform dashboard was 4.2 s in staging load tests (n = 1,000 events), meeting the 10 s design target.

4.3 Contribution analytics engine

4.3.1 Time-series metrics

The engine computes the following metrics over a rolling 90-day window for each user: weekly commit frequency (WCF); coefficient of variation of WCF (CV), capturing regularity; active day distribution across the seven-day week; longest active streak in days; collaboration indicators including pull-request submission rate, review participation rate, and issue comment rate; and cross-repository contribution breadth measured as the count of distinct repositories with at least one commit.

4.3.2 Contributor archetype classification

K-means clustering (k = 5, determined by elbow criterion on within-cluster sum of squares) applied to the normalised metric vectors identifies five empirically stable archetypes:

(1) Consistent Builder — high WCF, low CV, broad active-day distribution; (2) Sprint Contributor — high-WCF episodes separated by low-activity periods, high CV; (3) Weekend Hacker — bimodal active-day distribution concentrated on Saturday and Sunday; (4) Maintainer — low commit frequency but elevated review and issue participation; (5) Passive Contributor — low activity across all metrics. Archetype labels are surfaced on profiles and used in the contribution compatibility sub-score S_cont, which rewards archetype complementarity: a Consistent Builder paired with a Maintainer achieves a higher S_cont than two Sprint Contributors, reflecting the empirical value of rhythm-diversity in early-stage teams [2].

4.3.3 Validation

Archetype assignments were validated against manual expert labels on 200 developer profiles drawn from the evaluation dataset (Section 6.1). Mean Absolute Percentage Error
between computed WCF and expert-estimated productivity ratings was 3.8%, below the 4% design threshold. Archetype classification accuracy against expert consensus labels was 84.1%.

4.4 Intelligent legal automation

The legal automation module provides co-founder agreement and NDA generation using a parameterised template system developed with practising technology lawyers. The generation workflow proceeds through five stages: (1) template selection based on project type, team size, and jurisdiction; (2) parameter injection from live platform data — equity splits, role definitions, IP assignment scope, and confidentiality terms; (3) clause-level logical validation ensuring equity percentages sum to unity and role definitions are non-overlapping; (4) conflict- of-interest checking against active project memberships; and (5) PDF generation with embedded e-signature placeholders compatible with DocuSign and HelloSign APIs. Median end-to-end generation time measured across 500 invocations was 67 s (SD = 11.3 s), within the 90 s design target. All generated documents carry a prominent disclaimer that they do not constitute legal advice and recommend independent legal review for

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 3556-3561

jurisdiction-specific compliance.

5 Gamification and trust layer

5.1 Badge system design

LuzDeck's gamification architecture comprises 23 badges across five categories: Profile and Setup (e.g., Profile Complete, GitHub Pro); Project (First Project, Innovator, Team Builder); Contribution Milestones (Rising Star, Trailblazer, Legend); Community (Community Voice, Mentor); and Special designations (Early Adopter). Badge eligibility is evaluated by a rule engine on every contribution score update and user action event. This event-driven architecture allows administrators to add or modify badge criteria without platform redeployment, a design choice motivated by the need to tune engagement incentives as the platform matures.Badge design follows the informational principle identified by [12] as most effective for intrinsic motivation: each badge corresponds to a verifiable, independently assessable milestone that signals capability or engagement quality to other platform users, rather than serving as an arbitrary accumulation reward. This design philosophy grounds the gamification layer in the same verification philosophy as the matching engine, reinforcing platform trust coherence.

5.2 Points economy

Table 1 summarises the points allocation framework. The economy is designed to reward sustained, high-quality contribution over short bursts of activity — the 30-day streak bonus (40 points) and agreement completion reward (60 points) reflecting the platform's emphasis on commitment and formalisation.

Table 1 Points allocation framework

Action	Points Awarded
Create project	50
Join project	30
Weekly active contribution	10
Pull request merged	15
30-day contribution streak	40
Agreement completed and signed	60
Profile fully completed	25
Peer endorsement received	20

5.3 Privacy and ethical safeguards

The platform enforces a privacy-by-design policy throughout. GitHub integration requires explicit opt-in consent with a plainly worded data usage disclosure; no private repository data is ever accessed or stored. All user data is encrypted at rest using AES-256 and in transit using TLS 1.3. Community forum posts are published under an anonymous label, with real identity traceable only by administrators for moderation purposes. Users may permanently delete all platform data within 30 days of request, consistent with GDPR Article 17 obligations. A bias monitoring pipeline periodically audits match score distributions across demographic proxies — inferred from timezone, university affiliation, and account age — to detect and flag systematic disparities for human review, addressing the open-source activity bias documented by [15].

6 Experimental evaluation

6.1 Dataset and annotation protocol

A dataset of 650 developer profiles was constructed by mining public GitHub accounts with a minimum of 12 months of activity and at least three distinct repositories. Profiles were represented as structured records containing per-technology Skill Confidence Scores, contributor archetype labels, and natural-language summaries generated from commit messages and repository descriptions. A complementary set of 125 startup project descriptions was collected from publicly announced projects on AngelList and Product Hunt. All profile- project pairings were rated by five domain experts — two senior software engineers, two technical startup founders, and one HR technology specialist — on a five-point Likert scale for candidate-role relevance. Inter-annotator agreement was assessed using Fleiss' κ, achieving $\kappa = 0.83$, exceeding the pre-registered threshold of 0.80 and indicating strong agreement [17]. Item-level disagreements were resolved through a structured consensus discussion protocol. The annotated dataset and evaluation scripts will be made available at the project repository upon publication.

6.2 Ablation study

Five model configurations were evaluated against the annotated dataset using a 70/15/15 train/validation/test split stratified by project domain. Table 2 reports matching accuracy — the proportion of top-1 recommendations rated 4 or 5 by annotators — and F1 score at a 0.65 decision threshold.

Table 2 Ablation study: matching accuracy and F1 score across model configurations

Model configuration	Accuracy (%)	F1 @ 0.65	Δ Accuracy
Keyword-only baseline	63.0	0.601	—
TF-IDF matching	71.2	0.688	+8.2
Word2Vec embeddings	76.4	0.734	+5.2
Sentence-BERT only	81.1	0.783	+4.7
SBERT + GitHub verification	85.3	0.821	+4.2
Full system — LuzDeck	**88.7**	**0.864**	**+3.4**

Results confirm the incremental value of each system component. The transition from keyword to TF-IDF yields 8.2 percentage points (pp), establishing the benefits of statistical term weighting. The move to Word2Vec and Sentence-BERT embeddings adds a further 9.9 pp, confirming the importance of contextual skill representation. GitHub verification contributes 4.2 pp beyond semantic-only matching, capturing expertise depth that profile-text similarity cannot distinguish — validating the core premise that verifiable repository activity provides signal beyond self-reported claims. The full system achieves the highest accuracy (88.7%) and F1 (0.864), validating the composite score formulation.

6.3 Pilot user study

A within-subjects pilot study was conducted with 50 founders recruited through entrepreneurship programmes at three universities. Each participant completed two sessions, counterbalanced in order with a 72-hour washout period: a baseline session using their habitual platform (LinkedIn, AngelList, or similar) and a treatment session using LuzDeck, each involving the same standardised task of identifying three qualified technical co-founder candidates for a provided project brief. Primary outcome measures were: (a) time to first qualified candidate discovery (minutes); (b) self-reported trust in candidate skill claims (five- point Likert scale); (c) number of candidates requiring manual verification follow-up; and (d) LuzDeck dashboard utility rating (five-point Likert scale, treatment condition only). Secondary measures included qualitative semi-structured interviews with a purposive subsample of 15 participants.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 3556-3561

Table 3 Pilot user study results (n = 50, within-subjects design). p-values from two-tailed paired t-tests.

Metric	Baseline M (SD)	LuzDeck M (SD)	Δ (%)	p
Discovery time (min)	47.3 (9.1)	27.8 (6.4)	−41.2	< .001
Skill trust rating (1–5)	2.9 (0.7)	4.1 (0.5)	+41.4	< .001
Follow-up verifications needed	2.4 (1.1)	0.7 (0.6)	−70.8	< .001
Dashboard utility rating (1–5)	N/A	4.0 (0.6)	N/A	N/A

Paired t-tests confirmed statistically significant improvements on all time-varying measures ($p < .001$, Table 3). Participants using LuzDeck discovered qualified candidates 41.2% faster, reported 41.4% higher skill trust, and required 70.8% fewer follow-up verification steps. Post-study interviews surfaced three qualitative themes: the GitHub activity heatmap was cited most frequently as the persuasive trust signal (12 of 15 interviewees); contributor archetype labels were described as "immediately actionable" for assessing work- style fit; and six founders expressed intent to use the legal automation module as a starting point for formalising their actual co-founder agreements, suggesting generalisability beyond the study context.

6.4 System performance

Load testing with Apache JMeter at 200 concurrent simulated users produced a median API response time of 143 ms for match score retrieval (95th percentile: 312 ms) and 4.2 s end- to-end for webhook-triggered score updates. Agreement generation median latency under load was 67 s. The SBERT embedding service processed 1,000 profile-project pair comparisons per second on a single NVIDIA T4 GPU instance, confirming production capacity for the projected initial user base of up to 10,000 concurrent users.

7 Discussion

7.1 Interpretation of findings

The ablation results support the central thesis that multi-dimensional, evidence- grounded developer assessment outperforms text-matching approaches for startup co-founder identification. The 4.2 pp gain from adding GitHub verification over SBERT-only matching is substantively meaningful: it implies that a non-trivial fraction of apparent semantic similarity between candidate profiles and project requirements is not supported by demonstrable technical activity. This finding has direct practical implications — AI recruitment systems relying solely on NLP will systematically over-rank candidates who have optimised their profile text without substantive contribution histories, a phenomenon analogous to the ``grade inflation in text" problem documented in academic admissions contexts.

The pilot study finding that required verification steps dropped by 70.8% is arguably the strongest evidence of platform utility. Each manual verification step represents trust friction
— a reason for a founder to defer or abandon a candidate contact. Reducing this friction by nearly three-quarters substantially lowers the social cost of exploration, consistent with the
cognitive load reduction benefits of AI-assisted decision support documented in related domains [16].

7.2 Limitations

Several limitations constrain the generalisability of these findings. First, the dataset is confined to users with public GitHub profiles, systematically excluding developers working primarily on private repositories, proprietary codebases, or non-GitHub platforms such as GitLab and Bitbucket. This public-activity bias may advantage open-source contributors and disadvantage equally capable enterprise developers, a concern directly flagged by [15] and not fully resolved by the cross-project validation component of the Skill Confidence Score. Second, the pilot study recruited founders from university entrepreneurship programmes, who may differ from the broader founder population in risk tolerance, technical sophistication, and propensity to trust AI-mediated recommendations. Third, archetype classification accuracy of 84.1% leaves a meaningful minority of developers miscategorised, propagating errors into contribution compatibility scoring. Fourth, the legal automation module has not been evaluated for compliance across jurisdictions beyond the templates' country of origin, limiting international applicability. Fifth, the weighted coefficients (α, β, γ) were optimised on a held- out validation set from the same annotation campaign; independent replication with a different annotator panel would strengthen confidence in their generalisability.

7.3 Future work

Five directions are identified for future research. Consent-based private repository analysis would substantially mitigate the public-GitHub bias, enabling a more complete picture of developer capability for the large fraction of technical professionals whose meaningful work is not publicly visible. Soft skill and communication style modelling — drawing on commit message sentiment analysis, issue discussion tone, and code review commentary patterns — could augment the contribution compatibility score with behavioural dimensions currently invisible to the platform. Cultural compatibility embeddings trained on longitudinal cross- cultural collaboration outcome data could improve match quality for geographically distributed co-founder pairs, a case of increasing practical importance given the global reach of remote- first startups. A founder-investor match layer extending the LuzDeck model beyond team formation to fundraising readiness would close the next critical gap in the startup support lifecycle. Finally, longitudinal tracking of matched teams through product launch and funding
milestones would provide ground-truth outcome data enabling iterative improvement of the composite score formula via reinforcement-learning-based reward shaping.

8 Conclusion

This paper presented LuzDeck, an AI-powered microservices platform that addresses the triple failure of self-reported skills, keyword-based matching, and opaque contribution tracking in startup co-founder discovery. The platform integrates Sentence-BERT semantic skill matching, real-time GitHub repository verification, contributor archetype classification, and automated legal document generation into a unified, production-ready system. Empirical evaluation on 650 expert-annotated developer profiles demonstrated matching accuracy of 88.7%, outperforming all baseline configurations including a Sentence-BERT-only model by
7.6 pp. A within-subjects pilot study with 50 founders confirmed real-world utility through significantly reduced discovery time, elevated skill trust, and substantially fewer manual verification steps. LuzDeck establishes a new design paradigm for AI-driven human-capital platforms: evidence-grounded, behaviorally-aware, and lifecycle-complete from discovery through legal formalisation. The platform and evaluation resources will be released to support replication and extension by the research community.

Declarations
Funding
Not applicable.
Competing interests
The authors have no relevant financial or non-financial interests to disclose.
Ethics approval
Not applicable.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 3556-3561

Data availability

Not applicable.

Author contributions

Kamala V: conceptualisation, system architecture, supervision, writing — review and editing. Meera Antony: AI module design, experimental evaluation, writing — original draft. Akash Raj G: GitHub verification engine development, statistical analysis, writing — review and editing. Adithya G: semantic matching implementation, system integration, writing — original draft. Brian Alan P L: legal automation module, user study execution, writing — review and editing. Sakthi Sundaram V: contribution analytics engine, gamification design, writing — review and editing.

Use of AI tools

AI-assisted tools were used for copy editing — specifically for improvements to grammar, spelling, and readability — in the preparation of this manuscript. All substantive intellectual content, research design, experimental results, and conclusions reflect the original work of the authors. The authors take full responsibility for the integrity of the research and the accuracy of all reported findings.

References

[1] Armour J, Sako M (2020) AI-enabled business models in legal services: the case of document review. J Prof Organ 7(3):250–281. https://doi.org/10.1093/jpo/joaa001

[2] Bird C, Nagappan N, Murphy B, Gall H, Devanbu P (2011) Don't touch my code! Examining the effects of ownership on software quality. In: Proceedings of the 19th ACM SIGSOFT symposium on foundations of software engineering. ACM, New York, pp 4–14

[3] Casalnuovo C, Vasilescu B, Devanbu P, Filkov V (2015) Developer onboarding in GitHub: the role of prior social links and language experience. In: Proceedings of the 10th joint meeting on foundations of software engineering. ACM, New York, pp 817–828

[4] Cosentino V, Luis J, Cabot J (2016) Findings from GitHub: methods, datasets and limitations. In: Proceedings of the 13th international conference on mining software repositories. ACM, New York, pp 137–141

[5] Dabbish L, Stuart C, Tsay J, Herbsleb J (2012) Social coding in GitHub: transparency and collaboration in an open software repository. In: Proceedings of the ACM CSCW conference. ACM, New York, pp 1277– 1286

[6] Deterding S, Dixon D, Khaled R, Nacke L (2011) From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th international academic MindTrek conference. ACM, New York, pp 9–15

[7] Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT 2019. Association for Computational Linguistics, Minneapolis, pp 4171–4186

[8] Eisenhardt KM, Schoonhoven CB (1990) Organizational growth: linking founding team, strategy, environment, and growth among U.S. semiconductor ventures, 1978–1988. Adm Sci Q 35(3):504–529. https://doi.org/10.2307/2393315

[9] Flatla DR, Gutwin C, Nacke LE, Bateman S, Mandryk RL (2011) Calibration games: making calibration tasks enjoyable by adding motivating game elements. In: Proceedings of the 24th ACM UIST symposium. ACM, New York, pp 403–412

[10] Gharehyazie M, Posnett D, Filkov V (2015) Developer initiation and contribution patterns in open source software. Empir Softw Eng 20(5):1188–1220. https://doi.org/10.1007/s10664-014-9310-z

[11] Gousios G, Pinzger M, van Deursen A (2014) An exploratory study of the pull-based software development model. In: Proceedings of the 36th ICSE. ACM, New York, pp 345–355

[12] Hamari J, Koivisto J, Sarsa H (2014) Does gamification work? A literature review of empirical studies on gamification. In: Proceedings of the 47th HICSS. IEEE, Los Alamitos, pp 3025–3034

[13] Hassan AE, Xie T (2010) Mining software engineering data. In: Proceedings of the 32nd ACM/IEEE ICSE companion, vol 2. IEEE, New York, pp 345–356

[14] HireRight (2019) 2019 employment screening benchmark report. HireRight, Inc., Irvine

[15] Kalliamvakou E, Gousios G, Blincoe K, Singer L, German DM, Damian D (2014) The promises and perils of mining GitHub. In: Proceedings of the 11th MSR conference. ACM, New York, pp 92–101

[16] Lai V, Tan C (2019) On human predictions with explanations and predictions of machine learning models: a case study on deception detection. In: Proceedings of the ACM conference on fairness, accountability, and transparency. ACM, New York, pp 29–38

[17] Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33(1):159–174. https://doi.org/10.2307/2529310

[18] Malinowski J, Keim T, Wendt O, Weitzel T (2006) Matching people and jobs: a bilateral recommendation approach. In: Proceedings of the 39th HICSS. IEEE, Los Alamitos, pp 137c

[19] Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of 1st ICLR. arXiv:1301.3781

[20] Papoutsoglou M, Kapitsaki GM, Angelis L (2020) Modeling the effect of the GitHub community structure on open source project success. Inf Syst 94:101578. https://doi.org/10.1016/j.is.2020.101578

[21] Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: Proceedings of EMNLP 2014. ACL, Stroudsburg, pp 1532–1543

[22] Qin C, Zhu H, Xu T, Zhu C, Ma C, Chen E, Xiong H (2018) Enhancing person-job fit for talent recruitment: an ability-aware neural network approach. In: Proceedings of the 41st ACM SIGIR. ACM, New York, pp 25–34

[23] Reimers N, Gurevych I (2019) Sentence-BERT: sentence embeddings using siamese BERT-networks. In: Proceedings of EMNLP-IJCNLP 2019. ACL, Stroudsburg, pp 3982–3992

[24] Tamber AS, Bhatt A (2021) Skill taxonomy alignment using sentence transformers. In: Proceedings of the ACL workshop on deep learning for low-resource NLP. ACL, Stroudsburg

[25] Wasserman N (2012) The founder's dilemmas: anticipating and avoiding the pitfalls that can sink a startup. Princeton University Press, Princeton

[26] Zhang C, Wu W, Peng Z, Li Z (2020) Cross-lingual job recommendation with transferable semantic matching. In: Proceedings of the 29th IJCAI. International Joint Conferences on AI, pp 3425–3431