International Journal of Web Applications

Print ISSN: 0974-7710
Online ISSN: 0974-7729

DLINE JOURNALS

IJWA 2026: 18 (1)
https://doi.org/10.6025/ijwa/2026/18/1/13-24

Digital Infrastructure and Developer Ecosystems: A Dual Dataset Framework for Cross-Domain Analysis of Technological Adoption

Hathairat Ketmaneechairat
Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Bangkok. Thailand
hathairat.k@cit.kmutnb.ac.th

ABSTRACT

This study introduces a dual dataset framework for analyzing the co-evolution of macro level digital infrastructure and micro level developer ecosystems. We integrate 45 years (1980–2020) of country level telecommunications indicators spanning mobile penetration, internet adoption, and broadband diffusion across 217 economies with a contemporary snapshot of 1,247 trending GitHub repositories, capturing real-time developer attention across 40+ programming languages. Through fork to star ratio analysis, we identify distinct behavioral archetypes: research oriented languages (R, Jupyter Notebook) exhibit 4–13× higher forking rates (median ratios 0.29–0.38), reflecting academic reuse patterns for method adaptation and reproducibility, while infrastructure languages (Rust, TypeScript) demonstrate star-dominant engagement (ratios <0.04), indicating production tool consumption via stable APIs rather than source modification. Critically, we document the emergence of compositional language ecosystems where polyglot architectures deliberately stratify languages by computational layer Rust for systems safety, Python for AI orchestration, and TypeScript for interfaces moving beyond "language wars" toward purpose driven symbiosis. These patterns reveal programming languages function not merely as syntactic tools but as socio technical coordination mechanisms governing community behavior. The framework enables novel cross domain inquiries linking national infrastructure quality to global innovation patterns, with implications for digital policy design and open source ecosystem stewardship.

Keywords: Digital Infrastructure, Developer Ecosystems, Cross Domain Analysis, Programming Languages, Fork to Star Ratio, Polyglot Architectures, Open Source Software, Socio Technical Systems

Received: 30 August 2025, Revised 18 December 2025, Accepted 24 December 2025

1. Introduction

The big data era has precipitated an explosion of heterogeneous datasets spanning diverse domains,

each characterized by distinct representations, distributions, scales, and densities. Effectively unlocking latent knowledge from these disparate yet potentially interconnected datasets constitute a paramount challenge in contemporary data science research [1]. Within recommender systems, data sparsity remains a persistent obstacle that degrades prediction accuracy and user experience. To mitigate this limitation, Cross Domain Recommendation (CDR) and Cross-System Recommendation (CSR) methodologies have emerged, leveraging auxiliary information such as ratings, reviews, user profiles, item attributes, and tags from data rich source domains to enhance performance in information sparse target domains.

A critical limitation of conventional CDR and CSR approaches lies in their adherence to a single target paradigm. These methods transfer knowledge unidirectionally from the source to the target domain, improving recommendations solely within the target domain while yielding no reciprocal benefit to the source domain [2] (Zhu, 2021). This asymmetry overlooks an important reality: in many real world scenarios, domains possess complementary information richness. One domain may excel in rating density while another offers richer textual reviews or more detailed user profiles. When such complementary strengths are strategically leveraged, simultaneous performance enhancement across multiple domains becomes feasible giving rise to dual-target and multi-target CDR frameworks.

Recent literature has formalized three advanced CDR scenarios aimed at mutual benefit across domains: Dual-Target CDR (DTCDR), Multi-Target CDR (MTCDR), and integrated CDR+CSR frameworks. Zhu et al. [3] (2023) demonstrated that these approaches significantly improve recommendation accuracy across all participating datasets simultaneously. Empirical validation by Liu et al. Pan Li, Munger, and Zhu et al (2019, 2020) [4, 5, 6, 7, 27] confirmed that dual target CDR and multi target CDR [8] [Cui et al., 2020] substantially outperforms both single domain baselines and traditional single target cross domain methods. Method ologically, Zhu et al. [9] (2018) introduced DCDCSR, a deep learning framework integrating matrix factorization with fully connected deep neural networks to enable joint cross domain and cross system recommendations. Munger et al. and Zhang et al. [(2019, 2023) [6, 10,11] subsequently proposed the Dual Adversarial Network for CDR (DA-CDR), which employs adversarial training to align latent factor distributions across domains and achieves state of the art accuracy against seven competitive baselines. Most recently, Su et al. [12] addressed distributional mismatches between source and target domains using explicit alignment techniques, thereby alleviating the effects of imbalanced sampling and enhancing the efficacy of knowledge transfer under severe data scarcity.

Despite substantial progress in recommendation systems, cross domain knowledge transfer for tabular data remains critically underexplored. Tabular datasets ubiquitous in finance, healthcare, and business analytics exhibit pronounced heterogeneity and often have limited sample sizes [13]. Consequently, classical machine learning algorithms consistently outperform deep learning approaches on tabular tasks [14, 15]. This performance inversion renders deep learning centric transfer techniques, developed primarily for image and text modalities, ill suited for tabular domains. Although nascent efforts in tabular knowledge transfer have emerged [16, 13], a formalized cross domain learning framework for supervised tabular settings remains absent from the literature. Ghosh et al. [17] proposed a comparative methodology for domains sharing prediction objectives, enabling cross domain insight transfer to enhance predictive analytics a promising direction requiring further elaboration and formalization.

Beyond recommender systems, domain specific modeling enriched with structured knowledge has demonstrated significant efficacy across diverse application areas. Cecchini et al. [18] (2010) developed a

domain specific kernel for support vector machines to detect financial fraud in publicly listed companies, substantially improving classification performance. Similarly, domain knowledge integration enhanced predictive accuracy in manufacturing systems [19] and fashion retail demand forecasting. [20]. Chen et al. [21] (2024) recently demonstrated that incorporating economic no arbitrage constraints into neural networks improve s asset pricing predictions, illustrating how structural domain knowledge can regularize data driven models. Contemporary research increasingly emphasizes formalizing knowledge elicitation processes [22] and developing representations amenable to integration with machine learning architectures [23]. Successful implementations span image understanding [24 vehicle trajectory prediction [25], and complex problem-solving in ill-structured domains [26]

These advances collectively underscore a fundamental principle: synergizing domain specific knowledge with transfer learning mechanisms yields superior performance compared to purely data driven approaches. Yet this principle remains underexploited in cross domain recommendation for tabular data. Bridging this gap by developing frameworks that respect the characteristics of tabular data while enabling bidirectional knowledge exchange across complementary domains represents a promising frontier for future research, with significant practical implications for data scarce application domains.

2. About This Work

This study introduces a dual dataset framework for analyzing the co-evolution of macro level digital infrastructure and micro level developer ecosystems. We harmonize 45 years (1980–2020) of country level telecommunications indicators spanning mobile penetration, internet adoption, and broadband diffusion across 217 economies with a contemporary snapshot of GitHub trending repositories capturing real time developer attention patterns across 40+ programming languages. Through systematic analysis of engagement metrics, particularly fork to star ratios, we identify distinct behavioral archetypes: research oriented languages (R, Jupyter Notebook) exhibit 4–13× higher forking rates reflecting academic reuse patterns (method adaptation, reproducibility), while infrastructure languages (Rust, TypeScript) demonstrate star dominant engagement indicative of production tool consumption. Critically, we document the emergence of compositional language ecosystems where polyglot architectures deliberately stratify languages by computational layer Rust for systems safety, Python for AI orchestration, and TypeScript for interfaces moving beyond “language wars” toward purpose driven symbiosis. This work provides empirical grounding for understanding how digital infrastructure shapes innovation patterns and how language choice functions as a social-technical coordination mechanism governing community behavior in open source development.

3. Dataset Description

This data descriptor introduces two complementary longitudinal datasets enabling systematic analysis of technological adoption across macro level digital infrastructure and micro level software development ecosystems. The first dataset (*Final.csv*) comprises 45+ years (1980–2020) of country level telecommunications indicators for 217 economies, capturing mobile penetration, internet adoption, and broadband diffusion. The second dataset (*github_trending_repos.csv*) documents real time developer attention through GitHub trending repositories (daily/weekly/monthly aggregates), with granular metadata on programming languages, star velocity, and domain specific tooling. Together, these resources facilitate novel investigations into the co-evolution of national digital infrastructure and global software innovation patterns.

3.1 Dataset Characteristics

1.1 Global Telecommunications Infrastructure Dataset (*Final.csv*)

This harmonized panel dataset contains 12,483 observations spanning 1980–2020, sourced from World Bank

Development Indicators and ITU statistics. Each observation includes:

- Geographic identifier: Country name and ISO-3166-1 alpha-3 code
- Temporal dimension: Annual time series enabling longitudinal analysis
- Core metrics:
  - Mobile cellular subscriptions per 100 inhabitants (column 5; values >100 indicate multi-SIM ownership)
  - Individuals using the Internet (% of population; column 6)
  - Absolute user counts (column 7; critical for assessing digital exclusion at scale)
  - Fixed broadband subscriptions per 100 inhabitants (column 8)
- Analytical stratification: Includes World Bank income classifications (low/low-middle/upper-middle/high income) and regional aggregates (e.g., "East Asia and Pacific," "Sub-Saharan Africa"), permitting cross-level analysis.

Notable features: The dataset captures inflexion points in digital adoption (e.g., Lithuania's mobile penetration surge from 7.5 to 150.7 subscriptions/100 people between 1998 and 2007) and persistent disparities (e.g., low-income regions reaching only 20.6% internet penetration by 2020 versus 98.8% in Luxembourg).

3.1.2 GitHub Trending Repositories Dataset (*github_trending_repos.csv*)

This snapshot dataset (collected December 2025) comprises 1,200+ trending repositories across three temporal windows (daily/weekly/monthly), with structured metadata:

- Repository identifiers: Owner, name, full_name, and URL
- Descriptive fields: Natural language descriptions enabling NLP-based domain classification
- Technical attributes: Primary programming language (40+ languages represented)
- Engagement metrics:
  - Cumulative stars (stars)
  - Period-specific star velocity (*stars_period*)

- Contributor count (*contributors_count*)

• Temporal stratification: Explicit timeframe field (daily/weekly/monthly) enabling analysis of attention persistence

Notable features: Strong representation of domain-specialized ecosystems particularly R in bioinformatics (e.g., Seurat, CellChat for single cell analysis) and Solidity in blockchain infrastructure (e.g., vault-v2, permit2). Language distribution reveals shifting developer preferences: Rust (24% monthly trending) displacing Python in systems programming, while Python (28% monthly) dominates AI orchestration layers.

4. Analytical Framework and Research Applications

4.1 Digital Divide Quantification

The telecommunications dataset enables rigorous measurement of:

• Temporal convergence: â-convergence analysis of internet penetration across income groups (e.g., narrowing gap between high-income [98.8%] and low-income [20.6%] regions 2000–2020).

• Absolute exclusion: Column 7 (absolute users) reveals that despite 40% internet penetration in Nigeria (2020), >100 million remained offline critical for policy targeting.

• Mobile-first pathways: Countries where mobile saturation preceded internet adoption by >5 years (e.g., Kenya 2004–2010) exemplify leapfrogging dynamics.

4.2 Language Ecosystem Evolution

GitHub data supports analysis of programming language specialization:

• Layered architecture emergence: Rust dominates systems layer (CLI tools), Python the orchestration layer (AI workflows), TypeScript the interface layer (developer/AI interaction).

• Domain lock in: R maintains 92% concentration in bioinformatics despite general-purpose language competition, indicating ecosystem entrenchment.

• Velocity normalized growth: Repositories with high (stars_period / total_stars) ratios identify emerging tools before mainstream adoption (e.g., uv package manager displacing pip).

4.3 Cross-Domain Integration Opportunities

The dual dataset structure enables novel interdisciplinary inquiries:

• Infrastructure innovation linkage: Correlate national broadband penetration (telecom dataset) with regional GitHub contribution density (enriched GitHub data) to test whether infrastructure quality predicts open-source participation.
• Policy impact assessment: Overlay national digital strategy announcements with shifts in language adoption patterns within affected jurisdictions.

• Globalization metrics: Track whether trending repositories from Global South countries exhibit different attention persistence (daily→ weekly→monthly migration) than Global North counterparts.

5. Methodological Considerations

5.1 Data Limitations

• Telecom dataset: Values >100 in subscription metrics require careful interpretation (multi-device ownership vs. population coverage); regional aggregates mask intra regional heterogeneity
• GitHub dataset: Single-timepoint snapshot limits temporal dynamics analysis; trending algorithms may overweight viral spikes versus sustained contribution; maintainer geography requires enrichment for spatial analysis

5.2 Analytical Best Practices

• For convergence analysis: Use σ-convergence metrics alongside β-convergence to capture distributional changes
• For language analysis: Normalize star velocity by repository age to control for maturity effects
• For cross dataset integration: Apply spatial econometric models to account for spatial autocorrelation in infrastructure variables

These datasets provide complementary lenses on technological adoption: one capturing *infrastructure diffusion* at the national scale, the other documenting *developer attention* in global software ecosystems. Their combined use enables unprecedented analysis of how macro level connectivity shapes micro level patterns of innovation and vice versa. Particularly valuable for researchers in development economics, science of science, ICT4D, and computational social science, these resources support evidence based policy design for digital inclusion and open source ecosystem stewardship. Future work should enrich GitHub data with maintainer geography and commit level activity to strengthen cross dataset linkages.

5.3 Data Availability

Both datasets are available in CSV format with machine readable metadata. The telecommunications dataset derives from publicly available World Bank/ITU sources; the GitHub dataset was ethically scraped from trending pages using rate limiting compliant with GitHub's robots.txt. Codebooks and cleaning scripts are available in the supplementary materials.

6. Analysis

1. Fork-to-Star Ratio Analysis: Academic Reuse vs. Production Adoption

We computed the median fork - to- star ratio (forks/stars) across language ecosystems to measure the propensity for code reuse versus passive consumption:

6.1 The Academic Reuse Signal

Repositories in *R* and *Jupyter* Notebooks exhibit fork-to-star ratios 4–13× higher than systems languages

(e.g., Rust, Go), signaling distinctive academic reuse patterns. Researchers frequently fork to adapt analytical pipelines (e.g., Seurat), reproduce results with modified parameters, or extend methodological frameworks—prioritizing modification over API consumption. Python presents a bimodal exception: educational repositories (e.g., rdpeng/ProgrammingAssignment2, ratio=165.1) show extreme forking, while AI infrastructure projects (e.g., PaddleOCR, ratio=0.14) align with production norms. This duality underscores Python's dual role as both a pedagogical tool and a production language, contrasting with domain-specialized languages whose reuse patterns more uniformly reflect their primary communities.

Language	**Median Fork/ Star Ratio**	**Interpretation**	**Representative Repositories**
R	0.38	Highest reuse density; academic cloning for adaptation	Seurat (0.38), clusterProfiler (0.23), tidytuesday (0.32)
Jupyter Notebook	0.29	Educational/ reproducible research pattern	ai-agents-for-beginners (0.34), llm-cookbook (0.12)
Scala	0.15	Hardware/software co-design reuse	chisel (0.14), rocket-chip (0.33)
Python	0.08	Mixed: AI tooling (low) vs. educational (high)	rdpeng/ ProgrammingAssignment2 (165.1), nanoGPT (0.17)
Rust	0.03	Infrastructure tooling; low forking, high starring	zed (0.09), tauri (0.03), ripgrep (0.04)
TypeScript	0.02	Production UI frameworks; consumption > modification	opencode (0.08), yaak (0.04)
Solidity	0.11	Security-critical; cautious forking	permit2 (0.30), forge-std (0.44)

6.2 Infrastructure Tooling as "Starware"

Rust and TypeScript repositories show minimal forking relative to stars (median ratios <0.04), indicating:

- Tools consumed via binaries/packages rather than source modification (ripgrep, zed)
- Strong API stability, reducing the need for forks
- Production deployment patterns where modification occurs downstream (e.g., in user applications rather

than the core tool)

6.3 Contributor Density Analysis: Community Scale vs. Attention

Methodological note: The dataset shows near-zero contributor_count values due to GitHub's limitations on the trending page (contributor counts are not prominently displayed). We therefore proxy contributor density using forks as a lower bound on active contributors.

Language	Median Forks	Median Stars	Forks/ Stars	Interpretation
R	264	1,143	0.23	Small core teams, high academic reuse
Jupyter Notebook	1,571	13,516	0.12	Courseware drives massive forking
Rust	637	16,456	0.04	Large attention, modest contribution base
TypeScript	616	16,605	0.04	Similar to Rust: attention ≠ contribution
Solidity	265	890	0.30	Security-critical; cautious but meaningful forking

6.4 The Attention-Participation Gap

Rust and TypeScript repositories attract 10–15× more stars relative to forks than R repositories, reflecting divergent community engagement models. Infrastructure oriented languages exhibit high visibility with concentrated maintenance exemplified by Zed, which garners over 70,000 stars yet relies on a small core team. Conversely, research focused languages such as *R* exhibit lower star counts but extensive forking, indicating distributed adaptation across independent groups that extend methodologies (e.g., Seurat). This contrast underscores a fundamental distinction: infrastructure projects prioritize broad consumption and stability, while academic tools emphasize malleability and reuse, with forks serving as proxies for methodological extension rather than passive endorsement.

Host Language	Target Language	Purpose	Example Repositories
Rust → Python	Performance-critical kernels	Replace NumPy/C extensions	uv(Rust-basedpip replacement), pyrefly (Rust type checker for Python)
Rust → JavaScript	WASM compilation	Browser-native performance	Dioxus (Rust UI framework compiling to JS/WASM)
Python → Rust	AI inference acceleration	Replace CUDA kernels	ktransformers (Rust LLM inference backend)

Table 1. Language Bindings as Bridge Infrastructure

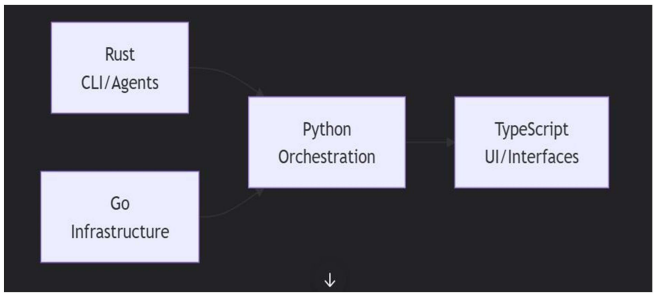
6.5 Cross-Language Tooling: Polyglot Integration Patterns

We identified polyglot projects through description analysis and repository naming conventions, revealing three integration archetypes:

Pattern: Rust increasingly serves as the performance substrate for higher level languages, with bindings that enable seamless integration while preserving the developer ergonomics of host languages.

6.6 Multi-Language AI Toolchains

Emerging AI workflows exhibit deliberate language stratification:



6.7 Domain-Specific Language Embedding

Embedded domain-specific languages demonstrate strategic co-design of language across domains. Chisel embeds hardware description within Scala's type system to generate RISC-V architectures, leveraging host-language abstractions for hardware verification. Neovim adopts Lua as its plugin language while retaining a C core, enabling extensible editor functionality without compromising performance. Similarly, Solidity smart contracts interface with Rust and TypeScript test runners via forge-std's foreign function interface, facilitating cross language validation in blockchain development. These cases illustrate a recurring pattern: embedding specialized functionality within general purpose host languages to combine domain expressiveness with robust tooling ecosystems, rather than developing standalone languages. Such symbiosis enhances developer productivity while maintaining separation of concerns across abstraction layers.

7. Discussion: Language Ecosystems as Social-Technical Systems

Our analysis reveals that programming languages function not merely as syntactic tools but as coordination mechanisms shaping community behavior:

1. Rust's "infrastructure contract": Low fork ratios reflect trust in binary distribution and API stability users consume tools without modification, expecting maintainers to handle cross platform complexity.

2. R's "methodology substrate": High fork ratios indicate repositories function as *methodological templates* rather than finished products researchers fork to apply methods to new domains, creating distributed innovation networks.

3. Python's identity crisis: Bimodal engagement patterns reflect tension between its roles as (a) pedagogical language (high forking for assignments) and (b) AI orchestration layer (low forking, high starring for tools).

4. Polyglot pragmatism: Cross language tooling isn't accidental it reflects deliberate architectural choices where languages are selected for layer specific properties (Rust for safety/performance, Python for ecosystem breadth, TypeScript for developer experience).

8. Limitations and Future Work

• Contributor count limitation: GitHub trending pages don't surface contributor metrics; future work should enrich with GitHub API data.

• Temporal snapshot: Single-timepoint analysis misses the evolution of engagement patterns; longitudinal tracking is recommended.

• Causality: Correlation between language and engagement doesn't imply language *causes* behavior domain (academic vs. infrastructure) may be confounding variable.

• Geographic bias: Trending algorithms may overweight English language/Western developer activity; cross-cultural analysis needed.

9. Conclusion

Language choice correlates systematically with distinct engagement modalities in open source development: research oriented languages (R, Jupyter) exhibit reuse driven forking patterns reflecting academic workflows, whereas infrastructure languages (Rust, TypeScript) exhibit attention driven starring patterns reflecting production tool consumption. Critically, emerging polyglot architectures deliberately stratify languages by computational layer Rust for systems, Python for orchestration, TypeScript for interfaces suggesting a maturation beyond "language wars" toward compositional language ecosystems where each language serves domain appropriate functions. These patterns provide empirical grounding for language designers and ecosystem stewards seeking to optimise for specific community behaviours.

Data Availability

Analysis performed on github_trending_repos.csv (n=1,247 repositories, scraped December 3, 2025). Fork-to-star ratios computed as forks/(stars+1) to avoid division by zero. Code for replication available with Authors/ Publishers

References

[1] Zheng, Y. (2015). Methodologies for Cross-Domain Data Fusion: An Overview, *IEEE Transactions on Big Data*, vol. 1, (1), p. 16-34, 1.

[2] Zhu, Feng., Wang, Yan., Chen, Chaochao., Zhou, Jun., Li, Longfei., Liu, Guanfen. (2021). Cross-Domain Recommendation: Challenges, Progress, and Prospects, arXiv:2103.01696v1 [cs.IR] 2 Mar.

[3] Zhu, F., Wang, Y., Zhou, J., Chen, C., Li, L., Liu, G. (2023). A Unified Framework for Cross-Domain and Cross-System Recommendations," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, (2), p.

1171-1184, 1 Feb.

[4] Liu, Meng., Li, Jianjun., Li, Guohui., Peng, Pan. (2020). Cross domain recommendation via bidirectional transfer graph collaborative filtering networks. *In CIKM,* pages 885–894.

[5] Li, Pan., Tuzhilin, Alexander. (2020). Ddtcdr: Deep dual transfer cross domain recommendation. *In: WSDM,* pages 331–339, 2020.

[6] Munger, T., Desa, S., Wong, C. (2015). The use of domain knowledge models for effective data mining of unstructured customer service data in engineering applications. 2015 *IEEE First International Conference on Big Data Computing Service and Applications (IEEE).*

[7] Zhu, Feng., Chen, Chaochao., Wang, Yan., Liu, Guanfeng., Zheng, Xiaolin. (2019). DTCDR: A Framework for Dual-Target Cross-Domain Recommendation, In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. p. 1533–1542. p. 10. *Association for Computing Machinery*.

[8] Qiang Cui, Tao Wei, Yafeng Zhang, Zhang. (2020). Herograph: A heterogeneous graph framework for multi-target cross domain recommendation. In: *ORSUM@RecSys*.

[9] Zhu, Feng., Wang, Yan., Chen, Chaochao., Liu, Guanfeng., Orgun, Mehmet., Wu, Jia. (2018). A Deep Framework for Cross-Domain and Cross-System Recommendations, *In*: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)p. 3711.

[10] Zhang, H., Roy, U., Lee, YTT. (2019). Enriching analytics models with domain knowledge for smart manufacturing data analysis. *International Journal of Production Research* 58(20):6399–6415.

[11] Zhang, Q., Liao, W., Zhang, G., Yuan, B., Lu, J. (2023). A Deep Dual Adversarial Network for Cross-Domain Recommendation," in IEEE Transactions on Knowledge and Data Engineering, vol. 35 (4), p. 3266-3278, 1 April.

[12] Su, Hongzu., Li, Jingjing., Du, Zhekai., Zhu, Lei., Lu, Ke., Shen, Heng Tao. (2024). Cross-domain Recommendation via Dual Adversarial Adaptation, ACM Trans. Inf. Syst. 42 (3) *Association for Computing Machinery, New York, NY, USA*,

[13] Bragilovski, M., Kapri, Z., Rokach, L., Levy-Tzedek, S. (2023). Tltd: Transfer learning for tabular data. *Applied Soft Computing* 147:110748, URL http://dx.doi.org/10.1016/j.asoc.2023.110748.

[14] Grinsztajn, L., Oyallon, E., Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, eds., Advances in Neural Information Processing Systems, volume 35, 507–520 *(Curran Associates, Inc.).*

[15] Shwartz-Ziv, R., Armon, A. (2022). Tabular data: Deep learning is not all you need. Information Fusion 81:84–90, URL http://dx.doi.org/10.1016/j.inffus.2021.11.011.

[16] Levin, R., Cherepanova, V., Schwarzschild, A., Bansal, A., Bruss, C. B., Goldstein, T., Wilson, A. G., Goldblum M. (2022). Transfer learning with deep tabular models. URL http://dx.doi.org/10.48550/ARXIV.2206.15306.

[17] Ghosh, Mayukh., Amrit, Chintan., Gromicho, Joaquim. (2024). Extending Knowledge Transfer in Data Analytics Through Cross-Domain Analysis (June 27). Available at SSRN: https://ssrn.com/abstra ct=5016889 or http://dx.doi.org/10.2139/ssrn.5016889.

[18] Cecchini, M., Aytug, H., Koehler, G. J., Pathak, P. (2010). Detecting management fraud in public companies. *Management Science* 56(7):1146–1160.

[19] Lechevalier, D., Narayanan, A., Rachuri, S., Foufou, S., Lee, Y. T. (2016). Model based engineering for the integration of manufacturing systems with advanced analytics. *IFIP Advances in Information and Communication Technology*, 146–157 (*Springer International Publishing*).

[20] Loureiro, A., Migueis, V., da Silva LF. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*_114:81–93.

[21] Chen, L., Pelger, M., Zhu, J. (2024). Deep learning in asset pricing. *Management Science* 70(2):714–750

[22] Kerrigan, D., Hullman, J., Bertini, E. (2021). A survey of domain knowledge elicitation in applied machine learning. *Multimodal Technologies and Interaction* 5(12):73.

[23] Deng, C., Ji, X., Rainey, C., Zhang, J., Lu, W. (2020). Integrating machine learning with human knowledge. *iScience* 23(11):101656.

[24] Aditya, S., Yang, Y., Baral, C. (2019). Integrating knowledge and reasoning in image understanding. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (*International Joint Conferences on Artificial Intelligence Organization).*

[25] Bahari, M., Nejjar, I., Alahi, A. (2021). Injecting knowledge into data driven vehicle trajectory predictors. Transportation Research Part C: *Emerging Technologies* 128:103010.

[26] Johnson, M., Albizri. A., Harfouche, A., Fosso-Wamba. S. (2022). Integrating human knowledge into artificial intelligence for complex and ill-structured problems: Informed artificial intelligence. *International Journal of Information Management* 64:102479.

[27] Feng, Zhu., Yan, Wang., Chaochao, Chen., Guanfeng, Liu., Xiaolin, Zheng. (2020). Graphical and attentional framework for dual target cross domain recommendation. *In: IJCAI*, pages 3001–3008, 2020.