Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project

Project Plan

Unlocking Global Capital: AI-Powered Prediction, Access, and Verification of Investor Decision Makers

by

Daniel Bowman

Email: daniel.bowman24@imperial.ac.uk

GitHub username: acse-db1724

Repository: ese-ada-lovelace-2024/irp-db1724

Supervisors:

Antony Sommerfeld

Dr. Yves Plancherel

June 2024

Table of Contents

List of Tables

Abstract

Automating investor contact maintenance is critical for scalable, reliable capital raising in a sector plagued by high turnover and data decay. Existing commercial services like Hunter and ZoomInfo rely on paid lookups and often take days to return results. In contrast, our novel pipeline seamlessly integrates three components—a comprehensive offline template miner that uncovers every common formatting skeleton, a lightweight real-time classifier that instantly predicts the correct template for any new name–domain pair, and on-the-fly third-party deliverability scoring—into one end-to-end system. Performance will be evaluated on held-out contacts, reporting template coverage, prediction accuracy, API precision/recall, and sub-50 ms query latency. This hybrid approach not only outperforms standalone pattern-mining or machine-learning methods but also undercuts costly data-provider fees, democratizing access to fresh, validated investor email information.

1. Introduction

1.1. Problem Description

Despite AIP's rich dataset maintaining up to date email addresses remains a problem due to investor contact data is notoriously brittle. Individuals in this profession change roles frequently and roughly 20% of decision makers retire or change firms each year. On top of this, monopolistic data access prohibits small firms from competing – for example leading commercial providers (Hunter, ZoomInfo) charge a fee (sometimes up to £120 a month) for sub second response time.

Predicting email templates across a wide variety of corporate domains poses several challenges. First, companies use numerous different skeleton templates–with a few formats covering most addresses and rare ones demanding comprehensive discovery and robust fallbacks [1] [2]. Finally, maintaining low latency under live load requires a prediction model that is extremely lightweight and highly accurate.

This project seeks to overcome this by proposing an AI-Driven solution to discover the finite set of email template skeletons used across the database and then predict the appropriate template for any new name-company pair. Finally, the pipeline will attach a confidence score via third-party verification APIs. The plan will also include scope for stretch goals to link predicted emails to LinkedIn profiles and other data that enrich the outputs.

1.2. Significance

A successful system will reduce the costs associated with maintaining valid addresses, minimize bounce rates, and enable smaller firms to compete on a level playing field.

From an academic and technical perspective, this work contributes a novel hybrid approach that unites exhaustive, order-aware pattern mining with high-speed, supervised prediction and real-time verification. Whereas previous studies have focused either on offline pattern discovery or on standalone ML classifiers, the integrated pipeline both uncovers the underlying syntactic templates and applies them in a live inference setting augmented by third-party

deliverability scoring. This will advance the state of the art in automated contact inference and offer a blueprint for scalable, commercial tools in data-driven outreach.

a) Scope and Aims

This project harnesses AIP's proprietary database of 300000 individual contacts and 150000 investment groups to infer and verify corporate email formats. The project will begin by mining addresses offline to derive a compact set of email-templates, then train a supervised model to predict the correct template for any new name–domain pair in real time. Predicted addresses will be enriched with confidence scores from third-party APIs, measured template coverage, prediction accuracy, and end-to-end response time. Automated outreach lie outside the scope of this work. Data privacy such as GDPR could pose a potential risk to this project but has been deemed out of scope in terms of this research.

The project aims to identify a minimal set of templates that together explain the vast majority of AIP's known email addresses and to develop a prediction model that achieves high precision and recall on unseen name–domain inputs, flagging low-confidence cases. Finally, the project will deliver a user-friendly interface that enables real-time lookup of predicted email addresses along with their associated confidence levels, demonstrating the feasibility of a deployable prototype.

2. Review of Existing Work

2.1. Pattern-Mining Methods

Early work in association-rule mining [1] demonstrated how to enumerate frequent co-occurring items via candidate generation and pruning, but its repeated scans impose heavy I/O. FP-Growth [3] addresses this by building a compact prefix-tree in two passes and recursively mining conditional sub-trees, reducing scans for repetitive patterns like email templates. Sequential miners such as TRuleGrowth [4] explicitly capture token order but at the expense of increased memory overhead.

A CNN+LSTM [5] with discrete wavelet preprocessing and a BiLSTM [6] with multi-scale attention have also been implemented with reasonable results. Ensemble methods [7] combine multiple architectures to stabilize training on noisy inputs. Although these DL methods excel at modelling noisy patterns, their computational-complexity outweigh their benefits for the strictly structured, high-precision task of email-template mining.

Large language models [8] can learn common email-template structures through name-company co-occurrence statistics with around 3% accuracy in under 50ms. While recall is low, it highlights a potential lightweight fallback for rare templates that are otherwise missed.

In practice, TRuleGrowth strikes the ideal balance: it preserves ordering, achieves high recall, and its higher runtime is acceptable because mining runs offline, producing a comprehensive set of skeleton templates to drive prediction.

2.2. Email-Template Prediction

Character-level models [9] demonstrate that 1D convolutions over one-hot character sequences yield robust classification of templates with minimal preprocessing and fast inference. BiLSTM-attention architectures [6] further capture long-range and local dependencies, learning candidate subsequence via multi-scale heads. Semantically conditioned LSTMs [10] and neural HSMMs [11] borrow from NLG in that they treat template placeholders as “slots” in control-vectors, ensuring each placeholder appears once and allows interpretable generation.

LightGBM [12] and XGBoost [13] offer sub-10 ms inference on sparse n-gram and categorical features, benefiting from histogram binning and sparsity-aware splits. CatBoost’s ordered boosting [14] handles high-cardinality domains without target leakage, while linear SVMs scale to thousands of character n-grams with automatic margin-based tuning [15].

Comparative surveys [16] have evaluated numerous regression models (including tree-based and ensemble-style learners) found that CatBoost was superior in performance, with LightGBM trailing just behind.

Given strict latency and interpretability requirements, a gradient-boosted tree model (LightGBM or CatBoost) over parser-extracted features strikes the best balance of speed, accuracy, and robustness for real-time prediction.

2.3. Gap Analysis

While prior work has explored individual components, no published system integrates them into a single, end-to-end pipeline. The project fills this gap by combining an exhaustive TRuleGrowth miner with a lightweight, high-throughput classifier and integrated verification into a unified workflow.

3. Methodology

3.1. Data Collection and Preprocessing

Raw names and emails are tokenized, case-normalized, trimmed, regex-validated, deduplicated, and have missing fields imputed via nearest-neighbour lookup in a company-alias embedding, with extreme outliers flagged for manual review.

3.2. Model Development

Offline pattern mining employs TRuleGrowth for its exhaustive, order-aware rule growth, ensuring recovery of even low-support sequential templates in a single batch pass despite AIP's large, noisy dataset and long tail of rare formats. The same method is applied to mine company-name to domain mappings, with derived skeletons capturing dominant transformations; any gaps in coverage will defer to deep-learning–based fallbacks.

Live inference uses engineered structural features to train a LightGBM classifier on sparse inputs. Its histogram-based binning, gradient-based sampling, and native handling of high-cardinality features deliver sub-10 ms scoring over the database with resilience to noisy data, and a linear SVM model automatically activates whenever confidence falls below a preset threshold.

3.3. Verification Integration and Fallbacks

Predicted addresses are sent through a pooled, rate-limited client to a third-party verification API (responses cached for 24 h); on API failure or quota

exhaustion, local regex and MX/A-record checks assign a confidence score, and any unverified or low-confidence results are flagged for manual review.

3.4. User Interface

The pipeline will be available as a Dockerized microservice offering a REST JSON endpoint and as an embeddable SDK/library—both loading the same serialized templates, models, and lookup tables at startup to ensure consistency, automatic scaling, and low-latency in-process calls.

4. Project Plan

Phase	Dates	Work to Do	Deliverables	Milestone
Initial Research and Planning	May 26 – June 13	Collect relevant literature. Document state of the art in these aspects and path forward. Draft Plan.	Final project-plan PDF.	Review draft plan with supervisor.
Data Cleaning and Exploration	June 16 – June 20	Clean and normalise name, email, firm fields. Build company domain fuzzy lookup table. Exploratory data analysis	Cleaned dataset, lookup table, EDA notebook and summary plots.	Show EDA findings.

		of format per domain.		
Offline Pattern Mining	June 23 – July 4	Build TRuleGrowth Model for both email templates and firm name to email domain transformation. Prune, merge, augment skeletons. Tune parameters.	Skeleton list, coverage report.	Review template coverage.
Email Skeleton Prediction Model	July 7 – July 25	Feature Engineering. Train and tune LightGBM (and a SVM fall back) with 5 fold cross-validation. Evaluate on held out contacts.	Model notebook, evaluation metrics, summary slides.	Review initial model performance.

Verification Integration	July 28 – Aug 8	Integrate third party API calls to score predicted addresses. Run large scale evaluation and analyse failures. Implement caching and fallbacks.	Combined verification report, failure analysis.	Demo verification workflow.
Stretch Goals	Aug 11 – Aug 15	Link results to LinkedIn profiles to validate predicted identities. Use AIP services to retrieve additional contact details. Enrich outputs with surplus individual information.	Prototype linking demo, enriched contact dataset.	Stakeholder check-in.

Packaging and Demo	Aug 18 – Aug 22	Expose model through python or C++ wrapper. Create API and demo.	Docker image, SDK/library, demo app and usage documentation	Internal demo review
Final Report	Aug 25 – Aug 29	Write final report. Populate figures and tables.	Final report PDF, reproducible pipeline script.	Submit final report.

Table 1: Project Plan

5. Validation

Twenty percent of contacts—stratified by domain size—are held out before any mining or modelling. For offline mining, template coverage and support-weighted recall on held-out emails will be reported. Live prediction performance will be measured via precision, recall and F1-score overall and by domain segment. Verification accuracy will be assessed through API precision and recall on a manually labelled subset. End-to-end latency (median and 95th percentile) under realistic concurrent load will be recorded to confirm sub-50 ms performance, with any bottlenecks flagged for further optimization. Performance is expected to be near-perfect for common domains and lower for rare ones; fallback mechanisms and monitoring of low-confidence invocations will guide the addition of new template rules where they are most needed.

6. Risks, Challenges, and Future Improvements

Key risks include inconsistent or malformed contact data, ambiguous company name to domain mappings and third-party API rate limits. Data noise is addressed through regex validation, outlier logging and embedding-based alias lookups. Domain disambiguation relies on TRuleGrowth-mined skeletons with a deep-learning fallback for low-confidence cases. API failures are mitigated through pooled, cached calls with local regex and MX record validation. Template coverage and latency metrics are monitored weekly, support thresholds adjusted as needed, and unresolved edge cases escalated to advanced modelling or manual review. Future work may generalise the framework to other structured communications, incorporate active learning on low-confidence predictions and explore privacy-preserving federated training.

7. References

[1] R. Agrawal, R. Imielinski and A. Swami, “Mining association rules between sets of items in large databases,” in *ACM SIGMOD*, Washington, DC, 1993.

[2] V. Pavilova, “scrupp.com,” Scrupp, 13 May 2025. [Online]. Available: https://scrupp.com/blog/email-pattern#:~:text=Email%20pattern%20finder%20tools%20automate%20the%20process%20of,that%20can%20help%20you%20find%20email%20patterns%20efficiently.. [Accessed 9 June 2025].

[3] J. Han, J. Pei and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” *SIGMOD Record (ACM Special Interest Group on Management of Data),* vol. 29, no. 2, p. 1–12, January 2000.

[4] A. Abdelwahab and N. Youssef, “Performance Evaluation of Sequential Rule Mining Algorithms,” *Applied Sciences,* vol. 12, no. 10, p. 5230, 2022.

[5] A. Jamshed, B. Mallick and P. Kumar, “Deep learning–based sequential pattern mining for progressive database,” *Soft Computing,* vol. 24, no. 22, p. 17233–17246, 2020.

[6] T. Yang, Y. Cheng, Y. Ren, Y. Lou, M. Wei and H. Xin, “A Deep Learning Framework for Sequence Mining with Bidirectional LSTM and Multi-Scale Attention,” arXiv, 2025.

[7] C. Djellali and M. Adda, “A New Deep Learning Model for Sequential Pattern Mining Using Ensemble Learning and Model Selection: Taking Mobile Activity Recognition as a Case,” in *16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)*, Halifax, Canada, 2019.

[8] H. Shao, J. Huang, S. Zheng and K. Chang, “Quantifying Association Capabilities of Large Language Models and Its Implications on Privacy

Leakage," in *Findings of the Association for Computational Linguistics: EACL 2024*, St. Julian's, Malta, 2024.

[9] X. Zhang, J. Zhao and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Advances in Neural Information Processing Systems*, vol. 28, Montreal, Canada, Curran Associates, Inc., 2015, p. 649–657.

[10] T. Wen, M. Gašić, N. Mrkšić, P. Su, D. Vandyke and S. Young, "Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[11] S. Wiseman, S. M. Shieber and A. M. Rush, "Learning Neural Templates for Text Generation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, 2017.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016.

[14] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems*, 2018.

[15] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *Proceedings of the 10th European Conference on Machine Learning (ECML)*, Chemnitz, Germany, 1998.

[16] D. V. d. O. Santos and W. C. Brandão, "Learning to Predict Email Open Rates Using Subject and Sender," in *Proceedings of the 20th International*

Conference on Web Information Systems and Technologies (WEBIST 2024), 2024.