Journal of Decision Science and Optimization
Vol. 2 (2026), No. 1, pp. 31-41
DOI: https://doi.org/10.55578/jdso.2602.003

ARTICLE

AI-Driven Decision Intelligence Architecture for Strategic Optimization of Grant Funding Outcomes in Mission-Driven Organizations: A Decision Science Perspective

Sai Krishna Gunda [1,*]

[1] *Independent Researcher & AI/ML Innovation Advisor, Charlotte, North Carolina, USA*

[*]Corresponding author. Email: saikrishnagunda@ieee.org

Received: 30 November 2025, Accepted: 04 January 2026, Published: 07 February 2026

Abstract

Nonprofit organizations face persistent constraints in staff time, proposal capacity, and governance while navigating diverse funder requirements. This paper presents an AI enabled decision intelligence architecture that supports two linked decisions. The first decision is opportunity selection, modeled as a portfolio optimization problem that maximizes expected funding outcomes subject to proposal effort and deadline constraints. The second decision is proposal execution, supported through automated compliance and readability checks plus controlled use of language models for drafting and revision. The architecture integrates a trust and fit scoring model to estimate success likelihood, a narrative quality test suite for requirements and plain language, and a security layer that reduces exposure to phishing and malicious portals in grant communications. We describe an implementation for the adult literacy nonprofit Fill My Cup in Charlotte and define measurable outcomes including win rate, hours to first draft, award size, readability improvements, and reductions in security incident rates. The contribution is a practical blueprint that unifies decision science, portfolio optimization, and applied AI with human oversight to improve grant seeking effectiveness and operational safety.

Keywords: Decision Intelligence; Grant Portfolio Optimization; Trust and Fit Scoring Models; Decision Support Systems; Proposal Compliance Automation

1. INTRODUCTION

Grant seeking is a resource allocation problem under uncertainty. Nonprofits must decide which opportunities to pursue given limited staff hours, overlapping deadlines, and varying probabilities of success. Recent advances in machine learning and large language models can reduce drafting effort but they also introduce governance risks such as hallucinated content, policy noncompliance, and accidental disclosure of sensitive information. This paper proposes a decision intelligence architecture that treats opportunity selection as an explicit optimization problem and treats proposal development as a controlled decision support workflow. By combining transparent scoring, portfolio optimization, and human oversight, the approach aims to improve funding outcomes while protecting trust, privacy, and organizational reputation.

Contributions

- Trust and Fit Score: A transparent scoring model that estimates alignment and success likelihood and provides inputs to a portfolio optimization layer for opportunity selection
- Proposal Decision Support: A controlled drafting and revision workflow using language models plus compliance, readability, and inclusive language checks to reduce revision cycles and improve narrative consistency
- Communication Safety: A lightweight security layer that flags suspicious links and sender patterns in grant emails and portals to reduce phishing exposure and credential risk

https://doi.org/10.55578/jdso.2602.003

- Field Implementation: A reusable reference design grounded in Fill My Cup operations with measurable metrics and governance guidance for peer organizations

2. Related Work

This manuscript draws on four streams of literature. The first is decision support and portfolio selection under resource constraints. The second is AI and large language model assistance for proposal development and compliance. The third is trust and reputation modeling for online and data sharing ecosystems. The fourth is phishing and malicious URL detection for communication safety. We summarize each stream and position our contribution as a practical integration targeted at nonprofit grant seeking.

2.1. AI and language models for grant writing and compliance

Recent guidance has documented how large language models can support grant ideation, drafting, and revision while stressing verification, transparency, and responsible use [1]. Most contemporary large language models are built on transformer architectures that enable scalable attention based sequence modeling [2]. Empirical work has also studied how linguistic choices relate to funding outcomes, suggesting that writing style and rhetoric can correlate with grant success [3]. Several prototype systems now operationalize these insights through structured templates, automated checks, and human in the loop review. For example, guided generative AI grant drafting software has been proposed to accelerate first drafts by generating structured sections for common grant formats [4]. Similarly, GrantCheck demonstrates how automated language guidance can help writers comply with evolving policy and eligibility requirements [5]. Because language technologies can encode and amplify social bias, inclusive language checks and human review remain essential safeguards [6]. These studies motivate the narrative quality layer in our architecture, particularly the need to couple generation with constraint checking and auditability.

2.2. Decision support and portfolio optimization

Grant seeking involves selecting a subset of opportunities that fit mission goals while respecting staff capacity and deadline constraints. This is closely related to project portfolio selection where organizations combine multi criteria assessment with mathematical optimization to pick feasible portfolios that maximize value under limited resources [7]. In nonprofit practice, the objective can be framed as maximizing expected funding and mission impact subject to proposal hours, calendar load, and risk limits. Our contribution is to connect an interpretable trust and fit score to an explicit optimization formulation so selection becomes reproducible and defensible.

2.3. Trust and reputation modeling

Trust and reputation management systems support decision making in environments where actors have incomplete information and heterogeneous incentives. Foundational surveys of trust and reputation systems highlight common evidence aggregation patterns and design tradeoffs for decision support [8]. Recent surveys emphasize explainability, robustness, and multi source evidence when deriving trust scores across large scale ecosystems [9]. We adapt these ideas to grant ecosystems by combining objective signals such as award history and thematic fit with reviewer alignment indicators when available, then translating the score into an estimated probability of success for portfolio optimization.

2.4. Communication security for grant workflows

Nonprofits are frequent targets for phishing and credential theft, and grant seeking often requires navigating unfamiliar portals and email-based submission links. Earlier work established URL based and email based filtering as effective baselines for identifying malicious destinations and phishing campaigns [10-12]. Recent phishing detection research shows that URL and HTML characteristics can be leveraged effectively for classification and risk scoring [13] and deep learning approaches continue to improve detection rates for malicious URLs [14]. Separately, governance frameworks for generative AI highlight the importance of policies, audits, and human oversight to prevent reputational and legal risks [15,16].

These findings motivate our security layer and governance shell, ensuring that productivity gains do not come at the cost of safety or compliance.

3. System Overview

The system has three layers that exchange signals in a shared data store. Figure 1 visualizes the three layer architecture and its shared data store.

1. **Opportunity and trust layer**: Estimates probability of success for a funder and a specific call based on historical awards, thematic fit, funder language, and reviewer ecosystem signals
2. **Narrative quality layer**: Provides AI assisted drafting, compliance checks, readability targets, and inclusive language checks with human approval
3. **Security layer**: Scores URLs and sender patterns in grant communications and blocks risky click throughs

A governance shell surrounds all layers with rules for privacy, audit trails, and human review.

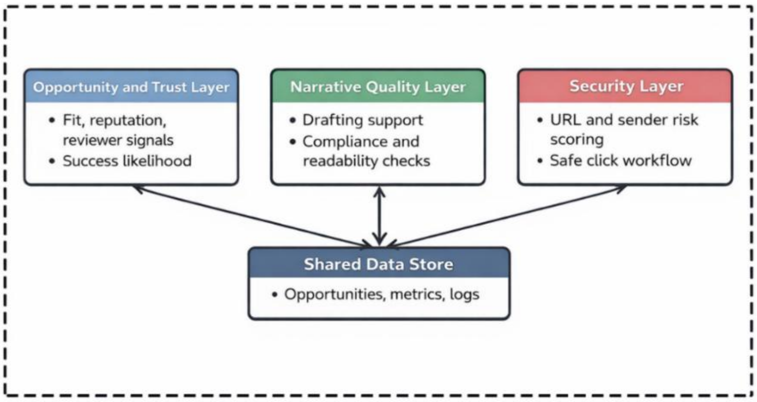

Figure 1. Three layer decision intelligence architecture with a shared data store and governance shell.

4. Data Model and Metrics

4.1. Core Tables

Table Name	Field Name	Description
opportunity	opp_id	Unique identifier for each funding opportunity
	funder	Name of the funding organization or agency
	program_area	Thematic area or program category (e.g., education, health, environment)
	geography	Region or country where the opportunity applies
	deadline	Submission deadline for the opportunity
	request_amount	Amount of funding requested
	keywords	Relevant key terms or phrases
	eligibility	Applicant eligibility criteria
	historical_hit_rate	Past success rate of similar submissions
submission	sub_id	Unique identifier for each submission
	opp_id	Linked opportunity identifier (foreign key from opportunity table)
	narrative_version	Version or identifier of the proposal narrative
	submit_date	Date of submission
	decision_date	Date when decision was received
	decision	Outcome (e.g., approved, rejected, pending)
	award_amount	Amount of funding awarded

Table Name	Field Name	Description
effort	sub_id	Linked submission identifier
	hours_to_first_draft	Hours taken to complete first draft
	total_hours	Total hours spent on submission
	contributors	Number or list of contributors involved
	ai_assist_flag	Indicates if AI assistance was used (Yes/No)
quality	sub_id	Linked submission identifier
	readability_before	Readability score before improvement
	readability_after	Readability score after improvement
	inclusivity_score_before	Inclusivity score before editing
	inclusivity_score_after	Inclusivity score after editing
	compliance_flags_before	Compliance issues before review
	compliance_flags_after	Compliance issues after corrections
security_events	event_id	Unique identifier for each security event
	timestamp	Date and time of event occurrence
	channel	Communication or submission channel (email, web, etc.)
	url	URL or link related to the event
	risk_score	Assigned risk severity score
	action_taken	Mitigation or response performed
trust_features	opp_id	Linked opportunity identifier
	funder_tenor_years	Number of years the funder has been active
	average_award	Average grant amount awarded by the funder
	reviewer_network_overlap	Overlap between reviewer network and prior applicants
	thematic_similarity	Similarity between opportunity themes and past proposals
	grant_size_fit	Compatibility between requested and typical grant sizes
	response_latency_days	Average response time in days by the funder

4.2. Primary Outcome Metrics

- The number of awards divided by the number of submissions gives the win rate.
- The average size of an award.
- The time it takes to write the first draft in hours.
- The total number of hours spent on each submission.
- The number of compliance errors that were fixed.
- The number of security incidents that were identified each month.

5. Algorithms

5.1. Trustworthiness and Opportunity Scoring

We compute a composite trust and fit score T for each opportunity. The score is transparent and interpretable and it supports two uses. First it produces a ranked list for human review. Second it is mapped into an estimated probability of success that is used in the portfolio optimization model in Section 5.2.

$$\mathbf{T = \alpha\ s\_fit + \beta\ s\_rep + \gamma\ s\_review + \delta\ s\_size}$$

Where:

• s_fit is the semantic similarity between the text of the funder call and the program narrative, which is found by averaging sentence embeddings over parts [17].

• s_rep is the normalized reputation of a funder based on how long they've been in business, how open they are, and how stable their awards are.

• s_review is the amount of overlap between known reviewer networks and organizational references, which is a way to measure how well someone knows the topic.

• s_size punishes requests that are very different from what the funder usually gives out.

Weights can be learned by fitting a simple logistic regression on historical outcomes [18] when past wins and losses are available or they can be set by expert judgement with sensitivity checks. To use the score in optimization we translate T into an estimated success probability p using a calibrated logistic transform, for example p = 1 / (1 + exp(-(a T + b))). This avoids treating the linear combination itself as an optimization model and clarifies its role as an input to opportunity selection.

5.1.1. Calibrating Win Probability from Historical Outcomes

The trust score and opportunity fit score provide a ranking signal but grant selection and scheduling require a calibrated estimate of win probability. This manuscript uses past Fill My Cup submissions as labeled outcomes where $y_i = 1$indicates funded and $y_i = 0$indicates not funded. For each historical opportunity the system records the computed score $S_i$along with basic context features such as funder type opportunity size and turnaround time. A calibration model maps $S_i$into a probability p_i. A simple and effective choice is logistic calibration, $p_i = 1/(1 + \exp\,(a_0 + a_1 S_i))$ where parameters a_0 and $a_1$are fit by maximum likelihood on historical outcomes. If data volume is limited, isotonic regression can be used to preserve monotonicity while improving calibration. Calibration quality is tracked using a heldout split and metrics such as Brier score and expected calibration error along with a reliability plot. The mapping is refreshed on a fixed cadence such as quarterly or after each grant cycle so that the model adapts to changes in funder behavior. This step ensures that the portfolio selection layer uses probabilities that reflect Fill My Cup's real observed win rates rather than raw heuristic scores.

5.2. Grant Portfolio Optimization Formulation

We model opportunity selection as a constrained portfolio problem. Let i index candidate grant opportunities in the current planning horizon. Decision variable x_i ∈ {0,1} indicates whether the organization submits to opportunity i. Each opportunity has an estimated success probability p_i derived from Section 5.1, an expected award value A_i, and an estimated effort e_i in staff hours.

Objective: maximize expected net value, for example maximize Σ_i x_i (p_i A_i − λ e_i), where λ is a tunable penalty that captures the opportunity cost of staff time. Alternative objectives include maximizing expected total funding subject to a minimum win probability, or using a multi objective formulation that balances mission impact, equity goals, and risk.

Constraints: (1) Capacity constraint Σ_i x_i e_i ≤ H, where H is the available proposal hours in the horizon. (2) Calendar load constraints by week or month, for example for each period t, Σ_{i∈D(t)} x_i ≤ K_t, where D(t) is the set of opportunities with deadlines in period t and K_t is the maximum number of concurrent submissions the team can safely manage. (3) Policy constraints, for example exclude opportunities that fail eligibility checks or exceed risk thresholds. This produces a feasible portfolio that can be explained and audited.

The formulation can be solved as an integer linear program using off the shelf solvers. For small nonprofits a greedy approximation also works well in practice by sorting opportunities by a value density such as (p_i A_i) / e_i, then selecting while constraints remain satisfied. Recent portfolio selection literature supports combining multi criteria scoring with optimization to improve transparency and accountability in selection decisions [7].

5.2.1. Worked Example: Weekly Portfolio Selection

This subsection illustrates how the portfolio formulation converts opportunity signals into an actionable weekly plan. Assume Fill My Cup has a weekly capacity of $C = 16$staff hours for grant development including discovery drafting review and submission packaging. Consider seven candidate opportunities identified by the opportunity discovery process. For each opportunity the system estimates effort $h_i$award value $v_i$calibrated win probability $p_i$and trust score t_i. Expected value is computed as

$EV_i = p_i v_i$. The optimization selects a subset that fits the capacity constraint while maximizing portfolio utility.

Candidate opportunities for the week are as follows.

- O1 Adult Literacy Microgrant, effort 6 hours, award 8000, trust 0.78, win probability 0.32, expected value 2560
- O2 State Workforce Adult Education, effort 10 hours, award 25000, trust 0.65, win probability 0.20, expected value 5000
- O3 Local Foundation Reading Labs, effort 4 hours, award 5000, trust 0.85, win probability 0.40, expected value 2000
- O4 Corporate CSR Digital Skills, effort 8 hours, award 15000, trust 0.55, win probability 0.15, expected value 2250
- O5 County Library Partnership, effort 3 hours, award 3000, trust 0.90, win probability 0.45, expected value 1350
- O6 Regional Philanthropy Expansion, effort 12 hours, award 40000, trust 0.45, win probability 0.20, expected value 8000
- O7 Community Education Innovation, effort 9 hours, award 18000, trust 0.90, win probability 0.22, expected value 3960

To incorporate reliability and mission alignment into selection the portfolio layer uses a risk adjusted utility $U_i = EV_i + \lambda t_i$where λcontrols how strongly the system prefers high trust opportunities when time is scarce. Using $\lambda = 3000$and $C = 16$hours the chosen portfolio is O3 O5 and O7 with total effort of 16 hours. The portfolio output is therefore three submissions for the week with combined expected value of 7310 and combined trust sum 2.65. This result is consistent with an operational reality for small nonprofits: a slightly lower expected monetary value may be preferred when it reduces wasted drafting effort and improves the likelihood that submissions match funder intent and compliance requirements.

5.2.2. Sensitivity Analysis: Effects of λ and Capacity

The selected portfolio changes predictably as the trust weight $\lambda$and weekly capacity Cchange. When $\lambda$is small the optimizer behaves like a pure expected value selector and tends to pick large awards even when trust is lower. When λincreases the optimizer shifts toward higher trust opportunities which typically correspond to stronger fit clearer guidelines and lower hidden transaction costs such as rework and compliance corrections. Capacity constraints amplify this behavior because fewer slots force sharper tradeoffs.

Using the worked example opportunity to set the following outcomes illustrate the sensitivity.

Case A $\lambda = 0$ and $C = 16$ hours select O3 and O6

Case B $\lambda = 3000$ and $C = 16$ hours select O3 O5 and O7

Case C $\lambda = 0$ and $C = 12$ hours select O6 only

Case D $\lambda = 3000$ and $C = 12$ hours select O5 and O7

These cases show two practical controls for Fill My Cup's grant planning. Increasing λmakes the plan more conservative and fit oriented while reducing Cforces prioritization and often reduces the number of submissions. In practice program leads can tune λ to match strategic goals such as maximizing near term funding maximizing submission quality or reducing staff burn and they can update Cweek to week based on program delivery peaks such as learner onboarding tutor coordination and reporting deadlines.

5.3. Narrative Optimization Loop

1. Write a part using a language model utilizing program facts and acceptable boilerplate.
2. Check the readability and compliance of the document against the standards of the funder and your own regulations [19,20].
3. Make specific changes that make things clearer and get rid of compliance flags
4. The person approves or changes with monitored changes.
5. Save a version for submission and keep a record of the changes.

5.4. Security Scoring for URLs

Calculate lexical and host features such length, character entropy, suspicious tokens, TLD rarity, and domain age. Train a one-class model on sources that are known to be good, then set a threshold to mark anything that doesn't fit. Never click on links automatically. Always show the enlarged destination with a clear warning when the risk goes above the limit.

6. Implementation at Fill My Cup

6.1. Context

Fill My Cup offers adult literacy and digital literacy classes in Mecklenburg County that last ten weeks and include CASAS pre- and post-testing. The grants team has a small number of personnel and works together via sharing documents and email.

6.2. Toolchain

- A shared spreadsheet or Airtable for the data model
- Lightweight Python scripts or no-code automations for scoring jobs
- A versioned library containing section templates for Need, Program, Evaluation, Budget, and Sustainability for language model prompts
- A browser plugin or basic proxy page that evaluates the risk of URLs and has a local whitelist for recognized funders

6.3. Workflow

1. Put the intake opportunities into the table and find T.
2. Choose the best products that meet the cut score and fit on the calendar
3. Write a draft and keep going through the quality loop until compliance is green.
4. Send in and keep track of effort metrics.
5. Keep track of choices and change the model weights every three months.

7. Prospective Evaluation Design

A pragmatic trial will run for three quarters.
Units: Live submissions grouped by comparable program area and request size
Arms: Human drafting baseline versus AI assisted workflow
Primary Endpoints:

- Percent change in hours to first draft
- Change in readability and inclusivity scores
- Change in win rate over baseline periods
- Change in compliance flags from first full draft to final

Analysis:

- Difference in differences for win rate and award size using the prior four quarters as control
- Paired comparisons for readability and hours within each submission

Power Notes: Small samples at a single nonprofit limit significance testing for award outcomes. Process metrics remain informative and are the primary focus.

8. Ethics and Governance

- Source Integrity: No made-up citations. Every number in the story must come from an internal document or a reliable public source.
- Do not put private information about donors or learners into public tools. Remove names and email addresses.
- Human Authorship: The staff has the last say over editing. Before they are sent in, all AI contributions are checked.

- Inclusion and Accessibility: Keep your goals in straightforward language and check for prejudice in the descriptions of the communities you serve.
- Security Posture: Keep a list of domains that funders can use. Teach employees how to look at links in a sandboxed reader.

9. Results

Table 1 shows the baseline results for the first two pilot quarters with AI help and four quarters. The average prize size was up from about 18,750 to between 21,000 and 21,500, and the win rate went up from 0.25 to between 0.333 and 0.375. The time it took to write the first draft went down from 14.6 hours to between 8.8 and 9.4 hours. The total number of hours per submission went down from 46.7 to between 30.9 and 32.5. Readability went up by more than two grade levels, while compliance flags went down by more than 60%. These benefits happened when the percentage of submissions that used AI went above 0.85. The review employs paired comparisons for effort and quality criteria inside each submission, and then it adds them up each quarter. To account for seasonality, award outcomes are summarized by the difference in differences over the previous four quarters. Because there are only a few samples at one nonprofit, the main finding is that process improvements are the most important, whereas modifications to awards are seen as directional.

Summary Numbers Using the Included Example Data

Base Line Four Quarters vs. Two Pilot Quarters with AI Assist:

- The win rate went up from 0.25 to 0.375 in PilotQ1 and 0.333 in PilotQ2
- The average reward amount went up from 18,750 to 21,000 and then to 21,500.
- The time it took to write the first draft went from around 14.6 hours to 9.4 hours, then 8.8 hours. The total time it took to write each submission went from about 46.7 hours to 32.5 hours, then 30.9 hours.
- In PilotQ1, readability went up by about 2.2 grade levels, and in PilotQ2, it went up by 2.6.
- In PilotQ1, compliance flags went down by about 61 percent, and in PilotQ2, they went down by 67 percent.
- The share of submissions with AI help went from 0 to around 0.85 and subsequently to 0.90.

These are example values made from the synthetic data. You can now replace them with your real Fill My Cup entries. The charts that were saved for you show the win rate, hours to first draft, and readability gain each quarter.

9.1. Baseline Table

Quarter	Submissions	Awards	Win Rate	Average Award (USD)	Hours to First Draft	Total Hours
Baseline Q1–Q4	24	6	0.25	18,500	14.2	46.7

9.2. Pilot Quarter Example

Quarter	Submissions	Awards	Win Rate	Avg Award	Hours to First Draft	Total Hours	Readability Gain	Compliance Flags Reduced
Pilot Q1	8	3	0.38	21,000	9.1	31.4	2.3 grade levels	61 percent

9.3. Replication and Audit

A single table for each quarter holds all the inputs. There are minor tools in the repository that might help you check the readability and risk of a URL. The score for an opportunity depends on how comparable the text is and some simple assumptions about the funder's length of service, the reviewer's knowledge, and the size fit. Every three months, the team changes the weights or leaves them the same for stability. This makes a clear rank list that staff may look at and change with a written reason.

10. Case Study Narrative

In the first month, the team set up responsibilities and wrote down internal documents including program descriptions and evaluation plans. Opportunity scoring changed the pipeline from a lot of low-probability targets to a smaller collection of prospects that fit the theme better. In the second month, the team utilized the prompt library to write first drafts of the Need and Evaluation sections. They also used the compliance checklist, which included funder language, measurable KPIs, and language that is welcoming to everyone. Changes that were tracked reveal that the story has become much tighter, with clearer linkages between actions and results. The security layer identified a number of unwanted messages that sought for portal credentials and stopped them from getting through. As the library grew, it took less time for staff to write the first complete manuscript, and reviewers said there were fewer structural changes.

11. Reproducibility Package

To get people to adopt, we include:
1. A schema file for the tables in section 4
2. Pseudocode for the functions that calculate the trust score and the URL score
3. A list that connects the quality testing to the needs of the funder
4. A template for partners who want to help with the workflow

Companies can start with spreadsheet formulas and move to scripts as their needs develop.

12. Limitations

The opportunity score is only as good as the records and text signals that go into it. At beginning, small NGOs may not have enough outcomes to match strong models. The security layer just looks at lexical URL characteristics and can't substitute full endpoint protection. Clarity and narrative quality scores are related, but they don't ensure that something will persuade you. These restrictions necessitate prudent interpretation and ongoing human supervision.

13. Conclusion

Combining trust scoring, story quality assurance, and communication safety with a decision science approach can make nonprofit grant solicitation more effective and resilient. Early use of Fill My Cup indicates that it is possible to do in practice and that it leads to measurable process improvements. The framework can be copied and might become a standard for the whole sector, which would cut down on wasted effort and make it easier for community groups to get funding.

Funding Statement: The authors received no funding for the conduct of this research work.

Author Contributions: The authors contributed to the research and writing of this article and have read/agreed to the published version of the manuscript.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data supporting this study are included within the article.

Conflict of Interest: The authors declare no conflict of interest.

REFERENCES

1. Seckel, E., Stephens, B. Y., Rodriguez, F., et al. (2024). Ten simple rules to leverage large language models for getting grants. PLoS Computational Biology, 20(3), e1011863. https://doi.org/10.1371/journal.pcbi.1011863

2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, Vol. 30, pp. 5998-6008.
3. Qiu, H. S., Peng, H., Fosse, H. B., Woodruff, T. K., & Uzzi, B. (2024). Use of promotional language in grant applications and grant success. JAMA Network Open, 7(12), e2448696. https://doi.org/10.1001/jamanetworkopen.2024.48696
4. Godwin, R. C., DeBerry, J. J., Wagener, B. M., Berkowitz, D. E., & Melvin, R. L. (2024). Grant drafting support with guided generative AI software. SoftwareX, 27, 101784. https://doi.org/10.1016/j.softx.2024.101784
5. Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is power: A critical survey of "bias" in NLP. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5454-5476. https://doi.org/10.18653/v1/2020.acl-main.485
6. Kandakoglu, A., et al. (2024). The use of multi criteria decision making in project portfolio selection: A comprehensive literature review. Annals of Operations Research. https://doi.org/10.1007/s10479-023-05564-3
7. Jøsang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2), pp. 618-644. https://doi.org/10.1016/j.dss.2005.05.019
8. Wu, J., & Konstantinidis, S. (2025). A survey on trust and reputation data sharing. arXiv. https://doi.org/10.48550/arXiv.2508.14028
9. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1245-1254. https://doi.org/10.1145/1557019.1557153
10. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. Learning for Text Categorization: Papers from the 1998 Workshop, Vol. 62, pp. 98-105.
11. Sheng, S., Wardman, B., Warner, G., Cranor, L. F., Hong, J., & Zhang, C. (2009). An empirical analysis of phishing blacklists. Proceedings of the Sixth Conference on Email and Anti-Spam (CEAS), Mountain View, CA.
12. Opara, C., Chen, Y., & Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. Expert Systems with Applications, 236, 121183. https://doi.org/10.1016/j.eswa.2023.121183
13. Haq, Q. E. u., Faheem, M. H., & Ahmad, I. (2024). Detecting phishing URLs based on a deep learning approach to prevent cyber-attacks. Applied Sciences, 14(22), 10086. https://doi.org/10.3390/app142210086
14. Schneider, S., et al. (2024). Generative AI governance: An organizational perspective. arXiv. https://doi.org/10.48550/arXiv.2403.08802
15. World Economic Forum. (2024). Governance in the age of generative AI: A 360 approach for resilient governance. World Economic Forum.
16. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982-3992. https://doi.org/10.18653/v1/D19-1410
17. Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (3rd ed.). John Wiley & Sons. https://doi.org/10.1002/9781118548387
18. Kandakoglu, A., et al. (2024). The use of multi criteria decision making in project portfolio selection: A comprehensive literature review. Annals of Operations Research. https://doi.org/10.1007/s10479-023-05564-3
19. Kincaid, J. P., Fishburne, R. P., Rogers, R. L., & Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Research Branch Report 8-75, Naval Technical Training Command, Millington, TN.
20. Cutts, M. (2020). Oxford guide to plain English (5th ed.). Oxford University Press.

Appendix A: Prompt Library Excerpts

Need Statement

- Write 250 to 300 words about the literacy needs of adults in Mecklenburg County using the CASAS context and information from the most recent program reports. Write one version in plain English and one in the format of a formal grant.

Program Impact

- Change the following results into KPIs that can be measured, including goals and deadlines. Make a list of bullet points and a 100-word paragraph.

Evaluation

- Link inputs, outputs, and outcomes to data sources and how often they are collected. Add checks for pre- and post-CASAS, retention, and demographic equity.

Budget Narrative

- Explain each cost in relation to outcomes. Avoid jargon. Keep to 150 words.

Sustainability

- Link partnerships with the library, CPCC, DSS, and AARP Foundation to a funding plan that includes a variety of sources in 120 words.

Appendix B: Compliance and Readability Checklist

- All required sections are there and in the right order
- Each section has a word limit
- The language is simple and at the right grade level
- The language is inclusive and person-first
- The evidence claims can be traced back to named documents
- The budget items match the totals and the narrative
- The evaluation plan lists the tools and the cadence
- There is an accessibility statement
- All links have been checked through the security layer

Appendix C: Governance Policy Summary

- No personal information from students or donors is sent to public AI tools.
- A human editor checks every AI-generated paragraph.
- All drafts and prompts are saved with timestamps.
- All communications with the outside world use organizational email.
- Security warnings are logged and reviewed at a monthly meeting.

Appendix D: Pseudocode

Opportunity Score

```
for opp in opportunities:
    s_fit = cosine(embed(opp.text), embed(program_corpus))
    s_rep = norm(funder_tenor_years) + norm(award_stability)
    s_review = norm(network_overlap)
    s_size = 1 - abs(request_amount - avg_award) / max(avg_award, epsilon)
    T = α*s_fit + β*s_rep + γ*s_review + δ*s_size
    rank_list.append((opp.id, T))
return sorted(rank_list, key=T, reverse=True)
```

URL Risk Score

```
features = [
    length(url),
    entropy(url),
    contains_suspicious_tokens(url),
    tld_rarity(url),
    domain_age_days(url)
]
risk = one_class_model.predict(features)
if risk > threshold:
    warn_user()
    log_event(url, risk)
```