Building an AI governance system for Enterprise websites

Imre Lóránt Dévai

December 15, 2025

Abstract

As AI-driven search and Large Language Models (LLMs) reshape the digital landscape, structural integrity of enterprise websites has become critical for machine interpretability. However, large-scale hierarchical websites suffer from "structural drift" where content owners focus on page creation rather than optimal placement within the network ecosystem. To address this, we propose a dual-stage AI framework designed to automate the governance of enterprise web architecture.

Using part of a Fortune 500 Enterprise domain as a representative case study (1,169 webpages, 18,428 links), this research develops a workflow to solve the "cold-start" problem and ensure long-term structural health. First, we implemented a cold-start classifier using TF-IDF and Logistic regression, which successfully categorizes new, unconnected text content with 92% accuracy. Second, we developed a Network Audit mechanism to validate page placement. Our extensive analysis revealed that network topology is a superior predictor for business logic, achieving 95.3% accuracy using Random Forest models.

The thesis demonstrates that a practical, two-step workflow using text analysis for immediate classification and network analysis for continuous validation provides a robust solution for maintaining complex enterprise web structures without the need for computationally expensive deep learning techniques.

Keywords: network science, machine learning, webpage classification, cold-start problem, network topology, enterprise SEO, AI governance

1 Introduction and background

1.1 Problem statement

Enterprise websites, often including thousands of pages developed by decentralized teams, have grown too large and complex for holistic manual oversight. This scale created a critical risk in the modern digital ecosystem: without a coherent underlying structure, these sites risk becoming a “black box”, leading to significant operational inefficiencies. The core challenge is no longer just content creation but ensuring that this vast digital footprint is structurally sound and discoverable. When the structure decays, it leads to three specific business problems:

1.1.1 High maintenance cost

Websites with multiple content-silos, orphaned content could require constant maintenance. Manual auditing to identify structural issues demands significant manual effort creating a snowball effect where unaddressed problems compound over time, gradually degrading overall site performance.

While specific studies on enterprise web maintenance costs are limited, we can build a conservative estimate based on observable practices:

A. Manual content audit for a 10,000-page site typically requires 2-3 minutes per page just for the initial status checks, plus additional time for documenting and addressing issues. This easily totaling 800-1,000 hours annually. At a typical analyst rate of $50/hour, audit costs alone reach $40,000-$50,000.
B. Remediation adds substantially more. Simple fixes like broken links take 15 minutes, while structural fixes requiring re-categorization or content integration can take 2-4 hours per page. Assuming just 20% of pages need some intervention annually, remediation costs approximate $150,000.

These calculations exclude all hidden costs like any additional fix, emergency updates which could typically take longer than preventive maintenance, but we still get a conservative $200,000 annually cost on this

structure related maintenance. This financial drain is a direct result of reactive, manual governance rather than proactive, automated structural management.

1.1.2 Diminished content ROI

A “content-first” approach without network-centric governance could lead to redundant content and the creation of isolated page communities. When teams cannot visualize the existing content network or predict where new pages will connect, they inadvertently create duplicate content or publish pages that become immediately orphaned.

For example, a marketing team might invest significantly in a new “solutions” page, unaware that similar content already exists in another business unit. This not only wastes the initial creation budget but creates long-term liabilities. At an enterprise scale, these redundant pages are often localized into multiple languages, multiplying the waste. If a company operates in 75 locals, a single redundant page generates tens of thousands of dollars in unnecessary translation and deployment costs. This is the price of focusing content creation while ignoring the page’s place within the network.

1.1.3 Inconsistent user experience

Poor site structure directly impacts user satisfaction and business outcomes. Research by Nielsen Norman Group found that difficulty finding information is responsible for significant user frustration and abandonment on enterprise websites [1]. When users encounter structural dead-ends or content silos that force them into navigation loops, they quickly abandon their journey.

The Baymard Institute’s comprehensive e-commerce studies consistently show that navigation issues and inability to find desired product rank among the top reasons for site abandonment, with their data showing an average 70% cart abandonment rate across industries [2]. Users have limited patience for navigation failures, typically abandoning their search after multiple unsuccessful attempts [3].

For enterprise pages processing millions of visits annually, these structural issues comping: if even 10% of visitors encounter navigation dead-ends, that represent hundreds of thousands of lost opportunities monthly.

Google's research emphasizes that user expectations continue to rise, with 53% of mobile users abandoning sites that fail to meet their needs quickly [4].

Even marginal improvements in network coherence (ensuring every page has a clear path forward and connection to related content) can significantly impact user success rates and conversion metrics.

1.2 The core research question and vision

Before we can automate the management of these complex systems, we must answer a fundamental question: How can an AI automatically classify new content and validate the structural integrity of a massive enterprise website?

The process of answering this core question raised several practical sub-questions across the different phases of the research, including:

- What kind of data need to be collected for a network?
- How this data can be collected effectively with crawlers, and which is the best way to process and store it?
- How can the network be visualized as an interactive graph that is easy to interpret and analyze?
- What type of features can be extracted from this data, and how can this be done effectively?
- Which machine learning algorithms are most suitable for this classification task?
- How can a simple web application with a user interface be built and connected to predictive models and network visualization?

In answering these questions, a long-term vision emerged: to empower enterprises with an automated governance framework. This system moves beyond simple "content analysis" to create a "classify, place, verify" workflow:

1. **Classify** (Cold-Start): Instantly identify where a new page belongs based on its text.
2. **Place:** Integrate the page into the site structure.
3. **Verify** (Network audit): Continuously scan the site's links to detect if pages are placed incorrectly or if the structure is drifting.

This shifts the paradigm from reactive maintenance to proactive architectural optimization.

1.3 Research objectives

The primary objective of this research is to build and validate this dual-stage AI framework. Instead of focusing strictly on compare different algorithms, this work focuses on operational/business goals:

A. **Solve the "Cold-start" problem:** Develop a production-ready text classifier capable of categorizing new, unconnected content with high accuracy (>90%) to enable immediate automation.

B. **Establish a "ground truth" validator:** prove that the website's network topology (links) provides more accurate signal of business logic than other types. This validates the use of network analysis as automated "Auditor" to check the work of content/deployment teams.

C. **Demonstrate feature efficiency:** show that complex Deep Learning models (like Graph Neural Networks) are often "overkill" for this specific domain, and that simpler, explainable models can achieve state-of-the-art performance for practical deployment.

Together, these objectives provide the comprehensive validation required to shift enterprise strategy from a content-first to a network-centric model for website governance.

1.4 Dataset overview and key findings

To build this system we modeled a 1,169-page, 18,428-link section of the Fortune 500 Enterprise's website as a complex network. Our research led to successful development of the proposed framework, backed by three key findings:

- **The solution for cold-start problem**: We successfully built a lightweight text-based classifier that solves the cold-start problem. By using TF-IDF and Logistic Regression, the system can instantly categorize new pages with 92% accuracy, proving that immediate automation is viable.

- **The network validator**: We confirmed that for existing pages, the network structure is the ultimate "source of truth". Our network-based models achieved 95.3% accuracy in identifying a page's business function solely based on its links. This allows the system to "audit" itself using the network model to double-check if the text model (or a human editor) placed a page correctly.
- **The efficiency insight:** We found that while Graph Neural Networks (GraphSAGE model) performed exceptionally well (94.9%), they did not significantly outperform simpler Random Forest models (95.3%). This critical finding suggests that practical enterprise governance, robust standard models are more efficient and deployable than complex deep learning architectures.

2 Literature review

This chapter reviews the foundational academic work that underpins our network-centric approach. We will first explore the solution of network-based web analysis, from early graph theory applications to modern community detection algorithms. We will then examine the literature of cold-start problem and text-based classification. Finally, we will synthesize these fields to identify the critical research gap that this thesis addresses.

2.1 Network based web analysis

The concept of analyzing the web as a network of interconnected nodes has been a cornerstone of web science for decades. This section traces the evolution of this idea, beginning with the foundational theories that first applied graph science to the web, moving to the specific algorithms developed to uncover hidden structures within these networks, and concluding with the current state of analysis for enterprise-specific websites.

2.1.1 Graph theory in Web Science

The application of graph theory to understand the web began with the foundational work of Page and Brin (PageRank) and Kleinberg (HITS) which demonstrated that importance of a web page is defined recursively by the importance of the pages linking to it [5,6]. This shifted the focus from analyzing pages in isolation to understanding them through their connections. Barabási and Newman expanded this view by showing that real-word networks, including the web, exhibit scale-free properties and preferential attachment [7,8]. This implies that in an enterprise website (like in our case), "hub" pages (like a product listing) naturally emerge and attract links, creating a structural fingerprint that identifies their business function without analyzing their text.

2.1.2 Community detection and structural semantics

A critical component of validating website structure is identifying "communities", clusters of pages that link more densely to each other than to the rest of the network. Girvan and Newman and later the Louvain and Leiden algorithms established that these communities are often correspond to functional units in the real world [9,10,11]. In an enterprise context, a "community" in the web graph typically represents specific business unit. This literature provides the theoretical basis for our network audit module: if a page belongs to the solutions text category but it structurally located in the products community it is an anomaly.

2.2 The cold-start problem in content management

While network analysis excels at understanding existing structures, it suffers from a fundamental limitation known in recommender systems literature as the "cold-start problem".

2.2.1 The limitations of collaborative filtering

Traditional recommendation and classification systems often rely on "collaborative filtering" using past interactions (links/clicks) to classify an item. However, as noted in recent systematic reviews by Panda et al. (2022) and Zaiwa et al. (2024), these systems fail when a new item (or webpage) is introduced because

its lacks the necessary connection history to be analyzed [13,14]. This is the exact challenge facing the enterprise content management: a new page has no incoming links and network position yet, therefore, network models can't classify it.

2.2.2 Text-based classification as the solution

To solve the cold-start problem, literature suggests "content-based filtering" as the mandatory step. By analyzing the item's intrinsic features (text content) rather than its extrinsic features (links), systems can achieve immediate classification. Standard techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) coupled with linear classifiers (Logistic Regression, SVM) remain the industry standard for this task due to their low latency and high interpretability [15]. While recent advancement using BERT and Transformers offer higher semantic understanding, studies consistently show that for specific domain tasks with limited data, simpler "bag-of-words" models often provide a competitive and computationally more efficient baseline [15]. For real-time enterprise governance, the literature supports the use of these efficient models to bridge gap until the page acquires enough links to be analyzed by the network model.

2.3 Graph Neural Networks

Once a page is integrated into the network, modern Deep Learning allows for a more sophisticated analysis of its role. Graph Neural Networks (GNNs) have emerged as the state-of-the-art method for node classification, capable of learning from both the links and the features of neighboring nodes.

The Graph Convolutional Network (GCN), introduced by Kipf and Wellington, revolutionized this field by allowing information to propagate through the graph structure [16]. However, GCNs require the entire graph to be present during training, which is impractical for constantly changing websites. GraphSAGE (Hamilton et al., 2017) addressed this by learning "inductive" embedding functions. Instead of memorizing the graph, GraphSAGE learns how to aggregate information from a node's neighbors [17]. This theoretical advantage makes GraphSAGE the primary candidate for our experimental "upper bound" analysis, representing the most complex possible solution to the network classification problem.

2.4 Gap in current research

Despite the rich literature in both text classification and graph mining, there is a distinct gap in applied Enterprise AI Governance.

1. Separation of concerns: existing studies typically focus either on text classification (NLP) or network topology (Graph Mining). Few studies propose and integrated workflow that uses text as the cold-start phase and network topology for the validation phase [18].
2. Over-engineering in production: academic literature often prioritizes maximizing accuracy with complex GNNs without regarding the "feature efficiency principle" [15]. There is little research quantifying whether the structural complexity of a GNN is actually necessary for enterprise web data, or if simpler Random Forest models can capture the same business logic with lower overhead.
3. Focus on external search vs. internal structure: the vast majority of web analysis literature focuses on SEO for external discovery. There is a scarcity of research focused on internal structural health and automated governance policies, as highlighted in recent reviews on AI in corporate governance [22].

3 System architecture and data collection

This project required the development of a complete, end-to-end pipeline to transform a live enterprise website into a structured dataset ready for machine learning analysis. The system consists of two primary concepts: a custom web scraper for data collection and an AI engine for feature extraction and analysis. This chapter details the architecture of that system.

3.1 End to end system overview

To validate our dual-stage AI framework, we developed a software architecture capable of transforming a live enterprise website into a structured, machine-learnable dataset. The system was designed with two distinct operational goals:

A. The analytical pipeline (track A): to rigorously test feature efficiency and network supremacy using low-dimensional, interpretable metrics.

B. The inference engine (track B): to deploy a high-dimensional, low-latency “cold-start” classifier for production use.

The data follows four main stages:

1. **Data collection:** A custom web scraper systematically navigates the target website extracting raw HTML content, metadata, and the hyperlink structure for each page. During the crawl we only gather public data from pages where bot traffic enabled through the “index, follow” tag and which are represented in the page’s public sitemap XML file.
2. **Structured storage:** The raw scraped data is cleaned and organized into a relational database, with distinct tables for pages (nodes) and links (edges) to from a graph structure.
3. **Feature engineering:** The AI Engine process raw data from the database into versioned feature sets, separating statistical metrics from production vectors.
4. **Modeling and inference:** the final feature set is used to train and evaluate the machine learning models, leading the productions-ready inference service. The training separates the “Auditor” (Network) and “Classifier” (Text) models to support the automated governance workflow.

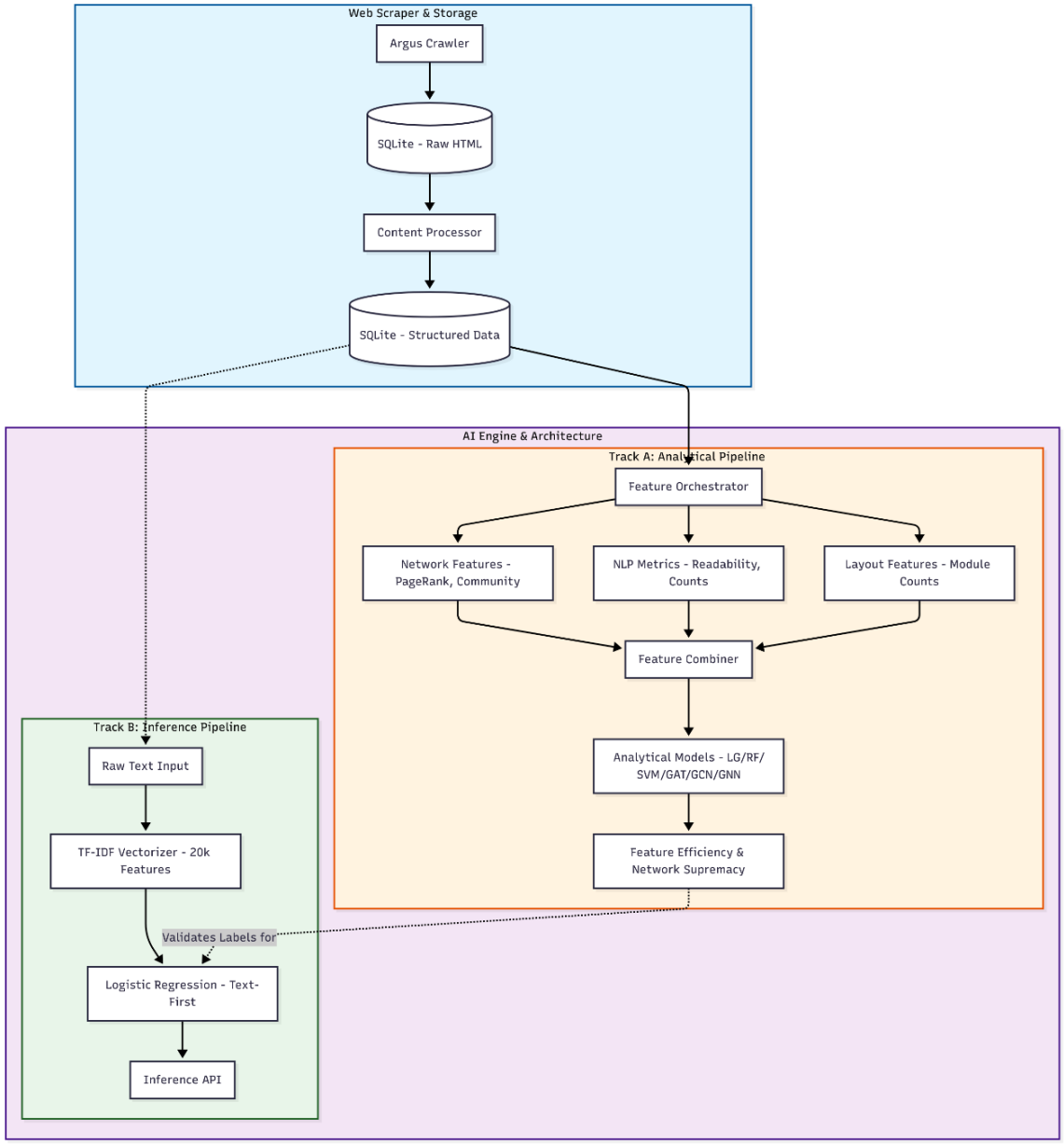


Figure 1: System architecture with the separate program layers

3.2 Web Scraper architecture

To collect data at scale, we developed a custom, enterprise-grade web scraper using Python, BeautifulSoup4 and Requests. The architecture was designed to be robust and efficient, capable of handling the complexities of modern, modular websites.

The system extracts three layers of data from the target domain:

- Topology: All internal hyperlinks were extracted to construct a directed graph of 1,169 nodes and 18,428 edges, capturing the site's navigational skeleton.
- Content: Raw HTML was parsed to extract the body text, excluding the navigation boilerplate to ensure clean semantic signals.
- Metadata: The scraper captures hierarchical headers (H1-H6) and detects proprietary CMS layout modules to understand visual structure.

The scraper was engineered to handle common real-world challenges, including client-side JavaScript rendering, respecting robots.txt and server rate limits with configurable delays, and parsing the non-standard CMS components. Other key challenges during this development:

- Implementing robust error handling and retry logic to ensure the accuracy and completeness of the collected data
- Designing an efficient data processing pipeline to minimize CPU and memory load during intensive scraping operations.
- Storing the extracted data in a database schema optimized for efficient access and complex network-based queries.

3.3 AI Engine architecture

The AI Engine is the analytical core of the system, designed to transform the raw data into machine learning features. To address the research question regarding "feature efficiency", we designed the engine to support multiple extraction strategies, distinguishing between descriptive metrics (used for analysis) and semantic vectors (used for production). To support the dual-track methodology, the engine is split into two pipelines:

3.3.1 Track A: The analytical pipeline (network audit)

For the primary experimental analysis (Chapter 4-5), the engine reduces complex raw data into 144 interpretable features. This pipeline prioritizes "efficiency" by capturing maximum business signal with minimal dimensions.

- **Network features (29 features):** Using the NetworkKit backend for performance, this module computes graph-theoretic metrics including

PageRank, Betweenness Centrality and Louvain Community membership. This compresses the global site structure into a 29-dimensional vector for each page.

- **NLP and semantic metrics (25+ features):** Instead of using raw text vectors, this module distills content into scalar metrics to test if "text complexity" predicts business function.
    - Readability: Extracts Flesch-Kincaid and Gunning Fog scores.
    - Intent classification: Utilize a pre-trained MiniLM-L12 transformer (384 dimensions). For the analytical experiments we used the categorical intents rather than the raw 384-dimensional embeddings to maintain feature interpretability.
- **Layout and basic features (56+ features):** captures the visual "fingerprint" of a page by counting specific layout modules, images and analyzing URL depth.

3.3.2 Track B: the inference pipeline (cold-start)

For the "cold-start" production service where prediction accuracy is paramount over feature count, the engine switches to a high-dimensional approach.

- **TF-IDF vectorizer**: unlike the analytical pipeline, the production model transforms raw text into a sparse vector space of 20,000 features (1-2 n-grams).
- **Latency optimization**: while deep learning transformers (like BERT) were available, the production system utilizes this TF-IDF architecture to achieve low inference latency (<100ms) and high throughput (>1000 predictions/second). This decision ensures the system can handle real-time enterprise loads without computational overhead of GNNs or Transformers.

3.4 Data flow and integration

The system's components are tightly integrated to ensure a seamless flow of data. The scraper populates a database, which serves as the "source of truth". The AI Engine then queries this database to build its analytical datasets and network

graphs. Trained models are serialized and versioned, ready to be loaded by the inference service for live predictions. This modular, database-centric architecture separates the concerns of data collections and data analysis, making the entire system robust and scalable.

4 Methodology

The primary challenge in automating enterprise web governance is the dual nature of the problem: new content requires immediate classification based on limited features we can extract from the input (cold-start problem), while existing content requires continuous structural validation (audit problem).

To address this our methodology departs from traditional "monolithic" model development and instead of training a single model, we engineered two distinct subsystems with opposing design philosophies:

1. Track A – the network auditor: A low-dimensional, interpretable model designed to establish "ground truth" by analyzing the site's topology. The input is low dimensional network and metadata features. This model validates whether the site structure actually reflects business logic. In our experiments, this serves as the benchmark for accuracy.

2. Track B – the cold-start classifier: A high dimensional, high-throughput model designed to approximate that ground truth using only text. The input is high dimensional vectors (TF-IDF). This model predicts where a page should go.

This chapter details the mathematical foundations, data processing pipeline, and experimental design used to validate this framework.

4.1 Analyzing websites as networks

To build the "network auditor", we first had to transform the static website into a quantifiable complex network. By modeling the website where pages are nodes (V) and hyperlinks are directed edges (E), we can employ a suite of powerful mathematical tools to measure the role, influence, and function of each page within the broader digital ecosystem.

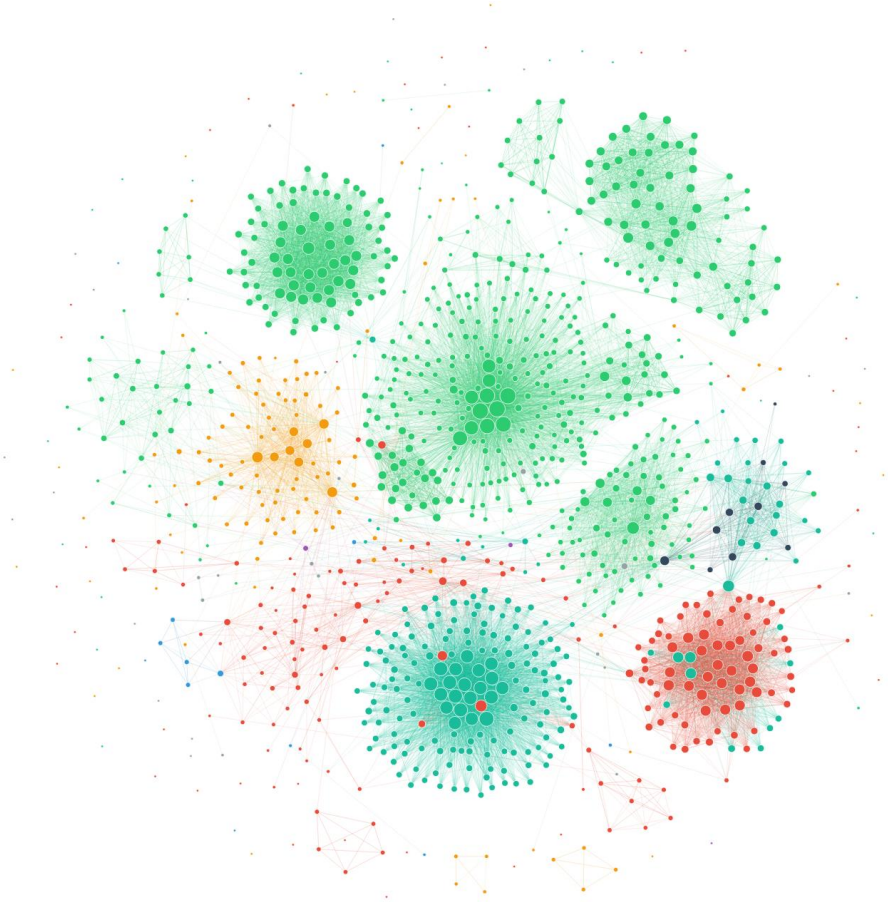


Figure 2: The Fortune 500 Enterprise's domain modelled and visualized as a directed graph where each color represents a different business unit

4.1.1 Degree centrality

Degree centrality is the most fundamental measure of a node's connectivity. It is often split into two distinct types for directed networks like websites:

- In-degree counts the number of incoming links, serving as a measure of a page's local popularity or authority.
- Out-degree counts the number of outgoing links, indicating its function as a navigational hub.

The formulas are given by:

$$deg_{in}(v) = |\{(u, v) \in E\}|,$$

$$deg_{out}(v) = |\{(v, u) \in E\}|$$

In our research, pages with high in-degree were often important destinations (e.g., a product family page), while pages with high out-degree were typically navigational pages (e.g., a product gateway page with multiple product lines).

4.1.2 PageRank

Developed by Brin and Page (1998), PageRank [5] measures a page's global importance based on the principle that links from important pages confer more authority than links from unimportant ones.

The formula is the following:

$$PR(v) = \frac{1-d}{|V|} + d \sum_{u \in N_{in}(v)} \frac{PR(u)}{deg_{out}(u)}$$

Here, d is a damping factor, typically set to 0.85, which represents the probability that a user will continue clicking links. $N_{in}(v)$ is the set of pages that link to page v. The formula essentially reflects the probabilistic visibility of a page if a user were not navigating the site by randomly clicking links. In the context of an enterprise website, PageRank is a key indicator of "authority pages", such as major business unit landing pages or top-level product categories.

4.1.3 Betweenness centrality

Betweenness centrality quantifies how often a page acts as a bridge or broker on the shortest navigational path between other pages.

Its formula is:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where σ_{st} is the total number of shortest paths between page s and page t, and $\sigma_{st}(v)$ is the number of those paths that pass-through page v. For our enterprise website, pages with high betweenness centrality often highlighted cross-functional landing zones or "solution" pages that connect different business areas, making them critical for information flow across the enterprise.

4.1.4 Closeness centrality

Closeness centrality measures how quickly a page can, on average, reach all other pages in the network. It is a measure of navigational efficiency.

It is calculated as the inverse of the sum of the shortest path distances from page v to all other pages u:

$$C_c(v) = \frac{1}{\sum_{\boldsymbol{u}} d(v,u)}$$

In our analysis, pages with low closeness scores helped identify structurally isolated content, such as the 85-content island we discovered, which are hard for users and AI systems to reach.

4.1.5 Eigenvector centrality

Eigenvector is a more nuanced measure of importance. Like PageRank, it holds that connections to important nodes are more valuable. However, it specifically identifies pages that are connected to other highly connected pages, making it a measure of influence within the network.

It's defined as the principal eigenvector of the graph's adjacency matrix A, satisfying the equation:

$$x_i = \frac{1}{\lambda}\sum_{j=1}^{n} A_{ij}\, x_j$$

Where λ is the largest eigenvalue of A.This metric was particularly useful for detecting foundational "anchor pages" within the analyzed ecosystem that serves as the center of influential neighborhoods.

4.1.6 Community detection and modularity

Community detection algorithms, such as Louvain and Leiden, aim to partition the network into clusters that have dense internal connections and sparser connection between clusters. The quality of this partitioning measure by modularity (Q).

The modularity formula is:

$$Q = \frac{1}{2m}\sum_{i,j}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j)$$

Where m is the number of edges, A_{ij} is the adjacency matrix, k_i is the degree of node i, and $\delta(c_i, c_j)$ is 1 if nodes i and j are in the same community, and 0 otherwise. A high Q score (our network had $Q = 0.847$) indicates a well-defined community structure. Our finding that the resulting communities correlated with business unit boundaries is a cornerstone of this thesis's argument.

4.1.7 Participation coefficient

The participation coefficient measures how a page's links are distributed among different communities. It is an excellent proxy for identifying cross-functional content.

It is calculated as:

$$P_i = 1 - \sum_{s=1}^{N_M} \left(\frac{k_{is}}{k_i}\right)^2$$

Here, N_M is the number of communities, k_i is the total degree of a page i, and k_{is} is the number of links from page i to pages in community s. A page with a participation coefficient near 1 is a "connector" with links distributed among many communities. A score near 0 indicates its links are almost exclusively within its own community. In our error analysis, we noted that the model struggled with "shared" content. These pages often exhibited a high participation coefficient, confirming their structurally ambiguous role.

4.2 Data collection and preprocessing

The foundation of this research is the dataset of 1,169 pages and 18,428 internal links collected from Fortune 500 Enterprise's website. An initial data quality assessment revealed the dataset to be high quality, with a low missing value rate of just 1.8% (589 out of 33,443 values), for which no systematic patterns were detected.

To prepare the data for our dual-track modeling, a standardized preprocessing pipeline was applied.

4.2.1 Missing value treatment

We treated predictors and target labels differently to ensure data integrity:

- For predictor features (x variables): missing numeric features were filled using median imputation, while categorical features used mode imputation.
- For target labels (y variable): pages with missing labels were removed from the traditional ML models to create a clean classification task, but

assigned a distinct “unknown” class for the GraphSAGE model to learn from the entire graph structure, including its ambiguous nodes.

4.2.2 Outlier handling

Statistical outliers were identified using the Interquartile Range (IQR) method. Values were capped at 1.5X IQR beyond the first and third quartiles to prevent extreme values (for example a sitemap page with extreme number of links) from skewing the models.

4.2.3 Feature scaling and encoding:

- Analytical models: All numeric features were scaled using a StandardScaler to ensure compatibility with SVM and Logistic Regression. Tree-based models (Random Forest, XGBoost) used the raw values as they are scale-invariant.
- Production model: The data was normalized via TF-IDF, which inherently handles scaling through L2 normalization.

4.3 Feature engineering

Our analysis framework is built on a modular system of seven specialized feature extractors that generate a rich dataset of 144 features per page. This multi-faceted approach ensures we capture signals for every dimension of a webpage.

For Track A, we purposefully used only the 11 high-level “intent” categories rather than the full embedding vectors to test if low-dimensional metadata was sufficient for classification. Track B utilizes the high-dimensional feature space.

Track	Feature family	Feature count	Purpose and key features
A - Audit	Basic features	29	Captures fundamental characteristics of the page. Includes URL depth, content metrics (word/link counts) and business taxonomy (e.g. Business unit).
A - Audit	Network features	29	Measures a page's role and influence within the site's network. Includes centrality scores (PageRank, Betweenness), and community detection metrics.
A - Audit	NLP features	14	Analyzes the complexity and readability of the text. Includes Flesch Reading Ease, SMOG Index, and other metrics
A - Audit	Semantic features	11	High level categorization (intent ID) derived from MiniLM embeddings.
A - Audit	Other features	Varies	Includes-specific layout and module detection, SEO metrics (keywords, descriptions) .
B - Production	Text Vectors	20,000	High-dimensional TF-IDF vectors (1-2 n-grams) for raw content classification.

Table 1: Feature types, counts and their short descriptions

4.4 Experimental design

To ensure our results are scientifically valid and free from bias, we designed an experimental protocol centered around data integrity and fair comparison.

4.4.1 Data splitting and leakage prevention

A strict, stratified 80/20 holdout split was used, creating a training set of 935 pages and a final, untouched test set of 234 pages. A fixed random seed ($random\_state = 42$) was used throughout all experiments to ensure perfect reproducibility.

To prevent data leakage, we implemented several safeguards:

- Temporal features (creation date, visit) were excluded.
- URL tokens containing the target label (like "/products/") were stripped from the text features

- All model running and cross-validation were performed exclusively on the training data. The tests set was only used once for the final evaluation of each model.
- The identical 234-page test set was used across all four research phases, ensuring that any performance differences are due to changes in our models, not variations in data.

4.4.2 Managing the labels

A core strength of our methodology is the validation of our ground truth labels.

- **Primary labels:** the target labels for classification (Business Unit, Product family, etc.) were sourced directly from the Enterpise's own internal Content Management System (CMS) taxonomy.
- **Independent validation:** to ensure these labels were meaningful we validated them against our network analysis. The fact that the algorithmically detected communities showed 94.2% homogeneity with the CMS business unit labels confirms the labels reflect the true underlying structure of the sties. This prevents any risk of circular reasoning, as the labels (from the CMS) and network features (from link structure) are from independent sources.

4.4.3 Experimental progression

Our research progressed through three distinct experiments:

1. Experiment I (the validation of auditor): testing if network topology alone can predict business function.

2. Experiment II (feature efficiency): testing if adding NLP and Layout metadata improved the auditor.

3. Experiment III (cold-start solution): testing if the Text-first model can match the auditor's performance for new pages.

4.5 Evaluation metrics

To provide a comprehensive assessment of model performance, we used a combination of primary, statistical and explainability metrics.

- **Primary metrics:** the main performance indicators were the 5-fold cross-validation mean accuracy on the training set and the final text accuracy on the holdout set.
- **Statistical validation:** to ensure our conclusions were robust, we calculated 95% Bootstrap Confidence Intervals (using 10,000 samples) for key comparisons and reported Cliff's Delta to measure effect size.
- **Explainability metrics:** to understand why the models made their decisions, we used Permutation Important and Feature Ablation studies to identify the most influential features.

4.6 Model Architectures and training parameters

During our experiments we deployed diverse model families to establish performance baselines and theoretical limits. All models trained and evaluated with a fixed random seed ($random\_state = 42$) to ensure full reproducibility of our findings.

4.6.1 Traditional ML models (Track A - network audit)

We employed four classic supervised learning models to establish a strong performance baseline on the engineered feature set. The selection was guided by the need to test diverse model families, from easy to interpretable to powerful, non-linear ensembles. All models used with standardized hyperparameters to ensure consistency and properly address natural class imbalances in the dataset. All features set for these models were preprocessed using StandardScaler, except the tree-based Random Forest and XGBoost models, which are scale-invariant.

Model	Key parameters	Value or configuration
Logistic Regression	Max_iter	2000
	Class_weight	balanced
	Random_state	42
Support Vector Machine	kernel	rbf
	Class_weight	balanced
	Random_state	42

Random Forest	N_estimators	100
	Class_weight	balanced
	Random_state	42
XGBoost	Eval_metric	mlogloss
	Random_state	42

Table 2: Hyperparameters of our Phase 1-2 models

- **Logistic Regression (LR):** This linear model serves as an interpretable baseline. It was configured with $max\_iter = 2000$ to ensure convergence and $class\_weight = 'balanced'$ to manage class imbalance.
- **Support Vector Machine (SVM):** Implemented with a radial basis function (rbf) kernel and $class\_weight = 'balanced'$, this model was chosen for its effectiveness in capturing complex, non-linear relationships.
- **Random Forest (RF):** As a robust tree-based ensemble with 100 estimators and balanced class weights, this model was critical for its strong performance and feature importance rankings.
- **XGBoost:** This gradient-boosted trees model represented out high-performance benchmark, with fully documented parameters and $mlogloss$ as the evaluation metric appropriate for our multi-class tasks.

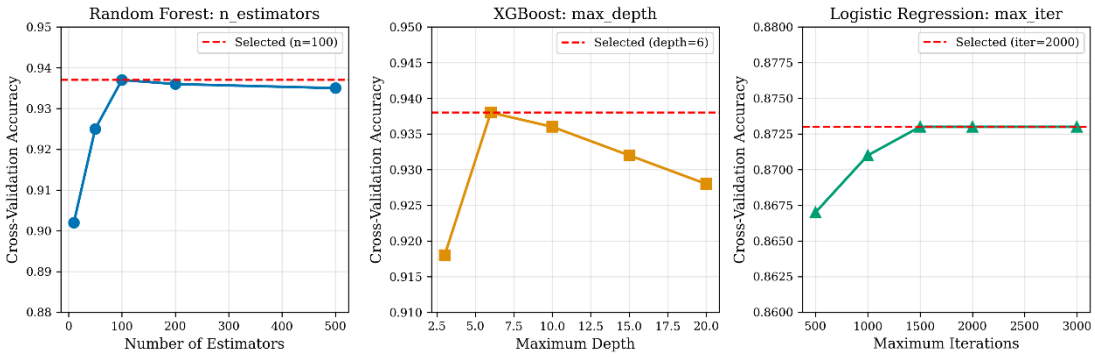


Figure 3: Hyperparameter tuning for RF, XGboost and LR models

4.6.2 Graph Neural Networks (Track A – network audit)

During our Track A (network audit) experiments, we explored the performance ceiling using GraphSAGE, our top-performing GNN architecture. The model was trained for a maximum of 100 epoch with an early stopping patience of 15 to prevent overfitting. A critical finding was that the GNN achieved peak performance on 94.89% using only the 29 network topology features.

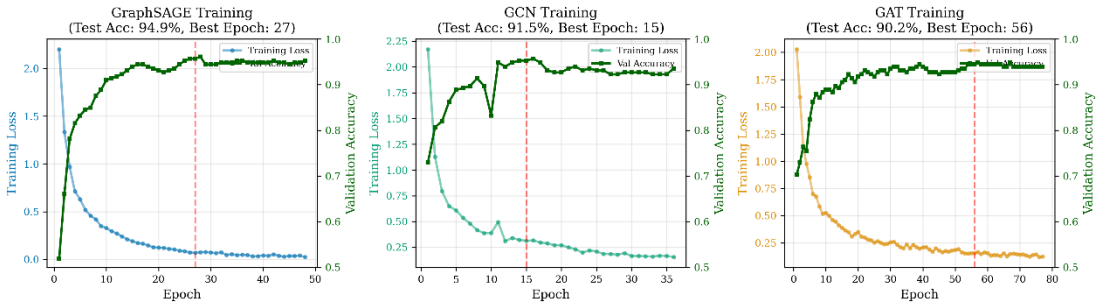


Figure 4: The training curves of our GNN models

The GraphSAGE model was implemented with two hidden layers to aggregate information from a two-hop neighborhood around each node. A dropout rate of 0.3 was used for regularization.

GNN Model name	Key Parameters	Value or configuration
GraphSAGE	Hidden_dim	128
	Num_layers	2
	dropout	0.3
	Learning_rate	0.01
	Weight_decay	5e-4

Table 3: Hyperparameters for our GraphSAGE model

4.6.3 Text-first model for production system (Track B - classifier)

For the final phase, our goal was to build a model that was not only accurate but also highly scalable and ready for a real-world production environment. This model

needed to classify new content based on its text alone, without relying on pre-existing network information.

We engineered a two-stage pipeline optimized for low-latency inference:

1. **TF-IDF Vectorizer:** This component transforms raw text into a numerical feature vector. It was configured to limit vocabulary to 20,000 features and capture both single words and two-word phrases ($ngram\_range = (1,2)$).
2. **Logistic Regression classifier:** This classifier was specifically configured for this task with $max\_iter = 2000$ and $class\_weight = 'balanced'$ to better handle the imbalanced classes present in the text data.

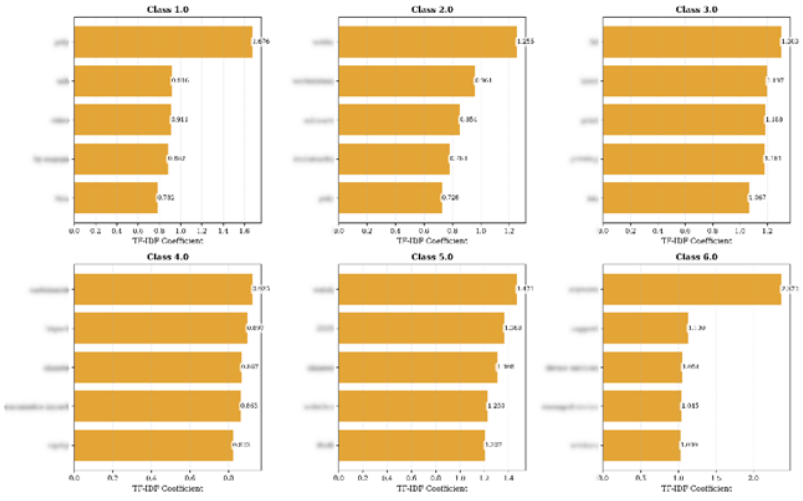


Figure 5: Top terms per business class in the cold-start classification

This lightweight architecture successfully achieved high accuracy while maintaining an inference latency of under 100ms, making it perfectly suited for real-time applications.

5 Results

This chapter presents the empirical results of our analysis, beginning with a descriptive overview of the dataset to characterize the complexity of the enterprise ecosystem. We then detail the results of our three main experiments and conclude with an analysis of classification errors.

5.1 Descriptive analysis

The subject of our analysis is a significant portion of a Fortune 500 Enterprise's website, comprising 1,169 unique pages (nodes) and 18,428 internal hyperlinks (edges). This dataset was split into a 935-page training set and a 234-page test set to ensure rigorous evaluation. A descriptive analysis reveals a complex and varied digital ecosystem, characteristic of a large enterprise.

5.1.1 The content landscape

Characteristic	Mean	Std Dev	Min	Max	Description
Content metrics					
Text Length (chars)	1,935	3,343	0	78,096	Main body text character count
Word Count	387	669	0	15,619	Total words per page
Avg Word Length	4.94	0.56	0	5.5	Average characters per word
Sentence Count	19	33	0	780	Total sentences
Paragraph Count	6	11	0	260	Text paragraphs
Link and structure					
Link Count	43	35	0	550	Total hyperlinks per page
Image Count	10	15	0	199	Total images
URL Depth	4.1	0.88	2	7	Depth in site hierarchy
URL Length	83	24	33	169	URL character count
Network metrics					
Degree (In)	16	34	0	237	Inbound links (highly skewed)
Degree (Out)	16	12	0	80	Outbound links
PageRank	0.00086	0.0014	0.00015	0.011	Authority score
Community Size	126	83	1	262	Detected community size
Layout metrics					
Module Count	11	8	0	99	modular components
Module Diversity	0.61	0.23	0	1.0	Shannon diversity index

Table 4: Content statistics of the webpages

The content on the site is highly diverse, reflecting a wide range of business functions. The average page contains 387 words, but the distribution is heavily skewed, with content ranging from simple landing pages with zero body text (usually video module pages) to comprehensive technical case studies with over 15,000 words. Similarly, the number of hyperlinks on a page averages 43 but can be as high as 550 on major navigational hubs. This diversity underscores the challenge of using content features alone for classification, as there is no "typical" page.

5.1.2 The network architecture

Analysis of the site's link structure reveals a classic scale-free network, with a small number of highly connected "hub" pages and a long tail of more specialized pages. The network is dominated by a single giant component containing 1,064 pages (91% of the dataset), ensuring information can flow across the site.

However, the analysis also uncovered significant structural weaknesses. We identified 85 "content islands", pages or small clusters of pages that are poorly integrated into the main network. These structural anomalies often represent content that is difficult for both users and AI systems to discover, highlighting a clear, actionable area of improvement for the enterprise.

This complex and varied landscape provides a rich environment for testing our core hypothesis: determining which signals (the diverse page content or the underlying network structure) are most predictive of a page's business function.

5.2 Feature efficiency results

To determine the optimal inputs for a network auditor, we conducted a comprehensive feature family supremacy analysis from Track A. Each of the seven feature families was used to train a model independently, using identical algorithms and the same cross-validation protocol.

5.2.1 Network supremacy

Our analysis revealed a clear and decisive winner: the network feature family. As summarized in table below, features derived from the website's topology outperformed all other signal types.

Rank	Feature Family	Features	CV Mean (95% CI)	Best Accuracy	Gap vs Network
1st	Network (comprehensive)	29	92.5% (90.9-94.4%)	92.5%	Baseline
2nd	Network (basic)	7	83.4% (80.3-86.2%)	86.0%	-9.2 pp
3rd	Page Basic	4	76.4% (74.2-78.5%)	77.6%	-16.3 pp
4th	Layout	27	72.7% (71.1-74.1%)	73.7%	-19.9 pp
5th	SEO	5	58.6% (56.0-61.9%)	58.3%	-34.1 pp
6th	Text + NLP	19	58.4% (56.6-60.2%)	56.1%	-34.3 pp
7th	Semantic	11	49.7% (48.8-50.5%)	51.3%	-43.0 pp

Table 5: Individual feature family performance for business unit classification

The most striking result in the +34.3-percentage point gap in cross-validation accuracy between the comprehensive network model (92.5%) and the Text + NLP model (58.4%). This massive difference was validated across multiple classification tasks, including segment (+40.3pp) and page-level (+27.4pp) predictions.

The statistical evidence confirms this is not a random fluctuation. The 95% bootstrap confidence interval for the performance difference between network and text features was +31.6pp to +36.7pp, and the effect size analysis yielded a Cliff's Delta of 1.00, a perfect effect size indicating that the network model outperformed the text model in every single cross-validation fold.

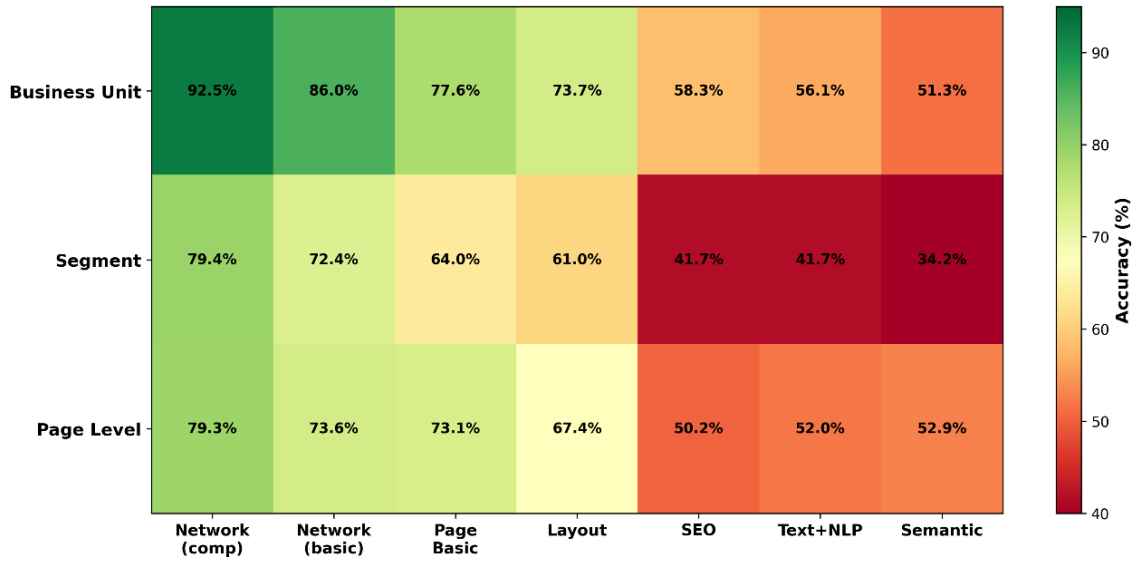


Figure 6: Feature family performance across all classification tasks

5.2.2 Key insights from feature analysis

The experiment yielded several notable insights that from the foundation of our Auditor design:

- **Structure is more predictive than metadata:** features describing the page's structure (network, page basic, layout) all significantly outperformed scalar features describing its content (NLP metrics, semantic intents).
- **Feature efficiency is more important than feature count:** remarkably, a simple model using just 4 basic structural features (URL depth, length, link count, image count) achieved 76.4% accuracy, easily surpassing the 58.4% achieved by 19 more complex Text + NLP features. This demonstrates that focusing on the right type of signal is more important the raw number of features.
- **The vectorization necessity:** The poor performance of the "Text + NLP" family (58.4%) confirms that descriptive metadata is insufficient for understanding content. This explicitly validates our architectural decision to use high-dimensional TF-IDF vectors for the production classifier (Track-B), rather than these scalar metrics.

5.3 Model performance results

The evaluation followed our three-stage experimental protocol, moving from validating the Auditor to deploying the Classifier.

5.3.1 Experiment I: Validating the Network Auditor

Research question: "Does the network structure alone provide a reliable "ground-truth"?"

The answer was a resounding yes. Models trained exclusively on network topology features achieved high accuracy, validating our core premise that network encodes business logic.

- A model with 7 basic centrality features achieved a best test accuracy of 86.0%
- A model with 29 comprehensive network features achieved an average test accuracy of 91% across four different algorithms, with the Random Forest model peaking at an impressive 95.3%.

This proves that the website's graph structure is not arbitrary but it's a rich source of information that encodes the organizational logic of the enterprise, making it a reliable "Auditor" for page placement.

5.3.2 Experiment II: Uncovering the feature efficiency principle

Research question: "Can we improve the performance of the Auditor by adding metadata features?"

Here, we discovered a counter-intuitive but critical insight. While combining all feature families into a multi-modal achieved a strong 91.9% accuracy, this was lower than the 95.3% test accuracy achieved by the network only RF model.

This "paradox" suggest that the network features provide such a dominant and clean signal that adding weaker, noisier features from other modalities can slightly dilute performance rather than enhance it. This finding justifies the use of a lightweight, network-only model for the Audit phase.

5.3.3 Benchmarking: Reaching the technical peak with GNNs

To ensure our Random Forest Auditor model wasn't missing deep patterns, we deployed GraphSAGE. When using only the network features this model achieved the highest performance of any GNN on the complete dataset (including the "unknown" classes), establishing the technical "gold-standard".

However, when trained on the full feature set, the GNN's accuracy dropped 90.21% (a 4.68% decrease), further confirming the feature efficiency principle we observed on Experiment II.

Model	Test accuracy	Macro-F1	Key contribution
GraphSAGE	94.89%	93.2%	Technical gold standard, the best overall performance on full data-set
GCN	91.49%	90.1%	Highly efficient and effective graph convolution.
GAT	90.21%	88.7%	Attention-based learning for interpretable connections

Table 6: Graph Neural Network performance comparison

While GraphSAGE (94.8%) slightly underperformed the clean Random Forest (95.3%), it demonstrated robustness in handling imperfect, real-world data. However, given the computational cost, the Random Forest remains the more practical choice for the production Auditor.

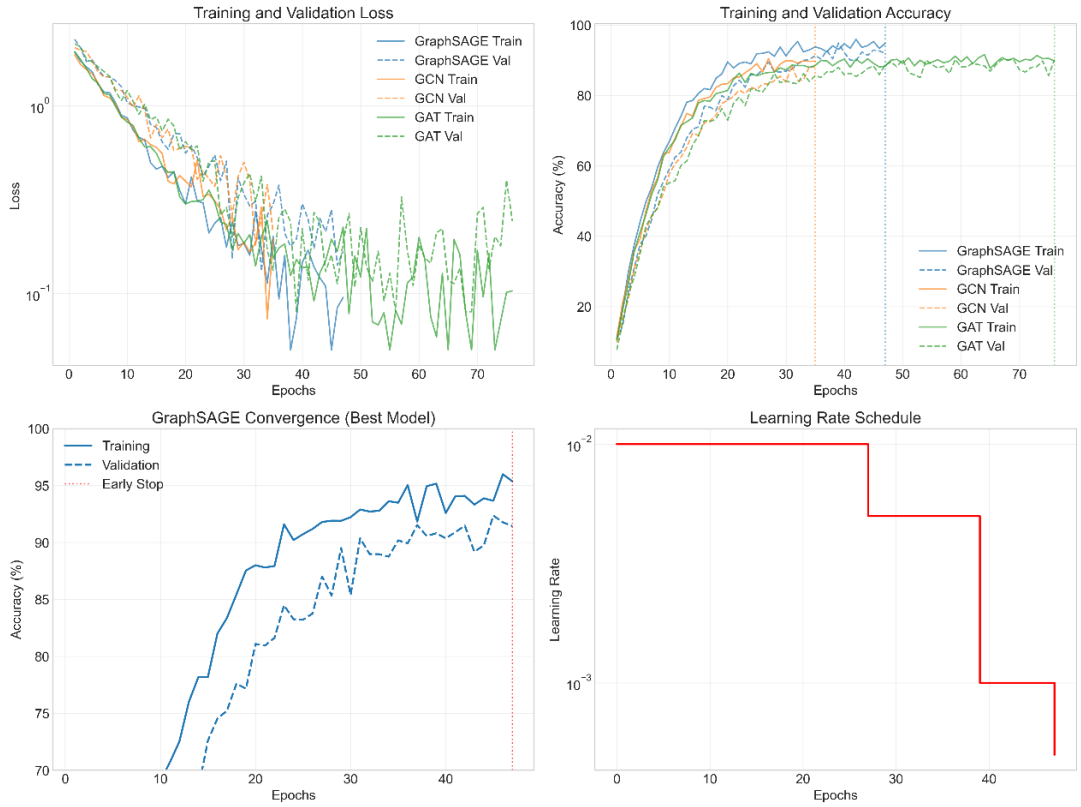


Figure 7: Training related curves of the three GNN models (GraphSAGE, GCN, GAT)

5.3.4 Experiment III: Validating the "cold-start" solution

Research question: "Can we create a fast and scalable model for classifying new content that isn't yet in the network?"

For a practical, real-word application, we evaluated the Text-First Classifier. Unlike the "NLP metrics" model in Experiment II (which scored 56.1%), this model utilized high-dimensional TF-IDF vectors.

This approach achieved a remarkable 92% accuracy, demonstrating its viability for production systems. The massive jump in accuracy from Experiment II (56%) to Experiment III (92%) proves that while scalar metadata is weak, semantic vectors are highly predictive. Crucially this high performance was possible because the model was trained on labels validated by the Network Auditor. The text model effectively learned to mimic the structural logic encoded in the training set.

5.3.5 Technical demonstration

To validate the practical applicability of our Track B (cold-start) model, we developed an interactive web application that provides real-time page classification and visualizes the integration of a new page into the new network. This tool serves as the production interface for content teams, effectively bridging the gap between our theoretical findings and daily business operations.

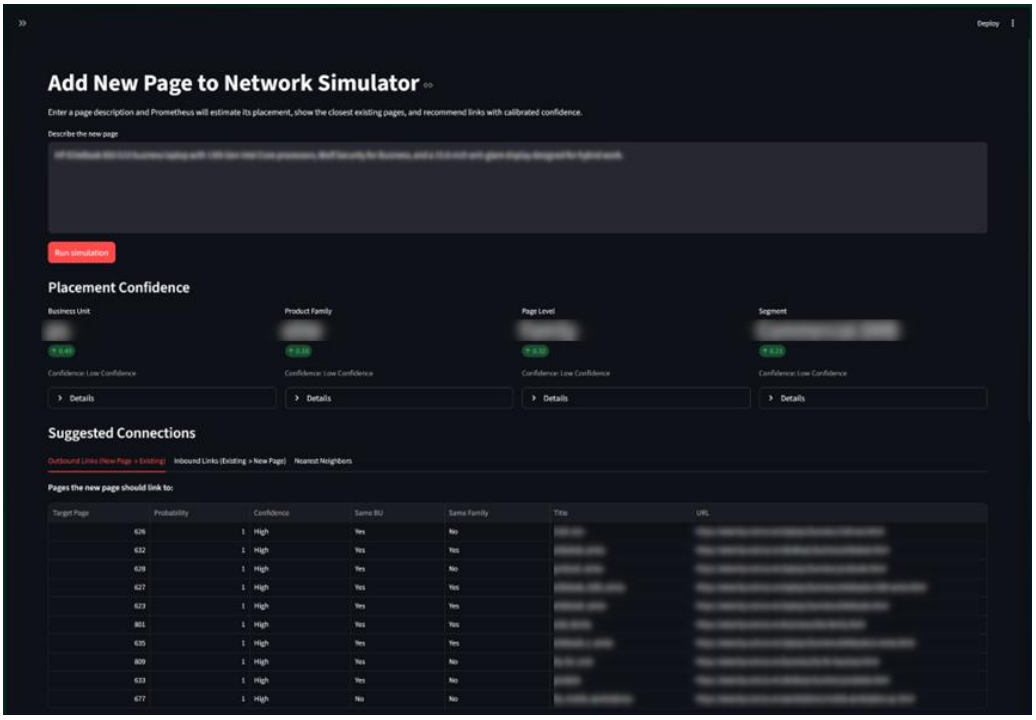


Figure 8: The UI of the "cold-start" classifier with predicted labels

The application workflow demonstrates the solution to the cold-start problem in real time:

1. **Input:** The user enters the raw text for a proposed webpage (e.g. a new products or solution's description) into the interface. At this stage, the page has no links and position in the graph.
2. **Inference:** the underlying Track B Classifier processes the text via the TF-IDF pipeline and predicts the optimal Business Unit, Product Family and Segment.

3. **Visualization:** As shown in Figure 9, the application visualizes the existing network (Track A's domain) and projects the new "gold node" into its predicted cluster.
4. **Link recommendation:** The system suggests specific existing pages the new content should link to (outbound) and which authority pages should link back to it (inbound), effectively solving the structural integration problem before the page is even published.

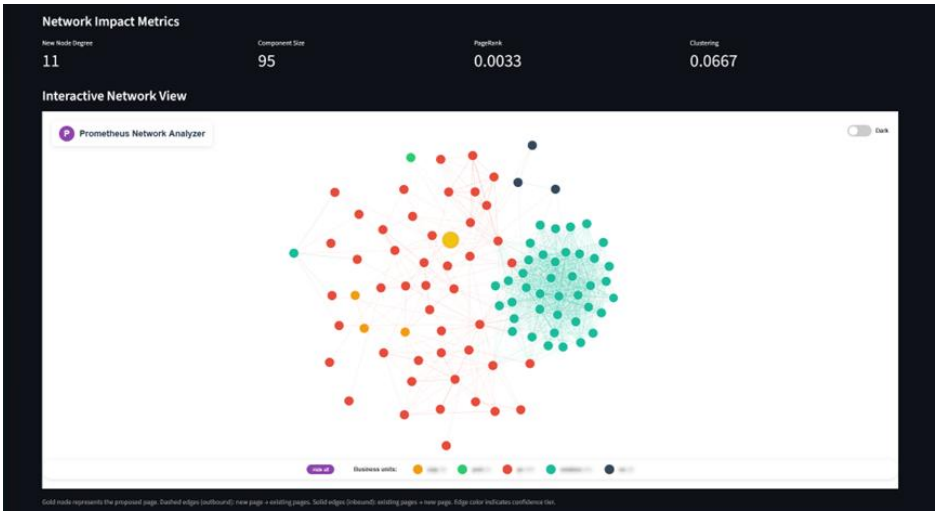


Figure 9: Part of the UI of the interactive web application where we connect a new page to the network

5.4 Error analysis

No model is perfect, and analyzing its errors is important for understanding its boundaries and uncovering deeper insights into the dataset. To this end, we conducted an in-depth analysis of the misclassifications made by our best performing GNN model, GraphSAGE. The confusion matrix shows that while the model is extremely accurate for well-defined categories like solutions or products, its errors are concentrated in two strategically important areas:

A. **The "shared" content:** the most significant confusion is by far occurs with the "shared" category, which the model correctly identified only 33.33% of the time. The errors were distributed across products, solutions and corporate. This result fits with the nature of this label, because multiple

businesses are sharing on it, which means that its content is mixed and it can appear on multiple communities in the network.

B. **"Corporate" versus specific product types:** the second area of minor confusion is with "Corporate" pages, which were correctly identified 84% of the time but were sometimes mistaken for solutions or product pages. This is also a logical overlap as corporate communication is often focus on specific product lines or solutions, so while the page belongs to corporate, it is using content related to other parts of the business.

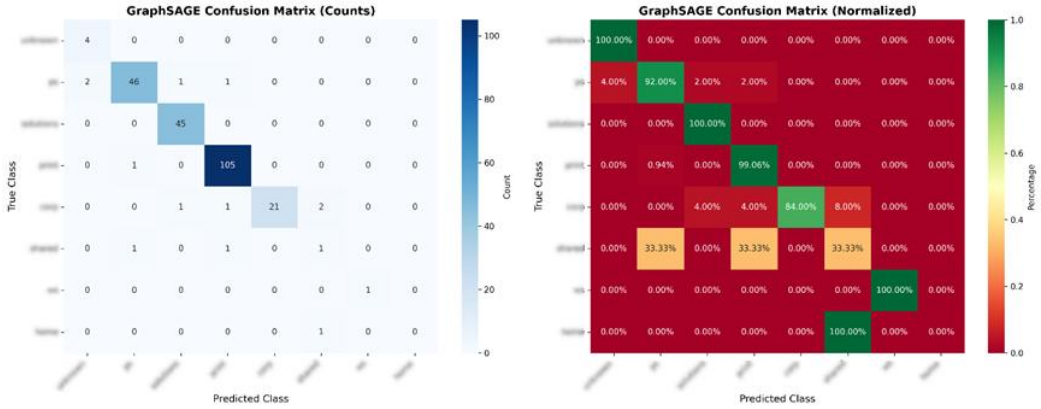


Figure 9: GraphSAGE confusion matrix

The model's errors are a useful diagnostic tool. They don't simply indicate a model failure but they pinpoint areas where the company's own information architecture has overlapping or using unclear definitions. The difficulty in classifying "shared" content provides a clear, data-driven recommendations for the enterprise to consider creating more distinct and structurally integrated landing zones for cross-functional content.

6 Discussion and conclusions

This chapter synthesizes the results presented in Chapter 5, interpreting their significance and discussing their broader implications for enterprise AI governance. We summarize the validation of our dual-track framework, explore the theoretical underpinnings of the observed "feature efficiency" principle and outline the practical applications of this work.

6.1 Key findings

This research was not a single experiment but a systematic engineering validation of a proposed "classify – validate – verify" workflow. Our three-stage experimental design led to foundational discoveries about the nature of enterprise web structure and the optimal architectures for managing it.

6.1.1 Experiment I: Validating the network-centric premise

The first critical finding was the validation of the Network Auditor. We proved that an enterprise website's network structure is not merely navigational but contains a high-fidelity signal of business logic. Models trained exclusively on 29 network topology features achieved 95.3% accuracy (Random Forest) in classifying business units. This result confirms that the site's graph structure provides a reliable "ground truth" for automated governance, capable of detecting structural anomalies without relying on potentially noisy page content.

6.1.2 Experiment II: Uncovering the limits of multi-modal data

The second major finding addressed the question of optimal feature selection. We observed counter-intuitive "efficiency principle": adding descriptive metadata (readability scores, layout counts) to the strong network signal actually reduced performance (from 92.5% to 91.9% in multi-modal tests). We also found that scalar metrics derived from text (Track A) were poor predictors (58.4% accuracy). This failure of "metadata" justified our architectural decision to use high-dimensional TF-IDF vectors for the production classifier, proving that text cannot be compressed into simple metrics without losing its semantic signal.

6.1.3 Experiment III: Solving the "cold-start" problem

Finally, we demonstrated that the deep structural intelligence validated in Experiment I could be transferred to a fast, scalable production tool. By training a Text-First Classifier (Track B) on network validated labels, we achieved 92% accuracy with lower than 100ms latency. This result proves that while scalar text features are weak (Experiment II), vectorized text features are highly predictive (Experiment III). This distinction validates our dual-track architecture: using low-

dimensional network features for auditing and high-dimensional text vectors for placement.

6.2 Practical implications

The findings of this research have significant practical implications, moving beyond academic theory to provide a tangible framework for helping large enterprises manage their digital ecosystems and prepare them for an AI-driven future. This work establishes the analytical groundwork necessary for what can be termed “AI-readiness” by unlocking three core capabilities.

6.2.1 A data-driven blueprint of the information architecture

This research proved that an enterprise’s website’s architecture is not an abstract concept but a measurable, machine-readable system with learnable rules.

Our network-centric models achieved up to 95% accuracy is classifying a page’s business function based solely on its connections. This provides enterprises with a powerful new capability, the ability to generate a quantitative, data-driven “blueprint” of their entire information architecture. Instead of relying on outdated sitemaps or manual audits, they can use this framework to see how their site is actually structured. This allows them to identify structural weaknesses, such as the 85 content islands we discovered, which sometimes represent valuable content that is poorly integrated and hard to discover and sometimes pages published live and made public by mistake (test sites, dummy page, etc.).

6.2.2 An automated system for ensuring structural coherence

The research demonstrates that the implicit rules governing the website’s organizational logic can be learned by an AI with high accuracy.

Our models achieved 92-95% accuracy across a range of classification tasks, from broad business units to more specific page functions. This proves the feasibility of creating an automated governance system. Such a system can continuously monitor the site for “structural drift”, pages that are miscategorized or become disconnected over time. For a business, this means moving from periodic,

expensive manual audits to a proactive, automated system that ensures the entire digital ecosystem remains coherent and logically structured.

6.2.3 A practical tool for real-time content governance

Our production classifier with the text-first model achieved 92% accuracy. This is a direct solution to a common business bottleneck. It enables the creation of an interactive, AI-assisted tool that can guide content creators to place new pages in the most effective location in real-time. Our technical demonstration proves this is not just a theoretical but a practical, deployable solution. This ensures that the website's structural integrity is maintained from the moment new content is created, significantly reducing the need for future clean-up and reorganization.

Together these capabilities from a comprehensive framework for transforming a large, complex website from a difficult-to-manage liability into a coherent, optimized, and AI-ready strategic asset.

6.3 The feature efficiency principle

One of the theoretically important results of this research is that we observed a machine learning principle called feature efficiency: the counter-intuitive finding that for this problem, simpler models with fewer, high-quality features consistently outperform more complex models with a larger feature set. This was not an isolated incident but a pattern that emerged across both traditional and advanced architectures.

6.3.1 The initial discovery with traditional models

During our baseline model experiments a Random Forest model trained on just 29 network features achieved a test accuracy of 92.5%. However, when we trained the same model on a comprehensive set of 135 features, its performance slightly decreases to 91.9%. Adding 106 additional features from content, layout, and other modalities resulted a net negative impact. While this 0.6 percentage point difference may seem marginal, it represents a consistent pattern across all cross-validation folds and, more importantly, demonstrates that 106 additional features requiring significant computational resources provides zero benefit.

6.3.2 Confirmation with Graph Neural Networks

The ultimate test of this principle came from our state-of-the-art GraphSAGE model. One might assume that a sophisticated GNN, specifically designed to learn complex feature interactions on a graph, would be able to filter noise and benefit from the additional features, but the opposite occurred.

- The GraphSAGE model trained on 29 network features achieved our peak accuracy of 94.89%.
- The exact same GNN architecture trained on 100 numeric features saw its accuracy fall by 4.68% to 90.21%.

This is a notable result. It proves that the principle is not an artifact of a specific model's limitations but a fundamental property of the problem itself. It provides the strongest possible evidence that for classifying organizations function, network topology is the dominant signal, and other features actively hinder performance, even for advanced architectures.

6.3.3 Theoretical explanations of this principle

Our analysis suggests three complementary theoretical reasons for this phenomenon:

1. **Signal to noise theory:** the network features provide an exceptionally strong and clean signal. Our analysis showed that the network-centric model (92.5% CV mean) significantly outperformed the text-only model (58.4% CV mean). When the primary signal is this dominant, adding weaker signals introduces more statistical noise than valuable information, forcing a model to learn to ignore irrelevant data and increasing the risk of overfitting.
2. **Feature redundancy and information overlap:** the network structure already implicitly captures most of the information present in the other feature families. As validated by the 94.2% homogeneity between detected communities and business units, a page's network position is a powerful proxy for its function. According to Conway's Law [20] organizational structure dictates technical structure; therefore, a page's content and layout

are often a result of its position in a network, not an independent signal. Adding these downstream features creates information redundancy without adding new, orthogonal insights.

3. **The "curse of dimensionality":** As shown in the ablation study, performance peaks with a curated set of features and then declines as more, less-relevant features are added. For both the traditional ML models, and the more advanced GNNs the optional set was around 29 features. This proving that quality is more important than quantity regardless of architecture. Beyond this optimal point, the models suffer from the curse of dimensionality, where the feature space becomes too vast and sparse, making it harder to learn the true signal.

6.3.4 Practical implications

Rather than viewing this as a limitation, the feature efficiency principle provides valuable guidance for real-world system design. Our research demonstrates that when a single feature family (like network topology) is overwhelmingly predictive, the best strategy is often to isolate that signal rather than dilute it with weaker indicators.

This finding feature challenges the common assumption that "more data is always better". Instead, it validates the focused approach: identifying the dominant signal for the specific problem and optimize for it. In our case, network structure proved so informative about organizational function that additional features only added noise.

This approach yielded concrete benefits in our production system: faster training times, simpler maintenance, lower computational costs, and paradoxically, better performance. By embracing the feature efficiency principle, we achieved both technical and practical deployability.

6.4 Study limitations and future work

Before discussing limitations, it's important to note the extensive methodological safeguards put in place to ensure the validity of our results. We proactively addressed several potential threats to internal validity:

- **Label-feature independence:** to prevent circular reasoning, our target labels were sourced from Fortune 500 Enterprise's independent CMS taxonomy, while our network features were derived solely from the site's link topology.
- **Leakage prevention:** we excluded temporal features, prevented URL token leakage in our text-first model, and used a single, fixed test set across all four experimental phases to ensure fair and unbiased comparisons.
- **Tautology avoidance:** community membership, while used to validate the quality of our labels (showing 94.2% homogeneity), was not used as a predictive feature in our network models to avoid tautological conclusions.

These measures provide a strong foundation for the integrity of our findings. Nevertheless, the study has several inherent limitations.

6.4.1 Study limitations

1. **Single domain focus:** The primary limitation of this research is that its finding based on the analysis of a single enterprise website. While this site is large and diverse, the "network-dominant" signal we observed may be characteristic of organizations with a highly structured information architecture. The findings may not generalize directly to websites with flatter, less hierarchical structures.
2. **Temporal scope:** our analysis is cross-sectional, representing a snapshot of the website at a single point in time. It does not capture the evolution of the site's structure, the decay of certain content areas, or the impact of major redesigns over time.
3. **Feature and model scope:** the feature engineering process, while comprehensive, relied on a curated set of features rather than a fully automated discovery. Furthermore, while our models performed exceptionally well, we did not perform an exhaustive casual analysis to

prove that network position causes better performance, only that it is strongly correlated.

6.4.2 Future research directions

The limitations of the current study directly inform a rich agenda for future work. The framework developed in this thesis serves as the foundational groundwork for several exciting research avenues including:

- **Cross domain validation:** the most critical next step is to apply this network-centric methodology to other large enterprise websites to test the generalizability of the network dominance principle. Our theory is that the proved signal dominance is universal and it can be observed on different enterprise websites from similar (IT, tech) industry but it would be interesting to see how it changes on a website from a way different industry, and how the size of the enterprise could impact that signal strength.
- **Temporal network analysis:** future work should incorporate a time-series dimension, analyzing multiple snapshots of the website to model its evolution, detect "structural drift", and predict which content areas are growing or decaying in importance. In our current research even if we captured timestamps for the data, we excluded that from the final feature set to reduce the complexity and drive the focus to prove the core research question, but this data would be essential for a future AI-driven website managing agent.
- **Integrating with user behavior:** major extension would be to integrate user engagement metrics (e.g. click-through rates, time on page, etc.) into the network model. This would allow us to move from analyzing intended structure to understanding the effective structure as experienced by users. With this fundamental improvement we can shift from the classic webpage analytics paradigm where we focus on performance of specific page, or page groups, with a network-centric view where we treat the website as a network where neighbors have influence on each other.
- **Integrating embeddings:** experimenting with more advanced embedding techniques is also a promising next step. In our current research we used

manually defined metrics to describe features (e.g., PageRank, Betweenness) but embeddings let the models to learn these latent structural relationships directly without that predefined "vocabulary". We can combine LLM and Network embeddings to create a hybrid model that does not require our hand-crafted feature engineering which was one of the main limitations of the current research.

- **From analysis to generation:** this thesis deliberately focused on establishing the analytical foundation. The logical next step is to use these insights to build generative systems. This includes using the network-validated patterns to guide AI in generating not just page placements, but also optimal page layouts and even the content itself.

6.5 Conclusion

This thesis addressed the critical challenge of preparing large, complex enterprise websites for an AI-driven future by asking a fundamental question: What is the most reliable signal for understanding a site's business logic? We uncovered a definitive answer: the network.

Our research validated a dual-track system that leverages this insight. We proved that network topology provides a 95% accurate "ground truth" for auditing, while high-dimensional texts allow for 92% accurate "cold-start" placement. Crucially we established the feature efficiency principle, demonstrating that for structural analysis, a focused set of 29 topological features outperforms complex multi-modal datasets- even when using state-of-the-art GNNs.

Ultimately, this work provides a blueprint for the "self-organizing enterprise website". By enabling AI to classify new content, place it within the structure and verify its integrity, we transform the corporate website from an unmanageable liability into a coherent, AI-ready asset.

References

[1] Nielsen Norman Group. "Ecommerce UX Research: Findings and Methodology." Accessed 27 Oct 2025. https://www.nngroup.com/reports/ecommerce-user-experience/

[2] Baymard Institute. "48 Cart Abandonment Rate Statistics 2024." Accessed 27 Oct 2025. https://baymard.com/lists/cart-abandonment-rate

[3] Nielsen, J. "How Users Read on the Web." Nielsen Norman Group, 1 Oct 1997. Accessed 27 Oct 2025. https://www.nngroup.com/articles/how-users-read-on-the-web/

[4] Google. "Find Out How You Stack Up to New Industry Benchmarks for Mobile Page Speed." Think with Google, Feb 2017. Accessed 27 Oct 2025.

[5] L. Page and S. Brin, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, Vol. 30, Issues 1–7, pp. 107–117, 1998.

[6] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, Vol. 46, No. 5, pp. 604–632, 1999.

[7] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, Vol. 286, Issue 5439, pp. 509–512, 1999.

[8] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, Vol. 45, No. 2, pp. 167–256, 2003.

[9] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences (PNAS)*, Vol. 99, No. 12, pp. 7821–7826, 2002.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2008, No. 10, P10008, 2008.

[11] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: guaranteeing well-connected communities," *Scientific Reports*, Vol. 9, Article 5233, 2019.

[12] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, Vol. 659, pp. 1–44, 2016.

[13] D. K. Panda and S. Ray, "Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review," *Journal of Intelligent Information Systems*, Vol. 59, pp. 341–366, 2022.

[14] J. M. Zawia et al., "Comprehensive Review of Meta-Learning Methods for Cold-Start Issue in Recommendation Systems," *IEEE Access*, Vol. 12, 2024.

[15] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations (ICLR)*, 2017.

[17] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30, 2017.

[18] W. Jiang, "Graph-based deep learning for communication networks: A survey," *Computer Networks*, Vol. 197, 2021.

[19] M. Hilb, "Artificial intelligence and corporate governance: A review of recent literature," *International Journal of Strategic Management*, 2024.

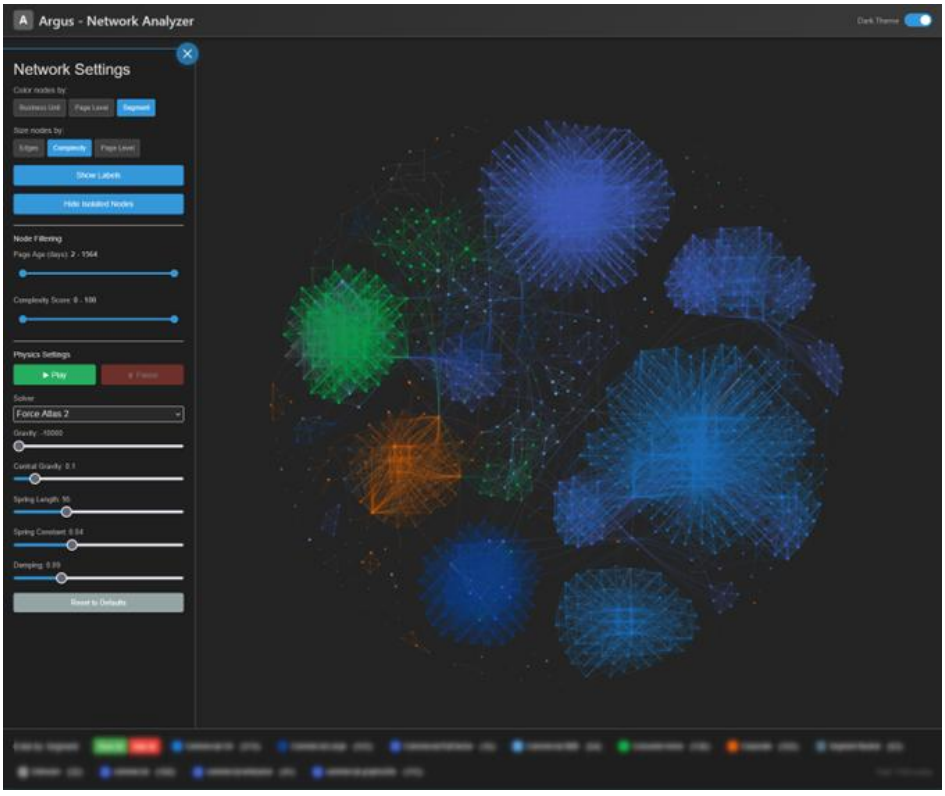
[20] M. E. Conway, "How Do Committees Invent?" *Datamation*, Vol. 14, No. 5, pp. 28–31, 1968.

[21] S. Fortunato and M. E. J. Newman, "20 years of network community detection," *Nature Physics*, Vol. 18, pp. 848–850, 2022.

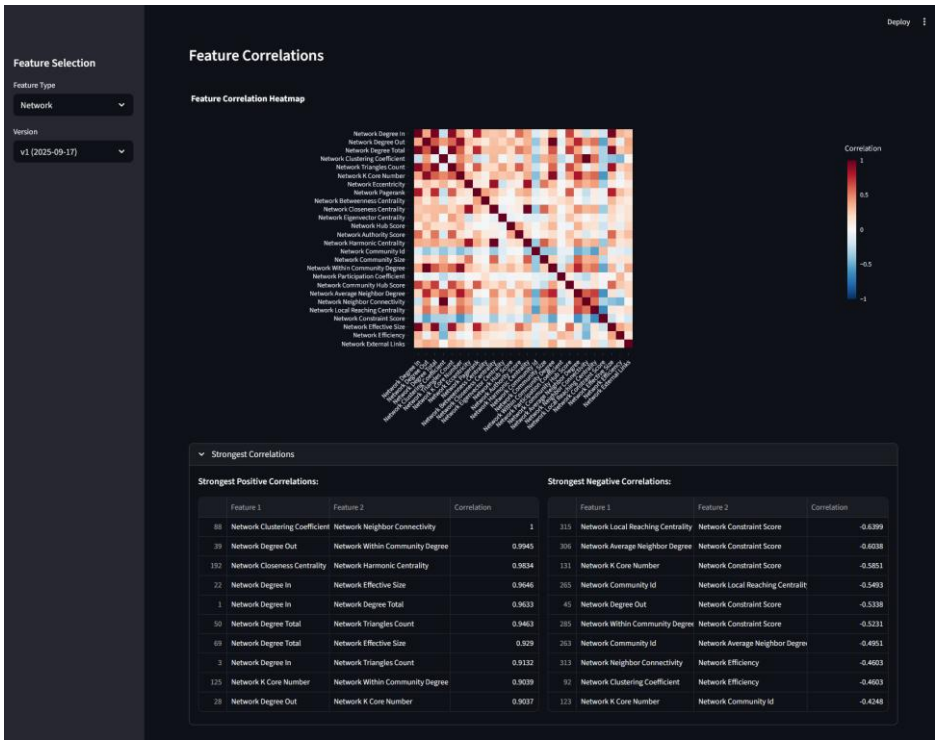
[22] P. M. C. Swagatam, "AI governance: a systematic literature review," *ResearchGate*, 2025.

Appendix

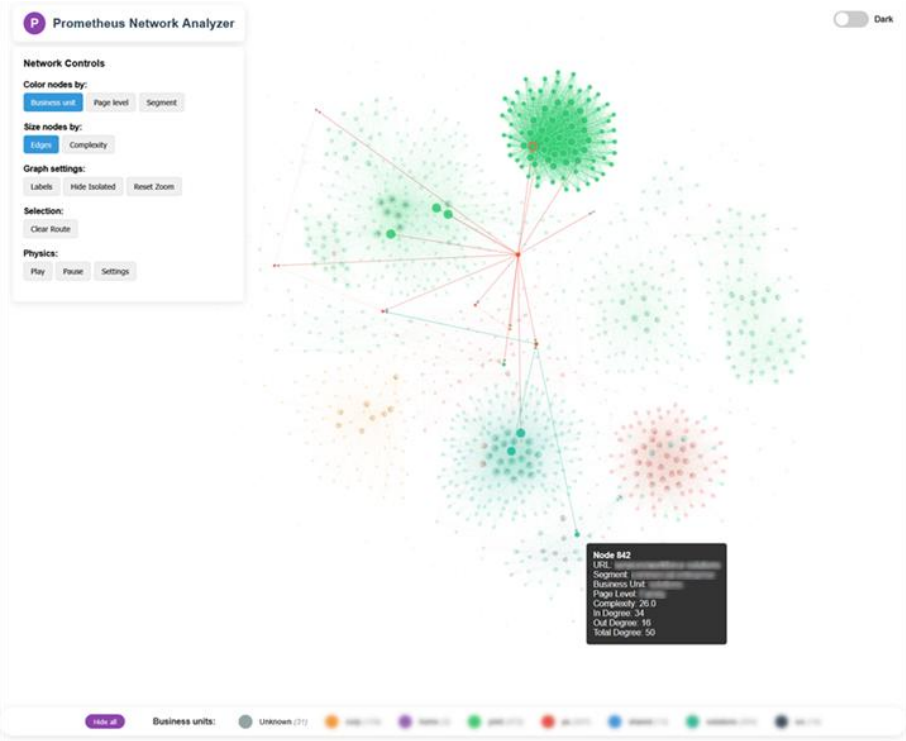
A1: The user interface of Argus web scraper – the custom Python web scraper used to collect the data for the research



A2: Argus Network Analyzer - This custom visualization tool used in early stage of the research in the EDA phase



A3: Dashboard for the extracted features to shop correlation heatmaps, box plot with outliners, etc.



A5: Prometheus Network Analyzer - this is the visualization engine used and embedded in our production system

Acknowledgements

To my wife, Diana, thank you for your unwavering support and patience throughout this long journey; this achievement is as much yours as it is mine. To my son, Gabriel, who joined us between the writing of the second and third chapters; You provided the most beautiful perspective and the greatest motivation to cross the finish line.