

ProofLayer: Control Coverage Screening for Government Cybersecurity Policies via Multi-LLM Orchestration

Devharsh Trivedi*

City University of New York, NY, USA

Abstract: ProofLayer is a two-stage system for screening government cybersecurity policy documents against control catalogues: a keyword stage that maps policy text to a twenty-two control set crosswalked to NIST SP 800-53 Revision 5, and a language-model stage intended to resolve the cases keyword matching cannot, with a cost-aware orchestration layer routing between models. This paper evaluates the first stage on ten Maryland state and local government policy documents and separates what is measured from what is only described. Reading each document in full gives a mean control coverage of 62.8 percent (standard deviation 24.1). An earlier analysis of this corpus truncated each document at fifteen pages and reported 49.1 percent; seven of the ten documents exceed fifteen pages, and the two longest gain 36 and 41 points when read in full. Both figures are reported, because a screening metric that moves this far with extraction depth is itself informative. Coverage correlates with document length at $r = 0.690$ against log word count, so 48 percent of its variance is explained by length alone. Of 138 covered document-control pairs, 18.1 percent carry a negation cue within 130 characters of the match and 44.2 percent rest on a single keyword out of a mean 6.1 available. A worked error analysis gives four cases, among them a footnote URL scored as transport encryption and an incident-reporting requirement carrying a two-business-day deadline scored as absent because the keyword list lacked the word *notifying*. The second-stage classifier is described but not evaluated: establishing whether it resolves these failures requires expert-annotated ground truth for this corpus, which does not yet exist, and model-generated labels are declined as a substitute. The orchestration cost saving is a token-price projection rather than instrumented spend. We therefore report policy-vocabulary coverage, not compliance, and release all 220 document-control pairs with matched keywords and text snippets.

Keywords: Cybersecurity policy analysis; control coverage screening; NIST SP 800-53; government cybersecurity; keyword matching; measurement validity; large language models; Maryland.

INTRODUCTION

State and local governments are increasingly targeted by sophisticated cyberattacks. Between 2022 and 2025, over 1,800 reported breaches affected public and private organizations in the United States, with ransomware attacks on Baltimore City costing an estimated \$18 million and similar incidents shutting down services across Maryland school systems and healthcare networks [1]. Despite growing awareness of these threats, a 2021 analysis by the Maryland Cybersecurity Council found that 53.8% of local emergency management offices lacked a consequence management plan, and only 31% of local school systems reported adequate cybersecurity budget allocations [2].

Federal compliance frameworks such as NIST Special Publication 800-53 Revision 5 [3], the Federal Risk and Authorization Management Program (FedRAMP) [4], and the Cybersecurity Maturity Model Certification (CMMC) [5] provide structured sets of security controls for assessing organizational cybersecurity posture. However, mapping government policy documents to these frameworks manually is prohibitively expensive: independent assessments for FedRAMP authorization can cost between \$50,000

and \$500,000 and take 6–18 months to complete [4]. This cost burden falls disproportionately on smaller agencies and municipalities that have limited IT security budgets.

Recent advances in large language models (LLMs) open a new possibility: automating the reading, interpretation, and gap analysis of free-text policy documents against structured control frameworks. Prior work has demonstrated LLM applicability to legal compliance analysis [6], post-incident policy gap detection [7], multi-framework regulatory gap detection [8], and compliance automation in financial services [9]. However, no prior work has applied LLM-based policy screening specifically to state and local government cybersecurity policy documents at scale, nor has the cost efficiency of a multi-LLM routing strategy been evaluated in this domain. This paper addresses the first gap only, and for the keyword stage alone; the routing strategy is described here but its cost efficiency remains unevaluated.

This paper makes one evaluated contribution and two descriptive ones, and we distinguish them explicitly because only the first rests on measurement.

1. **Evaluated.** An **empirical keyword-based control coverage mapping** of ten Maryland state and local government cybersecurity policy documents against 22 controls crosswalked to

*Address correspondence to this author at the City University of New York, NY, USA; E-mail: DEVHARSH.TRIVEDI86@lagcc.cuny.edu

13 NIST SP 800-53 Revision 5 families, reporting a mean coverage of 62.8% on full text, together with a characterisation of how far that figure is driven by document length, negation and single-keyword matches, and a worked error analysis. This is the result the paper stands on.

2. **Described, not evaluated.** The design and implementation of **ProofLayer**, a policy screening pipeline that ingests policy documents and produces structured audit reports. Its second stage applies an LLM to gap analysis and remediation guidance. That stage is implemented and specified here, but it has not been evaluated against expert annotation, so no accuracy claim is made for it. Section 6.3 states the annotation protocol that would settle the question.
3. **Projected, not measured.** A **cost-aware multi-LLM routing strategy** that directs cheaper models to extraction and stronger models to reasoning. An arithmetic model over published vendor rates projects roughly 24% lower per-audit inference cost than using a single high-capability model throughout. This is a projection from list prices, not an observed saving, and it inherits whatever error the underlying token estimates carry.

Separating these matters for how the paper should be read. A two-stage system whose second stage is unevaluated is, evidentially, a one-stage study with an architecture proposal attached, and we present it as such rather than letting the system diagram carry an implied claim.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the ProofLayer system architecture. Section 4 presents our methodology. Section 5 reports experimental results on the Maryland policy corpus. Section 6 discusses findings and limitations. Section 7 concludes.

BACKGROUND AND RELATED WORK

This section provides the context needed to situate ProofLayer within existing work. We first survey the federal and state cybersecurity compliance frameworks against which policy documents are audited, then review recent applications of LLMs to regulatory and compliance analysis, and finally examine the emerging literature on multi-LLM orchestration for cost-aware AI pipelines.

Cybersecurity Compliance Frameworks

NIST SP 800-53 Rev 5 [3] is the canonical reference for federal information system security controls, defining over 1,000 security controls and control enhancements across 20 families covering areas from access control (AC) and audit accountability (AU) to system and communications protection (SC). FedRAMP builds on NIST 800-53 to specify cloud security requirements for federal agency cloud services [4]. CMMC extends this framework to DoD supply chain contractors, defining three maturity levels comprising 110 practices mapped from NIST SP 800-171 [5].

Maryland has aligned its state Minimum Cybersecurity Standards to the NIST Cybersecurity Framework [10], and since 2022 has required all units of local government to certify compliance with state minimum standards under SB754 of 2022 [2]. Nevertheless, the 2021 Maryland Cybersecurity Council analysis found wide variation in policy coverage and implementation across agencies, particularly at the local government level.

LLMs for Compliance and Regulatory Analysis

The intersection of natural language processing and regulatory compliance has been an active research area. Early work used rule-based and classical machine learning methods to extract compliance obligations from regulatory text [8]. More recently, LLMs have shown strong performance on compliance-related tasks including legal document interpretation, privacy policy analysis, and policy gap detection [11]. Salman *et al.* [12] specifically examine LLMs for cybersecurity compliance, showing that continuous LLM-based monitoring can reduce reliance on expensive periodic audits.

Hassani *et al.* [6] argue that LLMs enable a shift from sentence-level classification to end-to-end compliance reasoning, addressing a key limitation of prior NLP approaches that processed policy text in isolated fragments. Their work on legal compliance automation at the IEEE Requirements Engineering conference provides a framework for thinking about how LLMs can handle the full context of a regulatory document. Independently, the “Automated Post-Incident Policy Gap Analysis” system by Oh *et al.* [7] demonstrates an LLM-driven multi-agent workflow that maps system event log evidence to MITRE ATT&CK entries and then evaluates organizational policies for adequacy, using GPT-4o and LangGraph. ComplianceNLP [8] combines knowledge-graph-augmented retrieval-augmented generation (RAG)

with multi-task obligation extraction to achieve 87.7 F1 on gap detection across SEC, MiFID II, and Basel III regulatory frameworks, outperforming GPT-4o+RAG baselines by 3.5 F1 points.

Our work differs from the above in three ways: (i) we target government cybersecurity policy documents rather than financial regulations; (ii) we apply keyword-based Stage 1 mapping as a no-API-cost baseline before engaging LLM reasoning; and (iii) we evaluate the cost efficiency of multi-model routing, a concern that prior work has not addressed in the compliance context.

Multi-LLM Orchestration

The idea of routing tasks to different LLMs based on complexity and cost is an emerging research direction. xRouter [13] trains a reinforcement learning router that routes requests to cost-appropriate models under explicit economic constraints, achieving strong Pareto-optimal operating points on cost-accuracy tradeoffs. Benchmarking by Kulkarni and Kulkarni [14] compares sequential, parallel, hierarchical, and reflexive multi-agent architectures for financial document processing, finding that reflexive architectures achieve the highest field-level accuracy but at 2.3× the cost of sequential baselines. ProofLayer applies analogous routing principles to the compliance auditing domain: lightweight models for pattern recognition and text extraction, stronger models for reasoning-intensive gap analysis and report generation.

SYSTEM ARCHITECTURE

ProofLayer system architecture. Policy documents enter through the ingestion layer, are processed by a two-stage compliance mapping pipeline, and routed to

cost-appropriate LLMs for gap analysis and report generation. Source: generated by the authors from the analysis pipeline described in Section 4.1.

Document Ingestion

ProofLayer accepts policy documents in PDF or plain text format. PDF text is extracted using pdfplumber [15], which handles multi-column layouts common in government documents. Extracted text is stored in memory with a unique document ID for subsequent processing. Documents exceeding 15,000 characters are flagged for routing to claude-sonnet, which offers strong long-context performance at competitive cost relative to token count. Character count serves as a practical proxy for token count (approximately 4 characters per English token); at this threshold a document’s content occupies roughly 3,750 tokens, which approaches the practical input budget available to gpt-4o after accounting for prompt template overhead, making the larger effective context window of claude-sonnet beneficial for full-document coverage.

AI Extraction

An optional AI extraction pass uses gpt-4o-mini to normalize and summarize security-relevant content from the raw document text, producing a condensed representation of security practices, policies, and procedures. This step costs approximately \$0.003 per document (under 2,000 input tokens) and reduces noise for downstream processing.

Control Mapping Engine

The control mapping engine is the analytical core of ProofLayer. It implements a deliberate two-stage pipeline that separates low-cost pattern recognition

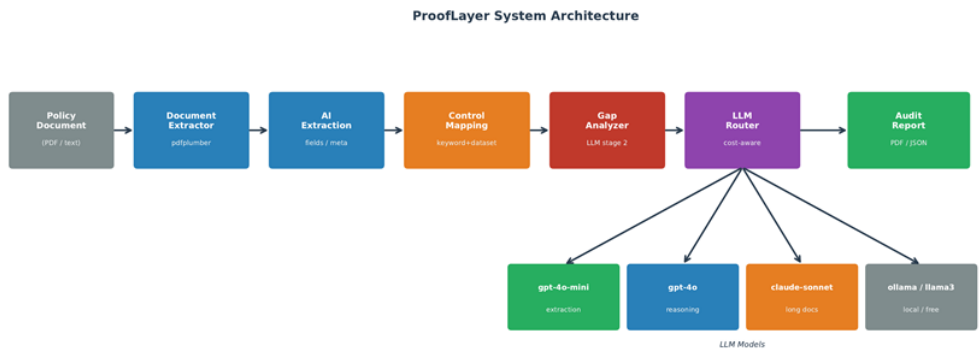


Figure 1: Illustrates the ProofLayer architecture, which consists of five components: a document ingestion layer, an AI extraction module, a control mapping engine, a multi-LLM orchestration layer, and an output generation module.

from high-cost contextual reasoning, allowing the system to scale economically to large document corpora while preserving analytical depth for controls that require it.

Stage 1 – Keyword Matching. Each document is scanned for occurrence of framework-specific keywords associated with each control in the dataset. This stage requires no LLM API calls, runs in under 50 milliseconds per document, and incurs zero LLM inference cost. Controls are marked as “present” if any of their associated keywords appear in the document text.

Stage 2 – LLM Gap Analysis. A structured prompt containing the document summary and the full control list is sent to a reasoning-capable LLM (GPT-4o by default, with optional routing to Claude Sonnet for long documents or Ollama/LLaMA-3 for free local operation). The model is asked to classify each control as *verified*, *partial*, or *missing* and to provide an explanatory note. The JSON response is parsed and merged with Stage 1 results.

Control Dataset

The ProofLayer control dataset contains 22 controls spanning 13 NIST SP 800-53 Rev 5 control families organized into ten category groupings, cross-referenced to their FedRAMP, CMMC 2.0, and SOC2 equivalents. Each control record includes a human-readable name, a set of associated keywords, the applicable framework mappings, a risk level (HIGH or MEDIUM), and a remediation recommendation. Table 1 summarizes the control families included.

Multi-LLM Orchestration Layer

ProofLayer implements a cost-aware task routing strategy shown in Table 2. The routing decision is made per task type and optionally per content length, directing extraction and keyword-mapping tasks to inexpensive small models and reserving high-capability models for reasoning-intensive gap analysis and report generation. When internet connectivity is unavailable or API budget constraints are binding, the system falls back to a locally hosted Ollama/LLaMA-3 instance,

Table 1: The complete ProofLayer control set and its verified crosswalk. Internal identifiers are PL-nn and carry no meaning outside this system. A dagger marks the sixteen controls whose previous internal label would have been read as a NIST identifier denoting a different control: the old AC-04, for instance, mapped to NIST IA-2, not AC-4. All twenty-two NIST SP 800-53 Revision 5 mappings below were checked individually. Risk column: H/M/L. The machine-readable crosswalk, including CMMC 2.0 and SOC 2 columns and the keyword lists, is released as data/control_crosswalk. {json,csv}.

ID	Control	NIST r5	FedRAMP	R
PL-01	Access Control Policy & Procedures	AC-1	AC-1	H
PL-02	Account Management	AC-2	AC-2	H
PL-03	Least Privilege / Role-Based Access	AC-6†	AC-6	H
PL-04	Multi-Factor Authentication	IA-2†	IA-2(1)	H
PL-05	Remote Access Controls	AC-17†	AC-17	M
PL-06	Configuration Management Policy	CM-1	CM-1	H
PL-07	Change Management / Change Control	CM-3†	CM-3	M
PL-08	Audit Logging & Monitoring	AU-2†	AU-2	H
PL-09	Audit Review & Alerting	AU-6†	AU-6	H
PL-10	Incident Response Plan	IR-1	IR-1	H
PL-11	Incident Reporting	IR-6†	IR-6	H
PL-12	Risk Assessment	RA-3†	RA-3	H
PL-13	Vulnerability Scanning	RA-5†	RA-5	H
PL-14	Data Encryption in Transit	SC-8†	SC-8	H
PL-15	Data Encryption at Rest	SC-28†	SC-28	H
PL-16	Network Segmentation	SC-7†	SC-7	M
PL-17	Security Awareness Training	AT-2†	AT-2	M
PL-18	Personnel Screening / Background Che.	PS-3†	PS-3	M
PL-19	Backup & Recovery	CP-9†	CP-9	H
PL-20	Contingency / Disaster Recovery Plan	CP-2	CP-2	H
PL-21	Third-Party / Vendor Risk Management	SR-1	SA-9	M
PL-22	Media Sanitization	MP-6†	MP-6	M

Table 2: ProofLayer multi-LLM routing strategy.

Task	Model	Approx. Cost/1K tok.	Rationale
Doc. Extraction	gpt-4o-mini	\$0.00015	Fast, cheap
Keyword/Sem. Mapping	gpt-4o-mini	\$0.00015	Sufficient for matching
Gap Analysis	gpt-4o	\$0.005	Needs strong reasoning
Report Generation	gpt-4o	\$0.005	Structured output required
Long docs (>15K ch.)	claude-sonnet	\$0.003	Strong long-context
Offline/Free mode	ollama/llama3	\$0.000	Local, no API cost

ensuring that auditing capability remains fully functional in air-gapped or cost-constrained environments without any code changes.

Output Generation

ProofLayer generates three types of outputs: (i) a structured JSON audit result containing per-control status, framework scores, and gap details; (ii) an LLM-generated narrative executive summary suitable for submission to an ISSO or inspector general; and (iii) a downloadable PDF audit report produced using ReportLab [16], containing score summaries, framework breakdowns, gap tables, and remediation guidance. An API endpoint at /fix additionally generates full policy document language for missing controls on demand, enabling agencies to obtain draft policy text as a remediation starting point without requiring specialized compliance writing expertise.

MATERIALS AND METHODS

This section describes the evaluation design used to assess ProofLayer on real Maryland government cybersecurity policy documents. We detail the composition of the document corpus, the keyword-

based Stage 1 mapping procedure applied in this study, the compliance scoring formula, and the cost estimation approach used to compare multi-LLM routing against a single-model baseline.

Policy Document Corpus

We evaluate ProofLayer on ten Maryland government cybersecurity policy documents listed in Table 3. Documents were selected according to three criteria: (i) publicly available from official Maryland government sources, (ii) primary focus on cybersecurity or information security policy content, and (iii) coverage of a distinct agency type or function. To the best of our knowledge, no publicly available Maryland cybersecurity policy documents satisfying these criteria were excluded. The corpus spans major branches and agencies including executive branch IT policy (DoIT), judicial branch, education (MSDE), health (DHMH), and local government analysis (UMGC/Maryland Cybersecurity Council). The corpus represents publicly available Maryland policy documents and is not intended to be statistically representative of all government cybersecurity policies.

Relationship to prior work on the same jurisdiction. These ten documents are the cybersecurity-policy

Table 3: Maryland government cybersecurity policy document corpus.

Document Label	Document	Size (chars)
MD Risk Mgmt Policy	MD-POL-201-01 Cyber Risk Mgmt Policy (2026)	17,572
MD System & Network Security	MD-POL-207-01 System & Network Policy (2026)	11,110
MD Continuous Monitoring	MD Continuous Monitoring Policy (2026)	11,413
MD Cyber Council Report 2025	MD Cybersecurity Council Biennial Report (2025)	41,422
MD IT Security Manual	MD IT Security Manual (2019, amended)	26,527
MD Local Gov Cyber 2021	MD State & Local Gov Cybersecurity Analysis (2021)	36,953
MD Health Dept IT Security	MD DHMH IT Security Policy	35,017
MD Judicial Info Security	MD Judicial Branch Information Security Policy	44,567
MD MSDE AUP 2024	MD MSDE Acceptable Use Policy (2024)	33,774
MD Procurement Manual	MD Procurement Manual (Cyber-relevant sections)	24,399

subset of a larger 51-document Maryland corpus assembled for an earlier study of document-level classification [17]. That study asked a different question at a different granularity: whether a given document is cybersecurity-relevant at all, evaluated against the six core functions of the NIST Cybersecurity Framework 2.0, comparing keyword matching to LLM-assisted classification. It found keyword matching to have high recall but poor precision at the document level. The present study takes the ten documents that survive that relevance filter and asks a control-level question instead: which of 22 specific NIST SP 800-53 controls each document addresses. The frameworks differ (CSF 2.0 against SP 800-53 Rev 5), the unit of analysis differs (document against document-control pair), and no result from the earlier study is restated here as new. We note the overlap explicitly because the corpora share a jurisdiction and provenance, and because the earlier finding that keyword matching is imprecise at the document level is part of why the present control-level results are framed as indicators of likely gaps rather than confirmed deficiencies.

Stage 1: Keyword-Based Control Coverage Mapping

For the empirical evaluation reported in this paper, we apply the Stage 1 keyword-based control mapping pipeline exclusively, since it provides a reproducible, deterministic baseline that requires no LLM API keys and enables full replication of results. The 22-control keyword dictionary was developed by the first author by mapping each control's formal NIST definition and associated assessment procedures to domain-specific keyword phrases (e.g., "encryption at rest," "disk encryption," "aes," "data at rest" for PL-15); each control maps to between 3 and 15 keywords. The full keyword dictionary is provided in the supplementary notebook. A control is marked as *present* if any associated keyword appears (case-insensitively) in the document text. Prompt lengths were estimated from representative policy documents in the corpus.

Scoring

To produce comparable, interpretable results across documents of varying length and scope, we assign each document a single control coverage score derived from the fraction of the 22 ProofLayer controls that are marked present. This score is then mapped to a risk tier consistent with NIST maturity guidance. The overall compliance coverage score for each document is computed as:

$$S = \left\lfloor \frac{|\text{matched controls}|}{|\text{total controls}|} \times 100 \right\rfloor$$

Documents are classified as **LOW** risk ($S \geq 80$), **MEDIUM** risk ($55 \leq S < 80$), or **HIGH** risk ($S < 55$) based on heuristic thresholds calibrated to approximate NIST maturity guidance; these specific percentage values are not prescribed by NIST SP 800-53 directly. The gap frequency for each control is the number of documents in which that control is absent, providing a measure of systemic weakness across the policy corpus.

Cost Estimation (projected, not measured)

LLM inference costs are estimated using published per-thousand-token pricing for each model as of mid-2025. For a standard audit (document extraction, keyword mapping, LLM gap analysis, and narrative report generation), we compare the ProofLayer multi-LLM strategy against a single-model baseline in which all tasks are routed to gpt-4o. Estimated input and output token counts are based on ProofLayer's production prompt templates and typical policy document lengths (extraction: 2,000 input / 400 output tokens; mapping: 1,500 / 300; gap analysis: 3,000 / 700; report: 2,500 / 800; long-document routing via claude-sonnet: 6,700 / 1,000). These are model inputs, not measured values. Actual costs will vary proportionally with document length. The full keyword dictionary, routing algorithm, cost model, and analysis code are provided in the supplementary Jupyter notebook.

RESULTS

All numbers below are computed from the full text of each document. The released files `results/coverage_by_pair.csv`, `coverage_by_document.csv` and `coverage_by_control.csv` regenerate every figure and table in this section, and `code/run_coverage.py` regenerates those files from the corpus.

A correction to the earlier analysis of this corpus

An earlier analysis of this corpus extracted text with a fifteen-page cap. Seven of the ten documents are longer than fifteen pages; the Maryland IT Security Manual is 219 pages, so its score was computed on roughly 7% of its text. Re-running the identical matching logic under that cap reproduces the earlier corpus mean of 49.1% exactly, which confirms the cause. Reading in full gives 62.8%.

Table 4: Effect of the fifteen-page cap. Documents at or under fifteen pages are unaffected and are omitted.

Document	Pages	First 15 pp	Full	Δ
MD Health Dept IT Security Policy	122	54%	95%	+41
MD IT Security Manual	219	59%	95%	+36
MD Local Gov Cybersecurity 2021	57	54%	77%	+23
MD Procurement Manual	90	22%	36%	+14
MD Cybersecurity Council Report 2025	30	50%	63%	+13
MD MSDE Acceptable Use Policy 2024	17	40%	45%	+5
Corpus mean		49.1%	62.8%	+13.7

We report this rather than silently substituting the corrected figures. A screening metric whose value moves by up to 41 points depending on how much of a document the extractor reads is telling the reader something about the metric, and a practitioner deploying such a tool needs to know it.

Coverage across the corpus

Mean coverage is 62.8% (SD 24.1, range 36.4 to 95.5). Under the threshold scheme of Section 4.3, three documents fall in the LOW band ($\geq 80\%$), two in MEDIUM (55–79%) and five in HIGH ($< 55\%$). Figure 2 plots full-text and truncated scores side by side.

Four of the twenty-two controls appear in all ten documents. The most frequently absent, all of them rated HIGH risk in our control set, are access control policy (PL-01, NIST AC-1) and incident reporting (PL-

11, NIST IR-6) at seven of ten documents each, encryption at rest (PL-15, NIST SC-28) also at seven, and account management (PL-02, NIST AC-2) and audit review (PL-09, NIST AU-6) at six. Section 5.4 shows that at least one of the PL-11 absences is a matcher failure rather than a policy gap.

What the coverage metric tracks

Three measurements bound the interpretation of the numbers above.

Length. Coverage correlates with document length at $r = 0.690$ against log word count across the ten documents, so $r^2 = 0.476$: nearly half the variance in the headline metric is explained by how long a document is rather than by what it contains (Figure 3). The clearest illustration is the Procurement Manual, which at 25,289 words scores the same 36.4% as a

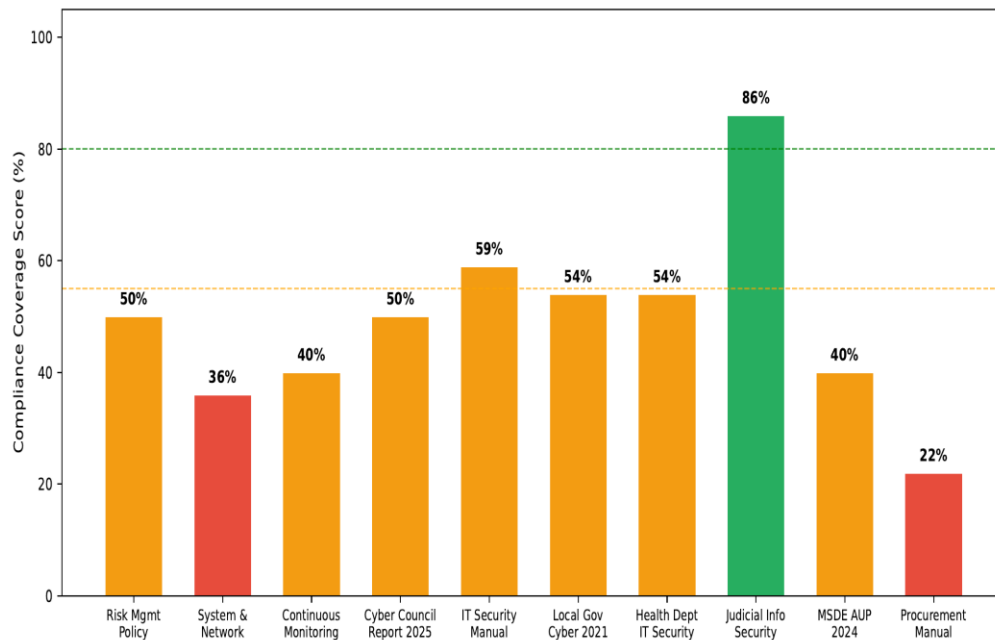


Figure 2: Control coverage per document, full text against the first fifteen pages. Green, orange and red mark the LOW, MEDIUM and HIGH bands of Section 4.3. The six documents with a visible grey bar are those longer than fifteen pages.

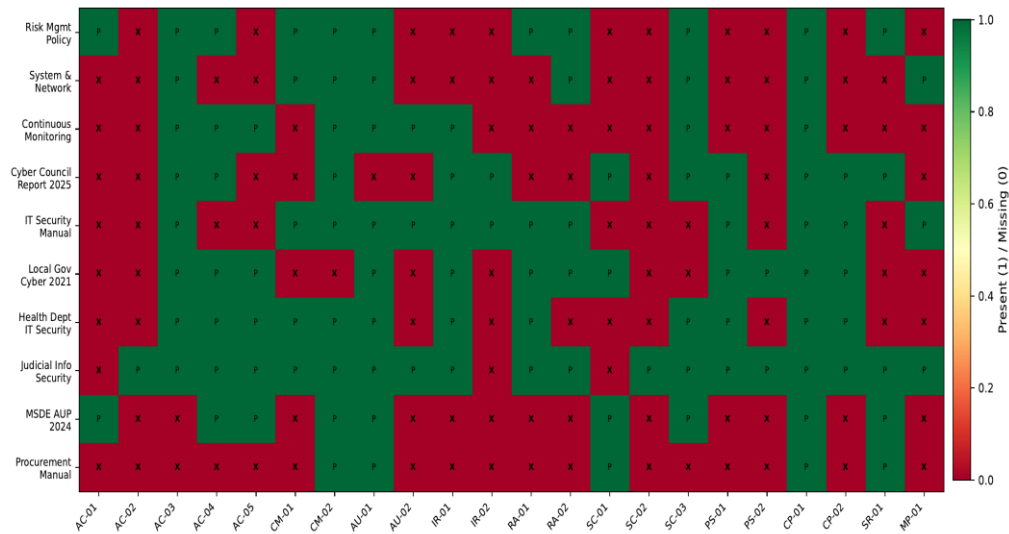


Figure 3: Coverage against document length. The fitted line is a least-squares fit on log word count, $r = 0.690$, $r^2 = 0.476$.

1,423-word network security policy. A long document about the wrong subject and a short document about the right one are not distinguished.

Negation. Of the 138 covered document-control pairs, 25 (18.1%) contain a negation cue such as “does not”, “not required” or “except” within 130 characters of the matched keyword. Lexical matching cannot tell an assertion from its denial, and roughly one covered pair in six sits next to language that may reverse it.

Thinness of evidence. 61 of the 138 covered pairs (44.2%) are covered on exactly one matched keyword, with a mean of 1.97 keywords matched out of 6.1 available. Coverage is frequently a single word.

Error analysis

No expert-annotated ground truth exists for this corpus, so we report no precision, recall or agreement statistic. What follows are cases identified by inspecting matched keywords in context, each traceable to a released row.

A URL scored as transport encryption. PL-14 (NIST SC-8) is scored present in the Cybersecurity Council Report on the keyword `https`, matching a footnote citation to `umgc.edu`. The document is citing a web page, not asserting a control.

Organisational roles scored as role-based access control. PL-03 (NIST AC-6) is scored present in two documents on roles, matching “the department responsible for roles: developing, maintaining, revising and enforcing information security policy”.

A negated clause scored as personnel screening. PL-18 (NIST PS-3) is scored present in the IT Security Manual on clearance, matching “procedures for the use of maintenance personnel that lack appropriate security clearances”.

An incident-reporting requirement scored as absent. PL-11 (NIST IR-6) is scored absent from the Judicial Information Security Policy, which in fact requires “notifying the JIS information security officer within two business days of a security incident”. The keyword list contains “incident reporting” and “notification requirement” but not “notify”. This single miss matters beyond its cell: an inference that Maryland policies have not been updated to reflect mandatory reporting timelines does not survive reading the document.

Three of these four are not keyword-selection mistakes. A URL is not a control, a term can appear inside its own negation, and a concept can be expressed in vocabulary absent from any finite list. Enlarging the lists fixes the fourth case and worsens the first three.

Projected orchestration cost

The figures in this subsection are token-price projections computed from published per-token rates and estimated token counts, not measured spend. We have not instrumented a run to record actual token usage, and a projection should not be presented in the same register as a measurement. They are reported here because the routing design is part of the system this paper describes, and withholding the arithmetic entirely would make that design harder to evaluate; they are labelled throughout as projected.

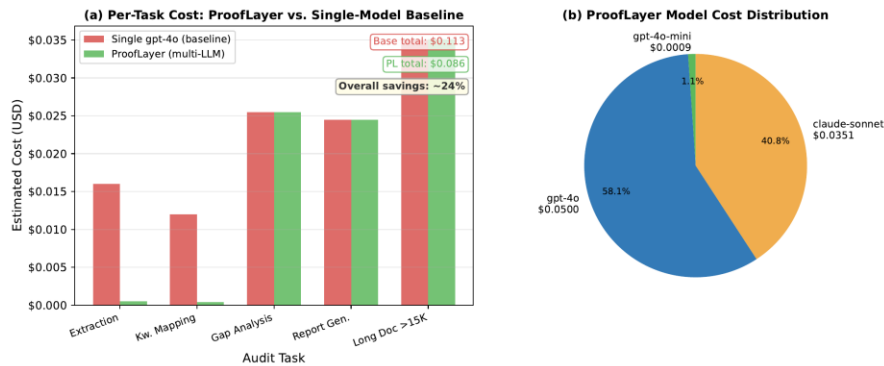


Figure 4: (a) Per-task cost comparison between single-model (gpt-4o only) and ProofLayer multi-LLM routing. (b) ProofLayer model usage distribution across a typical audit session. Source: generated by the authors from the analysis pipeline described in Section 4.1.

Table 5: Per-audit cost breakdown: ProofLayer vs. single gpt-4o baseline.

Task	ProofLayer Model	Baseline	PL Cost	Base Cost	Savings
Doc. Extraction	gpt-4o-mini	gpt-4o	\$0.0005	\$0.0160	96.6%
Keyword Mapping	gpt-4o-mini	gpt-4o	\$0.0004	\$0.0120	96.6%
Gap Analysis	gpt-4o	gpt-4o	\$0.0255	\$0.0255	0.0%
Report Generation	gpt-4o	gpt-4o	\$0.0245	\$0.0245	0.0%
Long Doc (>15K) ^b	claude-sonnet	gpt-4o	\$0.0351	\$0.0350	—
Total (typical audit)			\$0.0860	\$0.1130	23.9%

^bLong-document routing directs claude-sonnet to process the full document context (~6,700 input tokens vs. ~4,000 for the gpt-4o baseline). The lower per-token cost of claude-sonnet offsets the higher token count, yielding equivalent per-task cost; the benefit of routing is improved coverage, not cost reduction.

Figure 4 compares per-task costs between the ProofLayer multi-LLM strategy and a single gpt-4o baseline for a standard document audit.

Table 5 quantifies the cost reduction per task. Routing document extraction and keyword mapping to gpt-4o-mini reduces costs for those tasks by approximately 97% compared to gpt-4o, since these tasks require only token-level pattern recognition rather than complex reasoning.

Gap analysis and report generation are retained on gpt-4o, as these tasks require multi-hop reasoning over long policy contexts and structured output generation. Long documents exceeding 15,000 characters are routed to claude-sonnet, which processes the full document context (~6,700 input tokens) rather than the 4,000-token window used in the single-model baseline. The lower per-token price (\$0.003/1K vs. \$0.005/1K) offsets the higher token count, yielding equivalent per-task cost with improved long-context coverage. Overall, the multi-LLM strategy achieves an estimated 24% cost reduction per audit, driven entirely by routing extraction and mapping to gpt-4o-mini, reducing the projected per-document cost from approximately \$0.113 to \$0.086.

At scale, auditing all ten Maryland policy documents in this corpus costs approximately \$0.86 under the ProofLayer strategy versus \$1.13 under single-model routing. For a state government with hundreds of agency-level policies, this translates to substantial operational savings while preserving higher-capability models for reasoning-intensive tasks.

DISCUSSION

The results in Section 5 surface three interconnected themes that we examine in turn: the concrete remediation priorities implied by the identified gaps, the methodological limitations of a keyword-based Stage 1 approach, and how ProofLayer compares to recent LLM-based compliance systems in adjacent domains. Together, these themes define the near-term improvement trajectory for ProofLayer and illuminate the broader design space for automated government compliance tools.

What this does and does not license

The corrected numbers do not support a remediation ranking for Maryland agencies, and we withdraw the one an earlier version of this work offered. That ranking was computed on truncated text; on full

text the ordering of the most frequently absent controls changes, and Section 5.4 shows that at least one absence is a matcher failure rather than a policy gap. Recommending where a state should spend its remediation budget on the strength of a keyword count would be exactly the overreach this paper is arguing against.

What the results do license is narrower and, we think, more useful. Policy documents in this corpus vary widely in the security vocabulary they contain, from 36% to 95% of a twenty-two control set, and a large part of that variation is length rather than substance. A screening tool built on keyword presence will therefore rank a comprehensive manual above a focused policy almost regardless of content, will mark roughly one in six matches next to language that may negate it, and will miss requirements stated in vocabulary its author did not anticipate. Those are quantified properties of a method that is widely used as a first pass, and they are what a procurement officer evaluating such a tool should be told.

The language-model stage remains described, not evaluated

ProofLayer's Stage 2 language-model classifier is intended to address exactly the failures catalogued in Section 5.4: negation, context, synonymy and control scope are the cases a lexical matcher cannot reach. The system description in Section 3 specifies it, and the orchestration layer routes to it.

It is not evaluated in this paper. We report no precision, recall, F1, calibration or hallucination rate for it, because doing so requires an expert-annotated ground truth for this corpus and none exists. We considered generating labels with a language model and scoring the classifier against them. That would produce numbers, and the numbers would be circular: a model judging a model's output on a task where neither has been checked against a human. Our companion work documents precisely this failure, in which model agreement is mistaken for corroboration, and we are not going to commit it here.

The honest statement is therefore that the failure modes measured in Section 5.4 motivate a semantic stage, that ProofLayer implements one, and that whether it helps is an open question this paper does not answer. Establishing it requires independent annotation, which we identify as the necessary next step rather than a limitation to be noted and passed over.

LIMITATIONS

Stage 2 is not validated, and no accuracy is claimed for it. This study evaluates only Stage 1 keyword-based mapping. The Stage 2 LLM gap analysis is implemented and specified in Section 3 but has not been compared against expert annotation, so its outputs carry no measured accuracy and should not be read as such. The cost analysis in Section 6.2 is likewise an arithmetic projection over published vendor rates, not an observed saving.

We state the protocol that would settle this, both to make the gap concrete and because the design constrains what a future result would mean. The unit of annotation is the document-control pair, giving $10 \times 22 = 220$ judgments. Each pair requires a ternary label: the control is addressed, the control is absent, or the document is out of scope for that control. Two properties matter for the result to be worth having. First, at least one annotator must be independent of the system's authors, since an author labelling the output of their own tool cannot establish independence; agreement between two annotators should be reported as Cohen's κ on a shared subset. Second, the labels must be fixed before the Stage 2 outputs are inspected, or the comparison measures post-hoc rationalization rather than accuracy. Language models may reasonably be used to pre-annotate and to triage which pairs need human attention, but they cannot supply the ground truth against which an LLM stage is judged, because that measures agreement among models rather than correctness.

No ground truth. The control coverage scores and gap frequencies reported here are derived solely from keyword presence. Without expert annotation, precision, recall, or inter-rater agreement cannot be computed. A keyword can appear in a document in a context that does not satisfy the corresponding control (e.g., "encryption is currently unavailable"), and a policy can fully satisfy a control without using the exact terminology in the keyword dictionary. The results should be interpreted as indicators of likely gaps rather than confirmed deficiencies.

Keyword matching is not compliance verification. Keyword presence is used as a proxy for control coverage. The conclusions drawn about specific gaps reflect the output of a pattern-matching heuristic and would require expert review to confirm. Beyond false negatives and false positives in control detection, keyword matching cannot distinguish between

qualitatively different implementations of the same control. For example, the PL-15 keyword set matches any occurrence of “aes” in a document but cannot differentiate between modern authenticated encryption (e.g., AES-GCM) and older cipher modes that may be vulnerable (e.g., AES-CBC without independent integrity protection). A positive PL-15 match therefore indicates mention of AES-class encryption, not necessarily a robust or compliant implementation.

Small sample and absence of confidence intervals. Ten documents represent a single pass over a convenience corpus. No hypothesis testing or bootstrap confidence intervals are reported. Results may not generalize beyond this specific corpus.

Control dataset scope. The 22-control dataset spans major NIST 800-53 families but is not exhaustive. A full FedRAMP Moderate assessment requires over 300 controls. Future work will expand the dataset and validate keyword coverage against annotated policy corpora.

Single-document analysis. Each policy document is analyzed in isolation. In practice, an agency’s posture is determined by the union of all applicable policies; a gap in one document may be addressed by another. Multi-document cross-referencing is a planned ProofLayer feature.

Comparison to Prior Approaches

ComplianceNLP [8] achieves 87.7 F1 on gap detection in the financial regulatory domain using knowledge-graph-augmented RAG. Our work is complementary: ProofLayer targets government cybersecurity policies, applies a two-stage keyword-plus-LLM pipeline, and contributes an empirical analysis of a real policy corpus. Integrating knowledge-graph-based reasoning into ProofLayer’s Stage 2 is a natural extension direction that could improve precision on borderline cases.

CONCLUSION

This paper presented ProofLayer, an automated control coverage screening system that assesses government cybersecurity policy documents against standardized control frameworks (NIST 800-53, FedRAMP, CMMC). Applied to ten Maryland state and local government policy documents, the keyword-based Stage 1 analysis gives a mean control coverage of 62.8% when documents are read in full. The most frequently absent controls are access control policy (PL-01), incident reporting (PL-11) and encryption at

rest (PL-15), each missing from seven of ten documents, though Section 5.4 shows at least one PL-11 absence to be a matcher failure rather than a policy gap. Coverage correlates with document length at $r = 0.690$, so roughly half its variance across the corpus is attributable to length alone. The orchestration cost saving is a token-price projection rather than measured spend and is labelled as such throughout; no measured cost figure is claimed.

These results characterise what automated preliminary coverage screening does and does not measure for resource-constrained government agencies. ProofLayer’s open-source pipeline, available at the project repository, provides a starting point for state CISOs seeking a first pass over their cybersecurity policy posture, subject to the limits set out above. Future work will validate Stage 2 LLM gap analysis against expert-annotated ground truth, expand the control dataset to full FedRAMP Moderate coverage, integrate knowledge-graph-augmented reasoning, and develop multi-document cross-referencing to assess agency-level compliance posture across entire policy suites.

DATA AVAILABILITY

All data and materials used in this study are available. The ten Maryland policy documents analysed are published by their issuing state agencies and are publicly accessible; a manifest listing each document with its SHA-256 hash is provided so that readers can verify they hold identical files. The control dataset, the analysis notebook that reproduces every reported figure, and the ProofLayer implementation are available at <https://github.com/devharsh/ProofLayer>. The evaluated Stage 1 analysis requires no API credentials and can be rerun from the manifest alone. Any further material is available from the author on request.

FUNDING

This research received no grant from any funding agency in the public, commercial, or not-for-profit sectors. No financial support was received for the conduct of this study or the preparation of this article.

AUTHORS’ CONTRIBUTION

Devharsh Trivedi is the sole author and is responsible for conceptualisation, methodology, software, formal analysis, data curation, investigation, writing of the original draft, and writing, review and editing of the revised manuscript.

CONFLICT OF INTEREST

The author declares no competing financial interests. ProofLayer, the system evaluated in this study, was developed as an open-source research project with no commercial sponsor. All Maryland government policy documents used as evaluation data are publicly available under open distribution licenses. No commercial funding was received for this work, and no financial relationship exists between the author and any entity evaluated or referenced in this paper.

ACKNOWLEDGEMENTS

The author thanks the Maryland Department of Information Technology and the University of Maryland Global Campus for publishing the cybersecurity policy documents and council reports used in this study under open distribution licenses.

Use of AI tools. In the preparation of this work, the author used Claude (Anthropic) to assist with manuscript drafting, editing, and LaTeX formatting. After using this tool, the author reviewed and edited all content as needed and takes full responsibility for the content of the publication. No AI tool was used for data analysis, interpretation of results, or drawing of conclusions.

REFERENCES

- [1] Maryland Cybersecurity Council. Maryland Cybersecurity Council biennial activities report. Adelphi (MD): University of Maryland Global Campus; 2025. Available from: <https://www.umgc.edu/mdcybersecuritycouncil>
- [2] Ad Hoc Committee on State and Local Cybersecurity, Maryland Cybersecurity Council. Maryland state and local government cybersecurity: analysis and recommendations. Adelphi (MD): University of Maryland Global Campus; 2021. Available from: <https://www.umgc.edu/content/dam/umgc/documents/upload/maryland-state-and-local-government-cybersecurity-analysis-and-recommendations.pdf>
- [3] Joint Task Force. Security and privacy controls for information systems and organizations. Gaithersburg (MD): National Institute of Standards and Technology; 2020. Report No.: NIST SP 800-53 Rev. 5. Includes updates as of 10 December 2020. doi:10.6028/NIST.SP.800-53r5
- [4] FedRAMP Program Management Office. FedRAMP authorization boundary guidance. Washington (DC): U.S. General Services Administration; 2022. Rescinded and replaced by the FedRAMP Minimum Assessment Scope standard, 2025. Available from: <https://www.fedramp.gov/>
- [5] Office of the Under Secretary of Defense for Acquisition and Sustainment. Cybersecurity Maturity Model Certification (CMMC) model overview, version 2.0. Washington (DC): U.S. Department of Defense; 2021. Superseded in part by the CMMC Program final rule, 32 CFR Part 170, effective 16 December 2024. Available from: <https://dodcio.defense.gov/CMMC/>
- [6] Hassani S, Sabetzadeh M, Amyot D, Liao J. Rethinking legal compliance automation: opportunities with large language models. In: Proceedings of the 32nd IEEE International Requirements Engineering Conference (RE). Piscataway (NJ): IEEE; 2024. p. 432–40. doi:10.1109/RE59067.2024.00051
- [7] Oh Huan Lin, Jay Yong Jun Jie, Mandy Lee Ling Siu, Jonathan Pan. Automated post-incident policy gap analysis via threat-informed evidence mapping using large language models. arXiv:2601.03287 [preprint]. 2026. Available from: <https://arxiv.org/abs/2601.03287>
- [8] Guo D, Wu J, Yiu SM. ComplianceNLP: knowledge-graph-augmented RAG for multi-framework regulatory gap detection. arXiv:2604.23585 [preprint]. 2026. Available from: <https://arxiv.org/abs/2604.23585>
- [9] Kumar B, Roussinov D. NLP-based regulatory compliance: using GPT-4.0 to decode regulatory documents. In: Proceedings of the Georg Nemetschek Institute Symposium on Artificial Intelligence for the Built World; 2024. Available from: <https://arxiv.org/abs/2412.20602>
- [10] Maryland Department of Information Technology. State minimum cybersecurity standards best practices guidebook [Internet]. Annapolis (MD): Maryland Department of Information Technology; 2023 [cited 2026 Jun]. Available from: <https://doit.maryland.gov/policies/ci/Pages/state-minimum-cybersecurity-standards-best-practices-guidebook.aspx>
- [11] Motlagh FN, Hajizadeh M, Majd M, Najafi P, Cheng F, Meinel C. Large language models in cybersecurity: state-of-the-art. arXiv:2402.00891 [preprint]. 2024. Available from: <https://arxiv.org/abs/2402.00891>
- [12] Salman A, Creese S, Goldsmith M. Position paper: leveraging large language models for cybersecurity compliance. In: Proceedings of the IEEE European Symposium on Security and Privacy Workshops. Piscataway (NJ): IEEE; 2024. p. 496–503.
- [13] Qian C, *et al.* xRouter: training cost-aware LLMs orchestration system via reinforcement learning. arXiv:2510.08439 [preprint]. 2025. Available from: <https://arxiv.org/abs/2510.08439>
- [14] Kulkarni S, Kulkarni Y. Benchmarking multi-agent LLM architectures for financial document processing: a comparative study of orchestration patterns, cost-accuracy tradeoffs and production scaling strategies. arXiv:2603.22651 [preprint]. 2026. Available from: <https://arxiv.org/abs/2603.22651>
- [15] Singer-Vine J. pdfplumber: plumb a PDF for detailed information about each char, rectangle, line, and so on, and easily extract text and tables [Internet]. 2024 [cited 2026 Jun]. Available from: <https://github.com/jsvine/pdfplumber>
- [16] ReportLab Inc. ReportLab PDF library [Internet]. 2024 [cited 2026 Jun]. Available from: <https://www.reportlab.com/>
- [17] Trivedi D. Benchmarking municipal cybersecurity readiness through automated policy analytics: evidence from Maryland local governments. J Comput Sci Coll. 2026;42(3). In press. Presented at the CCSC Eastern Conference 2026.

Received on 05-08-2026

Accepted on 25-08-2026

Published on 11-09-2026

<https://doi.org/10.65879/3070-5789.2026.02.07>

© 2026 Devharsh Trivedi

This is an open access article licensed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution and reproduction in any medium, provided the work is properly cited.