

VERIINTERN AI: INTELLIGENT FAKE DETECTION SYSTEM FOR INTERNSHIP OFFERS

Bala Sowndarya B, Kaviyavarshini S, Kowsalya V, Mano Shruthi S and Mrs. S. Umadevi

Department of Artificial Intelligence and Data Science

Avinashilingam Institute of Home Science and Higher Education for Women

ABSTRACT

The proliferation of online internship advertising has been accompanied by a parallel rise in fraudulent internship solicitations that extract registration fees, harvest personal data, or impersonate reputed organizations through visually deceptive company names. Conventional detection mechanisms rely on static keyword matching and are easily circumvented by character-substitution tricks (e.g., “microsoft” for “microsoft”) and by offers that borrow a legitimate company’s name without authorization. This paper presents VeriIntern AI, a multi-layer fusion framework that evaluates internship offers along four independent dimensions: (i) linguistic signal analysis using a TF-IDF-based machine learning classifier with a negation-aware keyword fallback, (ii) corporate identity verification incorporating homoglyph normalization and fuzzy string matching against a curated database of over 200 verified organizations, (iii) URL and domain risk assessment using TLD, WHOIS, and structural pattern heuristics, and (iv) a real-time web intelligence agent that queries the Wikipedia MediaWiki API to validate organizational existence and detect scam contradictions, such as a verified company name paired with an illegitimate payment demand. The four layers are combined through a weighted fusion engine (50% web intelligence, 20% identity verification, 15% network safety, 15% ML classification) with explicit override rules for high-confidence edge cases. The system returns a binary verdict (legitimate or inauthentic), a confidence percentage, and a transparent reasoning report. The framework is implemented as a Flask-based web application with a dark glassmorphism dashboard and is deployed as a publicly accessible service, demonstrating the feasibility of combining lightweight ML, rule-based heuristics, and open knowledge bases to address a growing form of student-targeted fraud.

Keywords- Internship fraud detection, natural language processing, homoglyph detection, fuzzy matching, web intelligence, fusion engine, phishing URL detection, machine learning, fraud classification

I. INTRODUCTION

The expansion of digital recruitment platforms and social media-driven internship advertising has created an accessible but unregulated marketplace in which students searching for early-career opportunities are frequently targeted by fraudulent solicitations. These illegitimate offers commonly request unauthorized payments framed as “registration fees,” “security deposits,” or “processing charges”; impersonate well-known organizations through visually deceptive spellings; embed phishing or shortened URLs; and apply psychological pressure through urgency-based language such as “offer expires tonight” or “limited seats remaining” [1].

Existing detection approaches in this space, where they exist at all, tend to rely on simple keyword matching against a static blacklist of suspicious terms. This approach fails against two classes of attack that are increasingly common in practice. First, sophisticated scams use homoglyph substitution — replacing letters with visually similar characters or character sequences (for example, “rn” for “m”, “0” for “o”, or “@” for “a”) — to impersonate a verified brand while

evading exact-string blacklists. Second, scams increasingly borrow the name of a genuine, well-known organization without any spelling alteration, relying instead on a payment demand that contradicts how legitimate recruitment actually operates. A purely lexical filter cannot resolve this second case because the company name itself is entirely legitimate; only contextual reasoning about the relationship between identity and payment intent can.

This paper proposes VeriIntern AI, a system that addresses both failure modes through a four-layer parallel analysis pipeline feeding into a weighted fusion engine with explicit override logic. Rather than treating fraud detection as a single classification task, the system decomposes the problem into four semi-independent sub-problems — linguistic intent, corporate identity, network infrastructure risk, and real-world organizational verification — and reconciles their outputs through a fusion stage that is specifically designed to catch the case where individual layers disagree (for example, where a real company is correctly identified, but the offer text simultaneously demands an illegitimate payment).

The remainder of this paper is organized as follows. Section II reviews the problem context and related detection paradigms. Section III describes the overall system architecture. Section IV details the four detection layers individually. Section V describes the weighted fusion engine and override rules. Section VI presents the implementation and technology stack. Section VII discusses representative detection scenarios and qualitative validation. Section VIII discusses limitations and threats to validity, and Section IX concludes with directions for future work.

II. PROBLEM STATEMENT AND MOTIVATION

An analysis of observed illegitimate internship solicitation patterns indicates that payment demands account for approximately 35% of fraudulent tactics, followed by fake or impersonated company names (25%), phishing or suspicious URLs (20%), pressure tactics such as artificial deadlines (12%), and false guarantees of placement or certification (8%), as summarized in Table 1.

Tactic	Frequency
Payment demands	35%
Fake company names	25%
Phishing URLs	20%
Pressure tactics	12%
False guarantees	8%

Table 1. Observed Distribution Of Illegitimate Internship Tactics

A. Limitations of Keyword-Based Filtering

Standard detection methods that rely solely on keyword matching are vulnerable to two specific evasion strategies. The first is lexical negation: a phrase such as “no registration fee required” contains the keyword “registration fee” but expresses the opposite sentiment of a scam indicator. Naive keyword counting misclassifies such text as suspicious. The second is visual impersonation through homoglyphs, where a fraudulent company name is rendered to be visually indistinguishable from a legitimate one while remaining a

distinct string at the character level, defeating exact-match or simple blacklist comparisons.

B. The Identity–Intent Contradiction Problem

A more subtle failure mode arises when a scam offer uses the unaltered name of a real, well-known organization. In this case, identity verification alone returns a confident “legitimate” signal, since the company genuinely exists. Detecting the fraud requires recognizing that the offer text contains a request that is inconsistent with how the named organization actually operates — namely, that legitimate employers do not charge candidates registration or processing fees for internship placement. This motivates the system’s scam-contradiction logic, described in Section IV-D.

III. SYSTEM ARCHITECTURE

VeriIntern AI follows a four-layer parallel analysis pipeline that feeds into a weighted fusion engine, as illustrated in Fig. 1. A student submits offer details — free-text offer content, and optionally a company name and a URL. If the company name or URL are not explicitly provided, the system automatically extracts them from the offer text during a preprocessing stage. The four analysis layers then execute independently:

Layer 1 — Semantic Analysis (NLP and ML): linguistic classification of the offer text.

Layer 2 — Identity Verification (Homoglyph-aware): validation of the named organization.

Layer 3 — Network Validation (URL and WHOIS): infrastructure risk assessment of any associated link.

Layer 4 — Web Intelligence (Wikipedia): real-time cross-referencing against a global knowledge base.

The outputs of all four layers are converted into a common inauthenticity-probability scale and combined by the weighted fusion engine, which applies override rules for specific high-confidence edge cases before producing the final output: a binary verdict (SCAM or LEGITIMATE), a confidence percentage, and a structured reasoning report explaining the contributing factors.

IV. CORE DETECTION LAYERS

A. Layer 1: Linguistic Signal Analysis

The text analysis engine evaluates offer language using a tiered keyword scoring system with negation-aware detection. When a trained machine learning model is available, the system uses TF-IDF (Term Frequency–Inverse Document Frequency) vectorization with a Scikit-learn classification pipeline to predict an inauthenticity probability directly from the offer text. If no trained model is available, the system falls back to a deterministic tiered keyword scoring scheme, ensuring the system remains operational without a pre-trained artifact.

Keywords are organized into four tiers, summarized in Table 2: a Critical tier of 13 payment-demand phrases contributing a score impact of +0.50 to +0.90; a High tier of 7 process red-flag phrases (e.g., “no interview required,” “instant offer letter”) contributing +0.20 to +0.30; a Medium tier of 6 pressure-tactic phrases (e.g., “limited seats,” “guaranteed certificate”) contributing +0.06 to +0.20; and a Legitimate-signal tier of 16 phrases (e.g., “interview process,” “coding round”) that reduce the computed score by 20% to 60%.

Tier	Count	Score Impact
Critical (payment)	13	+0.50 to +0.90
High (red flags)	7	+0.20 to +0.30
Medium (pressure)	6	+0.06 to +0.20
Legitimate signals	16	−20% to −60%

Table 2. Keyword Tier Scoring Scheme

To address the lexical-negation failure mode described in Section II-A, the parser scans a 25-character prefix window preceding each matched keyword for negation terms such as “no,” “not,” “without,” “never,” and “don’t.” This allows the system to correctly distinguish “pay registration fee” (treated as a scam indicator) from “no registration fee” (treated as a legitimacy indicator) without requiring a separate semantic parser.

B. Layer 2: Identity Verification

The identity verification layer performs multi-stage corporate identity validation against a curated database of more than 200 verified organizations spanning global technology firms, Indian information-technology companies, consulting firms, banks, startups, pharmaceutical companies, fast-moving consumer goods (FMCG) firms, telecommunications providers, and government public sector undertakings (PSUs). The verification pipeline, shown in Fig. 2, proceeds through four sequential stages:

1. Direct match against the verified-company list (after lowercase normalization and symbol stripping) → VERIFIED, score 1.0.
2. Homoglyph normalization (e.g., rn→m, 0→o, vv→w) followed by re-matching → IMPERSONATION, score 0.02.
3. Fuzzy string matching using sequence-similarity comparison with an 80% similarity threshold → SUSPICIOUS, score 0.05.
4. Pattern match against known red-flag naming conventions → SUSPICIOUS, score 0.05; otherwise → UNVERIFIED, score 0.45 (deferred to the web intelligence agent).

The homoglyph detection map, summarized in Table 3, captures the most common visual substitution tricks observed in impersonation attempts, including digit–letter substitutions (0→o, 1→l, 3→e), symbol–letter substitutions (\$→s, @→a, !→i), and multi-character substitutions (rn→m, vv→w).

Trick	Example (fake → real)
rn → m	rnicrosoft → microsoft
vv → w	vvipro → wipro
0 → o	g00gle → google
1 → l	linkedin → linkedin
\$ → s	micro\$oft → microsoft
@ → a	@mazon → amazon
3 → e	d3loitte → deloitte
! → i	!infosys → infosys

Table 3. Homoglyph Detection Map

C. Layer 3: Infrastructure and URL Safety

The network validation layer performs automated domain analysis using multi-signal risk assessment, checking submitted URLs against a curated list of over 30 trusted domains and applying six independent risk-penalty factors, summarized in Table 4: suspicious top-level domains (TLDs) such as .xyz, .tk, .gq, or .ml commonly associated with free or spam domain registration (−0.25); suspicious structural patterns such as URL shorteners or gibberish hostnames indicative of obfuscation (−0.20); the presence of two or more hyphens, a structure common in phishing domains (−0.15); absence of HTTPS encryption (−0.10); excessive subdomain depth exceeding four levels, indicative of complex redirect chains (−0.10); and domain age under 90 days as determined via WHOIS lookup, indicating a freshly registered and therefore higher-risk domain (−0.20).

Risk Factor	Penalty
Suspicious TLD (.xyz, .tk, .gq, .ml)	−0.25
Suspicious pattern (shortener, gibberish)	−0.20
Two or more hyphens	−0.15
No HTTPS	−0.10
Excessive subdomains (>4 levels)	−0.10
Domain age < 90 days (WHOIS)	−0.20

Table 4. URL And Domain Risk Factors

D. Layer 4: Web Intelligence Agent

The web intelligence agent performs real-time verification by querying the Wikipedia MediaWiki API for the top five results matching the submitted company name. If no results are returned, the entity is treated as having no public presence (score 0.25). If results are found, the agent applies an exact-title match filter and inspects the matched page for organizational attributes such as founding date, revenue, and employee count to confirm the entity is genuinely a company rather than an unrelated topic; if this check fails, the entity is scored as “not a company” (0.40).

If the entity is confirmed as a real organization, the agent performs scam-phrase contradiction checking: it cross-references the offer text against the confirmed identity to determine whether the offer demands an illegitimate payment in the name of a verified organization. If a payment demand is detected, the offer is flagged directly as SCAM DETECTED (score 0.01), reflecting the operating principle that legitimate companies do not charge students for internship placement. If the offer text is otherwise clean, the entity is scored as VERIFIED ORG (0.95). This scam-contradiction mechanism is the system’s primary defense against the identity–intent contradiction problem identified in Section II-B.

V. WEIGHTED DECISION FUSION

The fusion engine combines the outputs of all four layers using a priority-weighted scheme in which the Web Intelligence Agent serves as the primary signal driver, reflecting the system’s design philosophy that real-world organizational verification is the most reliable single indicator of legitimacy. The component weights are summarized in Table 5: Web Intelligence Agent, 50%; Identity Verification, 20%; Network Safety, 15%; and ML Text Classification, 15%.

Component	Weight	Role
Web Intelligence Agent	50%	Primary signal
Identity Verification	20%	Impersonation, homoglyphs
Network Safety	15%	URL, domain, WHOIS
ML Text Classification	15%	Linguistic scam patterns

Table 5. Fusion Engine Component Weights

A. Override Rules

Because a purely linear weighted sum can be insufficiently decisive in specific high-confidence scenarios, the fusion engine applies three override rules, summarized in Table 4, that force the final score toward the correct verdict even when individual layers would otherwise disagree.

Rule	Trigger	Action
Homoglyph impersonation	Company ≤ 0.05	ML ≥ 0.85 ; agent ≤ 0.10
Misspelled company	Company ≤ 0.10	ML ≥ 0.75 ; agent ≤ 0.25
Payment, real name	ML ≥ 0.75	Agent, company ≥ 0.70

Table 6. Fusion Override Rules

The third override rule is the direct computational mechanism for resolving the identity–intent contradiction described in Section II-B: when the linguistic layer detects a high-confidence payment demand, the system forces both the web-intelligence and identity scores toward “scam” even though the company itself was correctly recognized as genuine, ensuring the final verdict reflects the deception rather than the organization’s legitimate existence.

B. End-to-End Workflow

The complete system workflow proceeds in six steps: (1) the user submits offer text and, optionally, a company name and URL; (2) the system auto-extracts the company name and URL from the offer text if not explicitly supplied; (3) all four analysis layers execute; (4) each layer’s output is converted to a common inauthenticity-probability scale and combined via the weighted fusion formula; (5) override rules are applied where trigger conditions are met; and (6) the system returns the final verdict, a confidence percentage, and a structured reasoning report.

VI. IMPLEMENTATION

A. Technology Stack

The system is implemented in Python 3.10+, using Scikit-learn 1.4 with TF-IDF vectorization for the ML classification component, Pandas 2.2 and NumPy 1.26 for data processing, and Flask 3.0 with Gunicorn for the backend web server. The frontend dashboard is built using HTML5, CSS3, and ES6 JavaScript. Real-time organizational verification uses the MediaWiki API (Wikipedia) accessed via the Requests and BeautifulSoup4 libraries, while network analysis uses python-whois and urllib. The application is deployed on Render via a Procfile-based deployment configuration and is publicly accessible at the project’s hosted endpoint.

B. API Design

The system exposes a single primary REST endpoint, POST /analyze, accepting a JSON payload containing the offer text, an optional company name, an optional URL, and a flag to skip the web-scraping stage if rapid response is preferred over full verification. The endpoint returns a JSON response containing the final verdict (e.g., “INAUTHENTIC”), a boolean scam flag, a confidence percentage, the four individual component scores (ML scam probability, agent legitimacy, company legitimacy, and URL safety), and a list of natural-language explanations supporting the verdict.

C. Codebase Organization

The application is structured around a core app.py module implementing the API server, the fusion engine, and ML scoring logic, supported by a utils package containing three specialized modules: company_check.py for the identity engine, scraping_agent.py for the web intelligence agent, and url_check.py for URL safety analysis. An automated test suite (test_scoring.py) validates the scoring logic, and the frontend dashboard is implemented with IBM Plex typography in a dark glassmorphism design system.

D. Testing and Validation

An automated test suite validates the system across five categories, summarized in Table 7: seven homoglyph detection cases (covering tricks such as mncrosoft, g00gle, vvipro, 1nfosys, micro\$oft, d3loitte, and @mazon), two fuzzy impersonation cases involving typographical variants (e.g., “Gogle,” “Microsft”), three verified-company recognition cases, seven negation-awareness cases distinguishing scam keywords from their negated forms, and four end-to-end ML scoring cases validating the full inauthenticity-probability computation.

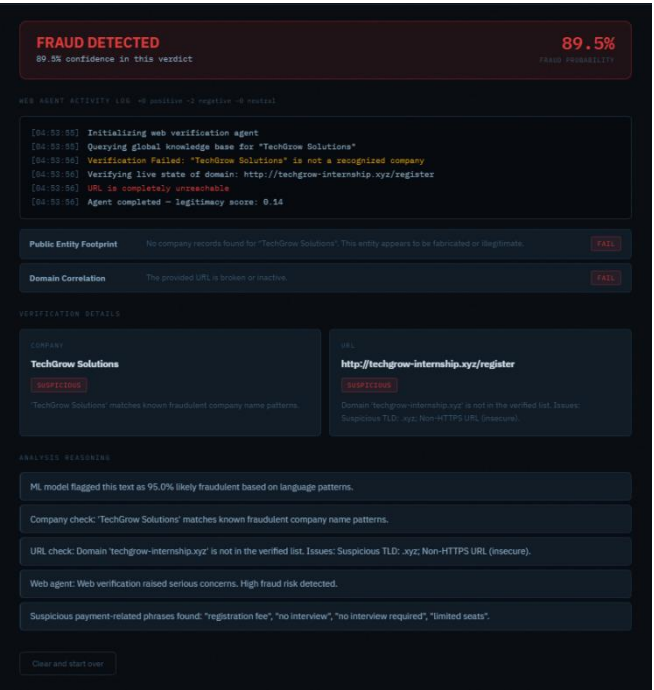


Fig.1. A web verification dashboard analyzing a Google internship posting and marking it as **legitimate** with a 95.6% confidence score.

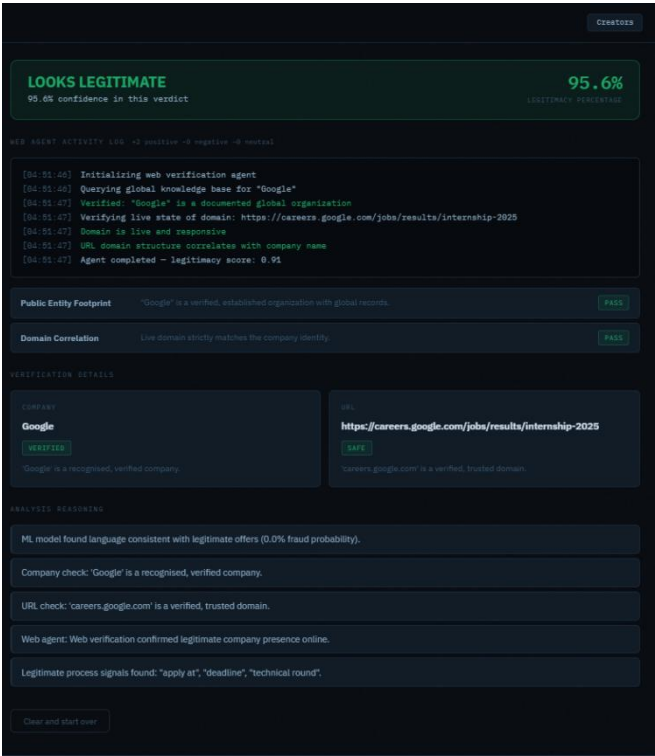


Fig.2.A web verification dashboard detecting a **fraudulent** job posting from "TechGrow Solutions" with an 89.5% fraud probability.

Test Category	Cases	Validates
Homoglyph detection	7	microsoft, g00gle, vvipro, etc.
Fuzzy impersonation	2	Typo variants (Gogle, Microsoft)
Verified companies	3	Real-company recognition
Negation awareness	7	Scam vs. negated keywords
ML scoring	4	End-to-end probability calc.

Table 7. Automated Test Suite Coverage

VII. DETECTION SCENARIOS

Table 7 presents five representative detection scenarios illustrating how the four-layer architecture and override logic combine to reach a final verdict, including cases of agreement across all layers and cases where the override rules resolve a deliberate contradiction between identity and intent.

Scenario	ML	Co.	URL	Agent	Verdict
Real co., legit. offer	0.00	1.0	1.0	0.95	LEGIT
Fake co., payment	0.90	0.45	0.35	0.25	SCAM
Real name + payment	0.75	1.0	1.0	0.01*	SCAM
Homoglyph trick	0.85+	0.02	0.60	0.10*	SCAM
Unknown, clean text	0.00	0.45	—	0.50	LEGIT

Table 8. Representative Detection Scenarios

The Third Scenario In Table 8 Is The Most Instructive: Although The Company-Verification Layer Correctly Assigns The Maximum Legitimacy Score (1.0) And The Url Is Assessed As Safe, The Linguistic Layer's Detection Of A Payment Demand (MI Score 0.75) Triggers The "Payment With Real Name" Override Rule From Table 6, Forcing The Agent And Company Scores Toward The Scam Range And Producing A Final Inauthentic Verdict. This Demonstrates The System's Core Design Objective: Resolving The Identity-Intent Contradiction That Defeats Identity-Only Or Keyword-Only Detection Approaches.

VIII. LIMITATIONS AND THREATS TO VALIDITY

Several Limitations Of The Current Implementation Should Be Acknowledged. First, The Verified-Company Database, While Covering More Than 200 Organizations Across Multiple Sectors, Is Necessarily Finite; Genuinely Legitimate But Lesser-Known Employers May Be Scored As Unverified And Deferred To The Web Intelligence Agent, Whose Wikipedia-Based Verification Is Itself Incomplete For Smaller Or Regional Organizations That Lack A Wikipedia Presence. Second, The Web Intelligence Agent Depends On Third-Party Api Availability And Rate Limits, Introducing A Dependency That Is Outside The System's Direct Control. Third, The Fallback Keyword-Tiered Scoring System, While Robust To Simple Negation, May Not Generalize To More Linguistically Sophisticated Phrasing Than The Curated Keyword Lists Anticipate; The Tf-Idf-Based MI Classifier Mitigates But Does Not Eliminate This Risk, And Its Accuracy Is Bounded By The Quality And Coverage Of Its Training Data. Finally, The Homoglyph Detection Map And Fuzzy-Matching Threshold (80% Similarity) Were Defined Based On Observed Impersonation Patterns And May Require Periodic Recalibration As Adversarial Tactics Evolve.

IX. CONCLUSION AND FUTURE WORK

This Paper Presented Veriintern Ai, A Multi-Layer Fusion Framework For Detecting Fraudulent Internship Offers That Combines Linguistic Pattern Analysis, Homoglyph-Aware Identity Verification, Url And Domain Risk Assessment, And Real-Time Web Intelligence Into A Single Weighted Decision Engine With Explicit Override Logic. The System's Central Contribution Is Its Resolution Of The Identity-Intent Contradiction Problem, In Which A Scam Offer Borrows The Unaltered Name Of A Genuine Organization While Simultaneously Demanding An Illegitimate Payment — A Case That Defeats Detection Approaches Relying On Identity Verification Or Keyword Matching In Isolation. By Fusing Four Independent Analytical Signals And Applying Targeted Override Rules For High-Confidence Edge Cases, The System Aims To Reliably Flag Both Visual Impersonation Attacks And Contradiction-Based Scams While Minimizing False Positives Against Legitimate Recruitment Language.

Future Work Includes Expanding The Verified-Company Database Through Automated Ingestion From Corporate Registries, Incorporating Additional Knowledge-Base Sources Beyond Wikipedia To Improve Coverage Of Regional And Smaller Organizations, Conducting A Quantitative Evaluation Against A Labeled Corpus Of Real And Fraudulent Internship Offers To Establish Formal Precision And Recall Metrics, And Extending The System As A Browser Extension Or Mobile Application To Provide Real-Time Protection At The Point Where Students First Encounter Internship Solicitations.

ACKNOWLEDGMENT

The Authors Thank Their Academic Institution For The Guidance And Resources Provided In Support Of This Project, Undertaken As An Academic Submission.

REFERENCES

- [1] AI-Based Fraud Internship Detection System: Project Proposal, Internal Document, Academic Institution, 2026.
- [2] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
- [3] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manag., vol. 24, no. 5, pp. 513–523, 1988.
- [4] Wikimedia Foundation, "MediaWiki Action API," [Online]. Available: https://www.mediawiki.org/wiki/API:Main_page
- [5] J. W. Ratcliff and D. E. Metzner, "Pattern matching: The gestalt approach," Dr. Dobb's J., July 1988.
- [6] OpenAPI / Internet Engineering Task Force, "WHOIS Protocol Specification," RFC 3912, 2004.
- [7] M. Pilgrim, "HTML5: Up and Running," O'Reilly Media, Sebastopol, CA, 2010.
- [8] Flask Documentation, Pallets Projects, [Online]. Available: <https://flask.palletsprojects.com>
- [9] Mano Shruthi S, "VeriIntern-AI," GitHub repository [Online]. Available: <https://github.com/ManoShruthiS/VeriIntern-AI> , 2026.