

AI Based Secure Persona Prediction and Data Leakage Prevention System

Pranay bawankar
Dept. of Artificial Intelligence
G H Raison College of Engineering
Nagpur

Jayshree mahajan
Dept. of Artificial Intelligence
G H Raison College of Engineering
Nagpur

Tejas ishwarkar
Dept. of Artificial Intelligence
G H Raison College of Engineering
Nagpur

Dhanshree gundawar
Dept. of Artificial Intelligence
G H Raison College of Engineering
Nagpur

Prof. Krupali dhawale
Asst. Professor, AI Dept.
G H Raison College of Engineering
Nagpur

Abstract—Organizations exchange confidential information through email, cloud platforms, and enterprise collaboration applications, and insiders with legitimate access are among the hardest sources of leakage to detect. Rule-based Data Leakage Prevention (DLP) tools match keywords and patterns but cannot determine who is sending the data, whether the behavior is consistent with that person's role, or why an alert was raised. Behavior analytics, in contrast, model users but largely ignore message content. This paper presents the design of an AI-based secure persona prediction and data leakage prevention system that combines these two views. A persona engine using DistilBERT and Bi-LSTM models infers a behavioral persona from communication text; a hybrid detector using Named Entity Recognition (NER) and validated pattern matching identifies sensitive content; and a dynamic risk scoring algorithm combines data criticality, frequency, behavioral anomaly, and channel context into a continuous risk score. A priority-based decision policy selects graded actions, suppresses duplicate alerts, and adapts thresholds using analyst feedback. SHAP and LIME are integrated to provide decision explanations and an audit trail. The paper distinguishes proposed components from prototype evidence, defines a reproducible evaluation methodology covering persona classification, sensitive data detection, alert quality, latency, and explanation quality across ten test scenarios, and discusses privacy, ethical, and technical limitations. No experimental results are reported; the contribution is the integrated architecture, decision policy, and evaluation protocol.

Keywords—Data Leakage Prevention, Explainable AI, Machine Learning, Natural Language Processing, Persona Prediction, Insider Threat, Risk Scoring, Named Entity Recognition, Alert Fatigue

I. INTRODUCTION

Digital communication is the backbone of modern organizations. Employees exchange information through email, messaging tools, shared documents, cloud storage,

and customer-facing applications, and each channel is a potential exit path for customer records, financial data, credentials, source code, and other intellectual property. Remote and hybrid work increase the number of devices, networks, and third-party services that touch corporate data, so the surface that must be protected continues to grow.

The insider threat is particularly difficult because insiders already hold valid credentials and know where valuable data is stored. Incidents fall into three broad groups: malicious insiders who exfiltrate deliberately, negligent insiders who leak through carelessness, and compromised accounts

controlled by outsiders. In all three cases, an individual message may appear ordinary. Detection therefore depends on understanding who is communicating, in what manner, to whom, and whether the communication is consistent with the sender's role and history.

Conventional DLP products answer the question, "Does this content match a policy?" They rely on keyword dictionaries, regular expressions for identifiers such as card or account numbers, document fingerprints, and fixed channel rules [1], [2]. These mechanisms work well for rigidly structured identifiers but have three recurring weaknesses. They lack context: the same string can be routine in a finance team's reconciliation and dangerous in a message sent to a personal address by a departing employee. They can produce many false positives, which overload analysts and lead to alert fatigue and missed true incidents. They are also weakly interpretable: a pattern match says that something was found, not why the event is risky.

Behavior analytics systems answer the complementary question, "Is this user behaving unusually?" Anomaly

detectors over logons, file access, and device usage can flag deviations from a user's baseline [3]–[5]. Used alone, however, they know little about what is being sent. A transfer flagged as unusual may contain nothing sensitive, while a perfectly typical message may contain a customer database extract.

Recent advances in Natural Language Processing (NLP) make a combined approach practical. Pre-trained transformers capture semantics and writing style, compact variants such as DistilBERT keep inference cost manageable [14], [15], and recent studies show that personality or persona-related characteristics can be inferred from text [6]–[10]. Explainable Artificial Intelligence (XAI) methods such as SHAP and LIME provide feature- and token-level explanations of model decisions [19], [20]. The motivation for this work is the gap between content-centric DLP and user-centric behavior analytics: effective leakage prevention needs content sensitivity, sender persona, behavioral baseline, channel and destination context, and a human-readable justification coordinated by a component that decides what action to take and when.

Research problem. How can sensitive content detection and text-based persona modeling be fused in a software-first system running on commodity infrastructure to produce timely, low-noise, and explainable leakage decisions for adaptive insider threats?

Contributions. (i) A system architecture that fuses persona prediction, hybrid sensitive data detection, and channel context through an explicit risk model; (ii) a dynamic risk scoring algorithm with persona-deviation, frequency, and criticality terms; (iii) a priority-based decision policy with alert de-duplication, role-calibrated thresholds, and analyst feedback to reduce alert fatigue; (iv) an integrated SHAP and LIME explanation layer that produces analyst-readable rationales and an audit trail; (v) a clear separation of proposed components and prototype evidence; and (vi) an evaluation protocol with metrics, ablations, and ten scenarios intended to be executed before any performance claim is made.

The remainder of the paper is organized as follows. Section II defines the problem. Section III states the aim, objectives, and research gap. Section IV reviews the literature. Section V presents the methodology. Section VI describes implementation and workflow. Section VII covers explainable and responsible design. Sections VIII and IX define the evaluation and analysis plan. Section X discusses privacy, security, and limitations. Sections XI and XII present future scope and conclusions.

II. PROBLEM STATEMENT

Let an organization have users $U = \{u_1, \dots, u_n\}$ communicating over channels C . A communication event is a tuple $e = (u, c, t, x, a)$, where u is the sender, c is the channel, t is the timestamp, x is the textual or attached content, and a is the set of recipients. The organization defines sensitive data classes S , including personal identifiers, financial data, authentication secrets, health records, and proprietary documents, each with a criticality weight $w_s \in [0, 1]$.

The DLP problem is to learn a decision function $f(e, H_u)$ that maps an event and the sender's history H_u to an action in $\{\text{allow, warn, block, escalate}\}$ so that leakage of sensitive content to unauthorized recipients is minimized, subject to two operational constraints: the false-positive rate must remain within a budget that analysts can sustain, and the decision must be returned within an interactive latency budget. Three challenges make this difficult.

Adaptive insider threats.

Insiders can evade fixed rules by paraphrasing, encoding, splitting data across messages, or switching to unmonitored channels, and their behavior can change over time. Detection must therefore rely on semantic and behavioral signals that are harder to evade than literal patterns.

Context blindness and false positives.

Pattern-only systems cannot reliably separate legitimate business use from risky use. High false-positive rates consume analyst time, erode trust, and can push users toward workarounds.

Unexplainable alerts.

Even a correct alert has limited operational value if the analyst cannot see why it was raised. Triage, user communication, and compliance review all need a defensible justification. Black-box models added without explanation can replace one opacity problem with another.

The problem addressed here is therefore: how can an organization detect and prevent sensitive data leakage by adaptive insiders using context-aware behavioral modeling while keeping alerts accurate, timely, and explainable?

III. AIM, OBJECTIVES AND RESEARCH GAP

Aim. To build an intelligent DLP framework that combines persona prediction, sensitive data detection, risk scoring, and explainable alerts in a single real-time pipeline.

Objectives.

- Persona prediction—Infer a communication persona and behavioral baseline for each user from text with DistilBERT and Bi-LSTM models, and measure deviation from the expected persona for the user's role.
- Sensitive information detection—Detect financial information, credentials, personal identifiers, and proprietary content using a hybrid of NER and validated pattern matching.
- Real-time risk management—Fuse persona, content sensitivity, frequency, anomaly, and channel signals into a continuous risk score and trigger graded responses within the latency budget.
- Accuracy optimization—Reduce false positives and false negatives through tuning, cross-validated model selection, threshold calibration, and analyst feedback, reporting precision, recall, F1 score, and false-positive rate rather than accuracy alone.
- Alert fatigue reduction—De-duplicate and prioritize alerts and adapt thresholds per role.
- Explainable AI—Attach SHAP and LIME explanations to decisions and store them as an audit trail.

- Scalability—Provide a modular, stream-oriented design that can scale across channels and users and support model updates without downtime.
- Privacy preservation—Minimize the collection, retention, and exposure of employee communication in the system's own storage.

What exists.

Content-based DLP provides pattern and fingerprint matching [1], [2]. Behavior analytics provide anomaly detection over activity logs [3]–[5]. NLP research provides persona and personality inference from text [6]–[10]. XAI research provides model-agnostic explanation methods such as SHAP and LIME [19], [20]. Privacy research provides differentially private and federated training approaches [21]–[24].

What is missing.

Based on the surveyed work, we identify the absence of a unified, software-first system that jointly handles (1) text-based persona inference; (2) content-level sensitive data detection; (3) behavioral baselines and drift; (4) channel and destination context; (5) adaptive decision-making that decides whether to allow, warn, or block and suppresses repeated alerts; and (6) per-decision explanation with an audit trail. Individual pieces exist; their coordinated, stateful integration is the gap addressed here. We do not claim that no such system exists anywhere; rather, the surveyed literature does not present this exact combination.

IV. LITERATURE REVIEW

A. Data Leakage Prevention

Surveys of DLP organize solutions by the state of the data (at rest, in motion, or in use) and by detection technique, including regular expressions, fingerprinting, statistical analysis, and machine learning [1], [2]. They report persistent challenges in accuracy, context, and the handling of unstructured content. These systems establish what can be matched, but not necessarily who is sending it or why the event is unusual.

B. Insider Threat Detection

The CERT insider-threat datasets provide synthetic but realistic logs of user activity for research [3]. Deep-learning models over structured activity streams detect anomalous users without labels [4], and Isolation Forest isolates anomalies by random partitioning without requiring a density model [5]. These methods are strong on behavior but are designed primarily for activity metadata, with limited use of message semantics and little explanation.

C. Persona and Personality Inference from Text

Early work linked linguistic cues to personality in conversation and text [6], and later surveys reviewed deep-learning approaches to personality detection [7]. Recent work applies transformer and Bi-LSTM models to personality prediction from text [8], while privacy-oriented research studies text-based personas for modeling user privacy decisions [9]. Work on persona dynamics emphasizes that user characteristics or persona traits can evolve across interactions [10]. This body of work supports text-based persona inference, but labels typically come from

self-report, social-media, or synthetic interaction data rather than workplace communication, and the objective is rarely security.

D. Text Representations and Sequence Models

Word2Vec and GloVe provide dense static embeddings from local and global co-occurrence statistics [11], [12]. The Transformer architecture [13] and BERT [14] provide contextual representations that disambiguate meaning from surrounding text. DistilBERT compresses BERT through knowledge distillation to a smaller and faster model [15]. LSTM [16] and bidirectional recurrent networks [17] model sequences with context from both directions.

E. Sensitive Entity Recognition

Neural sequence taggers combining Bi-LSTM and conditional random fields are established approaches for NER [18]. Structured identifiers such as payment card numbers can be validated with check-digit algorithms such as the Luhn algorithm [30]. Pattern matching and neural tagging are complementary: patterns are precise on rigid formats, whereas neural taggers generalize to free-text entities.

F. Explainable AI

SHAP assigns each feature a Shapley-value-based contribution to a prediction [19], and LIME fits a local interpretable surrogate around an individual prediction [20]. Both are widely used to expose the drivers of model decisions. They are approximations and can disagree; such disagreement should be reported rather than hidden.

G. Privacy-Preserving Learning

A system that models employees' writing is itself privacy-sensitive. Differentially private training can bound what a model reveals about individual training records [21], [23], and federated learning keeps raw data on local nodes while sharing model updates [22]. Recent work also studies differential privacy for text rewriting and private text generation [24], illustrating the utility–privacy trade-off.

Table I summarizes the surveyed work and the gap each study leaves open.

Study	Approach and technology	Contribution	Limitation / gap
Shabtai [1]; Alneya di [2]	DLP surveys; patterns, fingerprints, ML	Taxonomy and challenges of DLP	Review only; no persona-aware system
Glaser [3]	Synthetic insider-threat logs	Benchmark activity data	Metadata; little message content
Tuor [4]	Unsupervised deep anomaly detection	Detection without labels	No message semantics or explanation
Liu [5]	Isolation Forest	Efficient anomaly isolation	Needs informative behavioral features
Mairesse [6]; Mehta [7]	Linguistic and deep personality detection	Text carries personality signal	Not security-aimed; non-workplace labels
Naz et al. [8]	Transformer and Bi-LSTM personality prediction	Contextual text features for personality prediction	Public personality data; no DLP link
Fawaz et al. [9]	Text-based personas for privacy decisions	Persona modeling from prior privacy behavior	Privacy-decision setting; not leakage detection
Lim et al. [10]	Dynamic persona	Persona traits can evolve	No leakage objective

	modeling		
Devlin [14]; Sanh [15]	BERT; DistilBERT	Strong contextual representations	Generic; needs task-specific fine-tuning
Lample [18]	Bi-LSTM-CRF NER	Neural entity tagging	Needs labels; no validation layer
Lundberg [19]; Ribeiro [20]	SHAP; LIME	Feature and token attributions	Approximate; limited DLP integration
Abadi [21]; McMahian [22]; Meisenbacher et al. [24]	Differential privacy; federated/private text learning	Formal privacy and decentralized/private processing	Utility cost; not tied directly to leakage detection

TABLE I. LITERATURE REVIEW AND IDENTIFIED GAPS

V. PROPOSED METHODOLOGY

A. System Overview

The system is organized around a continuous loop (Fig. 2). A communication event is captured from a channel and parsed. Two parallel branches process it: a language branch preprocesses text, builds embeddings, and predicts the sender's persona; a content branch detects sensitive entities in the minimally altered original text. Both outputs, together with the user's history and channel and recipient context, are merged into a risk state on which a decision policy selects an action. An explanation is generated, the action is executed, and the outcome, including analyst feedback, is used to update baselines and thresholds.

Data leakage prevention across communication channels

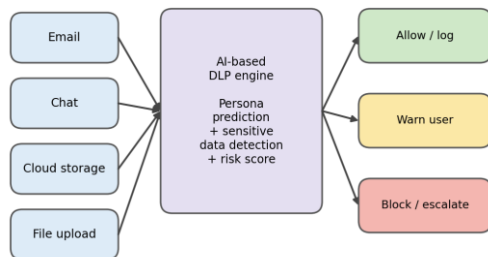


Fig. 1. Leakage prevention across communication channels.

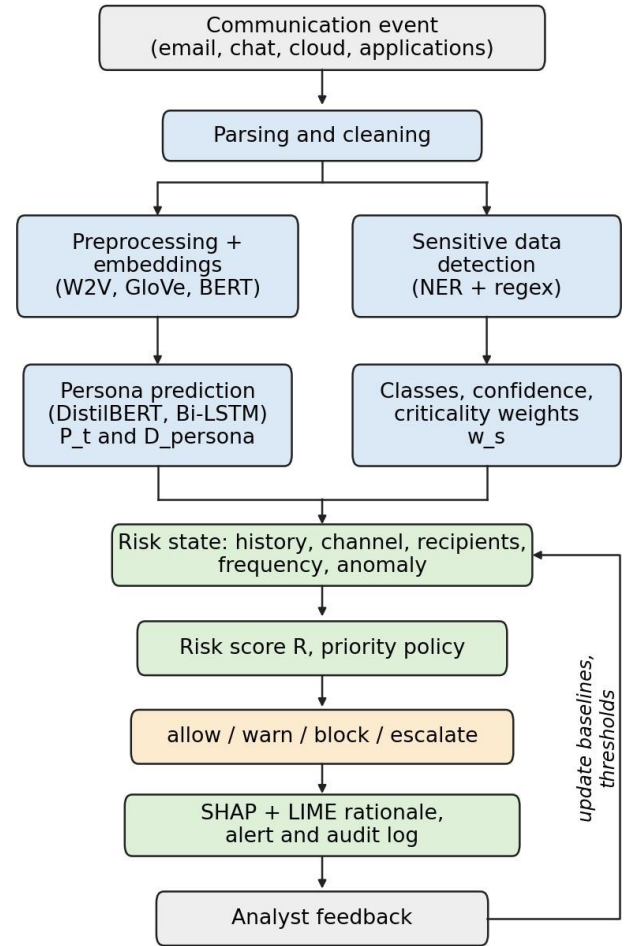


Fig. 2. Overall system architecture and feedback loop.

B. Text Preprocessing

Text is normalized before the language branch is applied. (1) Parsing and cleaning: headers, signatures, quoted replies, HTML, and boilerplate are removed, and attachments are converted to text where possible. (2) Tokenization: word tokens are used for classical embeddings, and subword tokenization is used for transformers so that rare words and identifiers do not cause out-of-vocabulary failures. (3) Stop-word removal is applied for Word2Vec, GloVe, and Bi-LSTM inputs only and is skipped for transformer inputs because function words can carry stylistic signal useful for persona inference. (4) Lemmatization reduces words to dictionary forms when classical models are used. (5) Protected-pattern masking is applied before text is persisted or logged so that structured identifiers are not retained unnecessarily.

C. Feature Extraction

Three representation families are considered for evaluation. Word2Vec [11], using skip-gram or CBOW, maps words to dense vectors that capture local co-occurrence. GloVe [12] is trained on global co-occurrence counts by minimizing the following objective:

$$J = \sum_{i,j} f(X_{ij}) [w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij})]^2 \quad (1)$$

where X_{ij} is the co-occurrence count of words i and j , f is a weighting function, w and $\tilde{w}$ are word and context vectors, and b and $\tilde{b}$ are bias terms. BERT-family encoders [14] give each token a context-dependent vector, which helps resolve

polysemy and capture stylistic cues. The pooled [CLS] representation is used as the message embedding, and long documents are chunked with overlap and aggregated. In addition to embeddings, interpretable behavioral features are computed, including message length, formality indicators, recipient count, external-recipient ratio, time of day, attachment type, and frequency of sensitive terms. These features feed the risk model and make SHAP explanations more interpretable to analysts.

D. Persona Prediction Engine

The persona engine assigns each user a distribution over behavioral personas, for example, routine collaborator, external-facing communicator, data handler, and high-volume sharer. The taxonomy is defined with domain experts and mapped to organizational roles because the same behavior can be normal for one role and anomalous for another (Fig. 3).

DistilBERT classifier.

A six-layer distilled encoder [15] is fine-tuned with a classification head over the pooled representation, producing a softmax distribution over persona classes. Training minimizes class-weighted cross-entropy:

$$L = - \sum_k \alpha_k y_k \log p_k \quad (2)$$

where α_k compensates for class imbalance.

Bi-LSTM model.

A bidirectional LSTM [16], [17] over Word2Vec or GloVe sequences integrates past and future context at each position. Final forward and backward states are concatenated and passed through a dense layer with dropout. It serves as a lighter baseline and ensemble member. A second Bi-LSTM over a user's sequence of message embeddings can model behavior over time.

Ensemble and persona drift.

The two models' probabilities are combined by a weighted average tuned on validation data. For each user, an exponentially weighted moving average of persona distributions forms a baseline profile B_u . The deviation of a new event is the Jensen-Shannon divergence [26] between the current distribution P_t and the baseline:

$$D_{\text{persona}} = \text{JS}(P_t || B_u) \quad (3)$$

A large value indicates that the event is inconsistent with the user's history or role. The baseline is updated only with events judged benign to reduce the risk of gradual profile poisoning. The sensitive-data detector operates on minimally altered text because lemmatization and stop-word removal can destroy character patterns needed by regular expressions. The two branches therefore use different views of the same message.

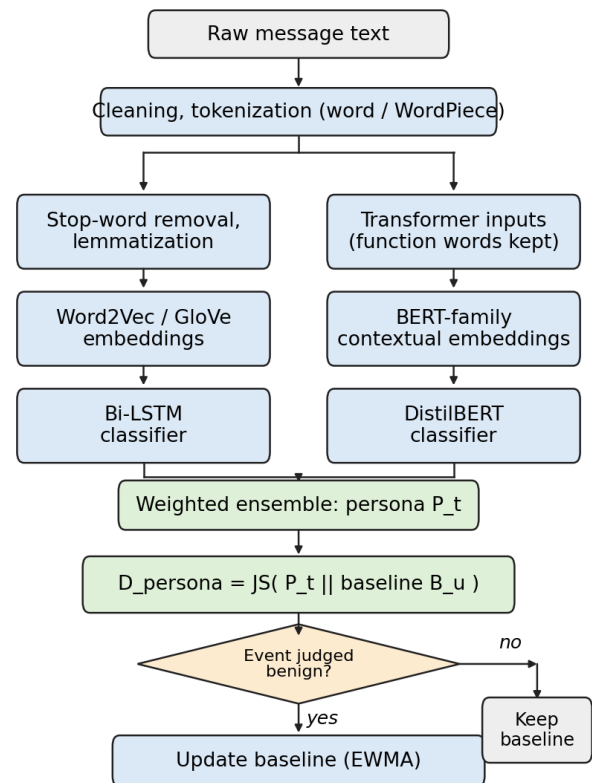


Fig. 3. Persona prediction pipeline from raw text to drift score.

E. Sensitive Data Detection

A hybrid design combines the strengths of rules and neural tagging (Fig. 4). The pattern and validation layer uses regular expressions to identify payment card numbers, bank account and routing numbers, national identifiers, API keys, private-key headers, access tokens, and password patterns. Matches are validated to reduce false positives: card numbers use the Luhn check [30], key-like strings can be checked using entropy and known provider prefixes, and weak matches can be supported by nearby context words such as “password.”

The NER layer uses a transformer-based tagger fine-tuned with BIO labels, with a Bi-LSTM-CRF [18] as a baseline. It recognizes entities without rigid formats, including person names, organizations, addresses, project code names, medical terms, and contractual or financial descriptions. Detections from both layers are merged and de-duplicated by span overlap and assigned a class $s \in S$ with confidence c . The class criticality w_s comes from organizational policy, so credentials and customer financial data can be weighted more heavily than internal memos.

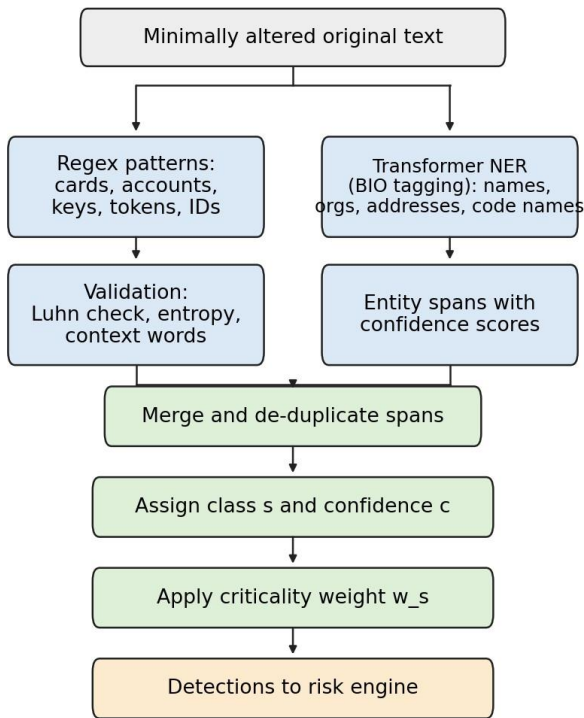


Fig. 4. Hybrid sensitive data detection using pattern validation and NER.

F. Risk Scoring Algorithm

For an event e from user u , the engine computes a score $R \in [0, 1]$ from detected items $d_1, \dots, d_m$ with confidence c_j and criticality w_j . Content criticality uses a saturating aggregation so that one critical item is not diluted by many trivial ones:

$$S_{\text{content}} = 1 - \prod_j (1 - c_j w_j) \quad (4)$$

Frequency compares recent sensitive events with the user's role-specific norm:

$$S_{\text{freq}} = 1 - \exp[-\lambda \cdot \max(0, n_{\text{recent}} - n_{\text{expected}})] \quad (5)$$

where n_{recent} counts sensitive events in a sliding window, n_{expected} is the baseline, and λ controls sensitivity. Anomaly combines persona deviation with contextual anomaly such as unusual time, an unfamiliar external domain, or abnormal volume:

$$S_{\text{anom}} = \alpha D_{\text{persona}} + (1 - \alpha) A_{\text{context}} \quad (6)$$

where $A_{\text{context}} \in [0, 1]$ can be obtained from an Isolation Forest [5] or an autoencoder over behavioral features. The final score is a logistic combination:

$$R = \sigma(b + \beta_1 S_{\text{content}} + \beta_2 S_{\text{freq}} + \beta_3 S_{\text{anom}} + \beta_4 Z_{\text{channel}}) \quad (7)$$

where Z_{channel} encodes channel and destination risk, for example, an external personal domain versus an internal channel. Coefficients are initialized from expert policy and then fitted by logistic regression on analyst-labelled incidents. All weights are design parameters to be tuned on validation data; none is claimed to be optimal.

G. Decision Policy and Alert Management

Scores and context are mapped to one action per event by a fixed priority policy. P1 Critical, such as high-criticality credentials sent to an unauthorized external destination or a score above the block threshold, leads to blocking or quarantine and escalation. P2 High, where the score is above

the warning threshold with a strong anomaly, holds the message for analyst review. P3 Medium, involving sensitive content with moderate risk, sends a just-in-time warning with an explanation and a justification prompt. P4 Low allows and logs. P5 Informational allows without notification.

Only the highest-priority applicable action is taken (Fig. 5). Alerts for the same user, content fingerprint, and destination within a cooldown window are grouped into one incident, and recurring benign patterns confirmed by analysts are down-weighted. Thresholds are calibrated per role: a confirmed false positive raises the local threshold, while a confirmed leak lowers it. These measures target alert fatigue directly, and the safety-critical arbitration for P1 and P2 is deterministic and auditable rather than left to a free-form model.

H. Explainable AI Integration

Explainability is a design requirement (Fig. 6). SHAP [19] attributes the risk score to interpretable features through Shapley values, so analysts can see which factors, such as external-recipient ratio, presence of credentials, persona deviation, or unusual time, raised or lowered the score. Global SHAP summaries support model audit and can reveal spurious reliance on features such as department or message length. LIME [20] perturbs the text of a single message and fits a sparse linear surrogate, highlighting tokens that most influenced the persona or NER output. Both outputs are rendered as a short rationale.

Because SHAP and LIME can disagree, the system reports their agreement and flags low-agreement explanations for human review. Explanations are stored with the alert as an audit trail. The numerical example previously shown in the draft has been removed from the explanatory text because no measured risk score has been reported.

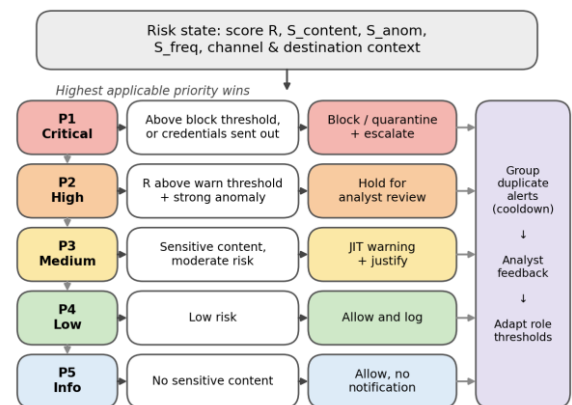


Fig. 5. Priority-based decision policy and alert management workflow.

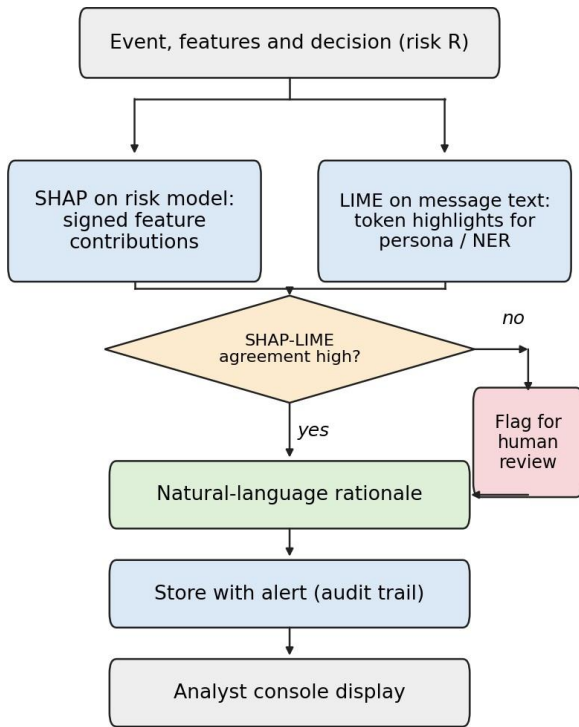


Fig. 6. Explainable AI integration and alert rationale generation.

VI. IMPLEMENTATION AND SYSTEM WORKFLOW

A. Technology Stack and Development Status

Table II lists the modules, technologies, purposes, and status at the time of writing. “Prototype evidence” is used only where the submitted manuscript contains interface screenshots; “Proposed” denotes a designed component that has not been fully integrated or experimentally validated. This distinction prevents interface evidence from being interpreted as performance evidence.

Module	Technology	Purpose	Status
Core language	Python	Orchestration of modules	Proposed
Preprocessing	NLTK / spaCy	Tokenization, stop words, lemmatization	Proposed
Embedding	Word2Vec, GloVe [11], [12]; BERT [14]	Text representations	Proposed
Persona models	DistilBERT [15] with Hugging Face Transformers [27]; Bi-LSTM	Persona classification and drift	Proposed
Sensitive detection	Regex with Luhn and entropy checks [30]; transformer NER [18]	Credentials, financial data, identifiers	Proposed
Anomaly model	Isolation Forest [5]	Contextual anomaly score	Proposed
Risk engine	Logistic fusion (7), scikit-learn	Risk score and actions	Proposed
Explainable AI	SHAP [19], LIME [20]	Per-alert and global explanations	Proposed
Ingestion	Message queue; channel	Collect email, chat, cloud events	Proposed

	connectors		
Profile store	Low-latency key-value store	Baselines, counters, thresholds	Proposed
Service layer	REST API (e.g., FastAPI)	Expose analysis as services	Proposed
Analyst console	Web dashboard	Ranked alerts, SHAP charts, feedback	Prototype evidence; integration/validation to be confirmed

TABLE II. TECHNOLOGY STACK, PURPOSE, AND DEVELOPMENT STATUS

B. Event-Level Processing Pipeline

(1) Capture and parse content, sender, recipients, channel, and time; (2) preprocess for the language branch; (3) embed and predict persona, computing P_i and $D_persona$; (4) detect sensitive content on minimally altered text and attach criticality; (5) compute contextual features and $A_context$; (6) score risk, computing $S_content$, S_freq , S_anom , and R ; (7) decide by the priority policy and role thresholds; (8) explain with SHAP and LIME and produce the rationale; and (9) enforce, log, and update profiles.

End-to-end latency is the sum of stage latencies:

$$T_{e2e} = T_{parse} + T_{prep} + T_{persona} + T_{detect} + T_{risk} + T_{xai} + T_{enforce} \quad (8)$$

Each term is logged separately so that bottlenecks can be identified before optimization. Cheap regular-expression checks can run first, and neural models can be applied to messages that pass the cheap filter or originate from higher-risk contexts. Explanations for low-risk events can be generated lazily so that they do not consume the interactive latency budget.

C. Deployment Workflow

Connectors place events on a message queue that decouples ingestion from analysis and absorbs bursts (Fig. 2). Stateless workers perform analysis behind an inference service with batching and, where needed, quantization. A low-latency profile store holds baselines, sliding-window counters, and thresholds. The enforcement layer returns the decision to the originating channel: low-risk messages pass transparently, medium-risk messages present the sender with an explanation and a justification prompt, and high-risk messages are quarantined and routed to an analyst queue. Analyst dispositions, such as true leak, false positive, or policy exception, can feed threshold calibration and periodic retraining. The system is intended to monitor data drift, score distributions, and per-class performance and to promote new models through shadow deployment before replacement of production models.

D. Worked Example

Suppose a finance analyst whose persona is usually routine collaborator sends an email containing a payment card number that passes the Luhn check and an API key to a personal external address at 11 p.m. The content branch would classify the detected items as high criticality; the persona engine could identify a shift toward an external-facing persona; and the contextual features could increase the anomaly score because of the hour and unfamiliar domain. Under the proposed policy, such an event is intended to reach the critical path and be quarantined for

analyst review. This is a design example, not an experimental result.

VII. EXPLAINABLE AND RESPONSIBLE DESIGN

The interaction and governance design follows principles intended to reduce analyst workload and protect employees, without claiming legal or regulatory compliance.

- Explanation with every action—Each warning, block, or escalation carries a short rationale and the top contributing factors with signed contributions.
- Concise alerts—Alerts state the data class, destination, persona deviation, and score rather than exposing raw model output.
- Avoiding repeated alerts—Cooldown windows, content fingerprints, and incident grouping suppress repetition of unchanged situations.
- Prioritization—The console ranks incidents by score and criticality.
- Uncertainty-aware decisions—When detection confidence is low or SHAP and LIME disagree strongly, the system prefers a warning or analyst review over an automatic block.
- Just-in-time education—Medium-risk prompts tell senders what was detected and why, reducing negligent leaks and discouraging workarounds.
- Human oversight—Blocking of high-impact communications can require analyst confirmation, and automated actions should have an appeal or reversal path.
- Purpose limitation—Persona labels are used only to assess data-movement risk, not to evaluate employee performance.
- Fairness auditing—Global SHAP summaries are reviewed to check that predictions do not depend on protected attributes or their proxies.
- Multilingual interaction—Language-specific models and identifier patterns can be selected per deployment region.

VIII. EXPERIMENTAL SETUP AND EVALUATION

No experimental results are reported in this paper. This section defines the evaluation methodology. Numerical values will be inserted only after experimental testing is completed, and every result marked “To be measured” remains pending.

A. Setup and Data

The evaluation will specify the compute platform, model variants and sizes, embedding sources, persona taxonomy, language set, and software versions so that results are reproducible. Real workplace leakage data are rarely shareable, so the plan uses publicly available corpora, namely the Enron email corpus [25] for message text and the CERT insider-threat datasets [3] for behavioral activity, with sensitive entities injected synthetically at controlled rates and with controlled paraphrasing. The limitations of synthetic injection will be reported explicitly. Persona labels will be produced with a documented annotation guideline by multiple annotators, and agreement will be reported. Data will be split by user and by time, with cross-validation, to reduce leakage between training and test sets.

Detection quantities follow the standard definitions:

$$P = TP/(TP + FP), \quad R = TP/(TP + FN), \quad F1 = 2PR/(P + R), \\ FPR = FP/(FP + TN) \quad (9)$$

Component	Metric	Testing method	Result
Persona prediction	Macro-F1, per-class precision and recall	Stratified k-fold; compare Word2Vec/GloVe + Bi-LSTM, BERT + Bi-LSTM, DistilBERT	To be measured
Persona drift	Detection delay, false-drift rate	Injected role-inconsistent sequences	To be measured
Sensitive detection	Entity-level precision, recall, F1	Regex-only, NER-only, hybrid on labelled and injected corpora	To be measured
Obfuscation robustness	Recall under paraphrase, encoding, splitting	Adversarially transformed content	To be measured
Risk scoring	AUROC, AUPRC, calibration error	Labelled or simulated incidents vs. rule baseline	To be measured
Alert quality	FPR, alerts per analyst per day, duplicate rate	Replay of event streams per configuration	To be measured
Missed incidents	False-negative rate at fixed alert budget	Injected leakage scenarios	To be measured
End-to-end latency	T_e2e and per-stage times (8)	Stage timestamps; median and high percentiles	To be measured
Throughput	Events per second	Sustained load on target hardware	To be measured
Explanation quality	Fidelity, stability, SHAP-LIME agreement	Perturbation tests, rank correlation	To be measured
Analyst usefulness	Triage time, usefulness rating	Small supervised study	To be measured
Threshold adaptation	Change in FPR and alert volume	Simulated feedback rounds	To be measured

TABLE III. EVALUATION METRICS AND TESTING METHODOLOGY

B. Ablation Configurations

To show the benefit of fusion, the following configurations will be compared on the same scenarios: (a) rule-based DLP only; (b) behavior anomaly only, without content; (c) the full system; (d) the full system without persona features; (e) the full system without NER, using patterns only; (f) the full system without adaptive thresholds; and (g) the full system without explanations, to measure their effect on analyst triage time.

C. Test Scenarios

#	Scenario	Purpose
1	Routine internal communication	Baseline; false alarms
2	Single sensitive item, authorized recipient	Detection without escalation
3	Credentials to personal external address	Critical detection and blocking
4	Bulk customer data in attachment	Document-level detection
5	Paraphrased or obfuscated content	Robustness beyond patterns
6	Data split across several messages	Frequency and history
7	Gradual persona drift, compromised account	Drift and poisoning resistance
8	Role change or new legitimate project	Adaptation; false-positive control
9	High alert volume burst	Grouping, throughput, latency
10	Mixed-language or code-mixed messages	Multilingual robustness

TABLE IV. PLANNED TEST SCENARIOS

D. Human Participation

Evaluation with security analysts or employee volunteers is outside the scope of the present work. Any future user study will require institutional ethical approval, informed consent, and clear disclosure of monitoring and will be reported separately. Initial testing is planned to use public and synthetic data only.

IX. RESULTS AND DISCUSSION

As measured results are not yet available, this section describes what will be measured, in which scenarios, and how performance will be analyzed.

A. Prototype Interface Evidence

The three prototype interfaces are shown in Figs. 7–9. They are retained as prototype evidence only; they are not used to claim detection accuracy, latency, or security performance.

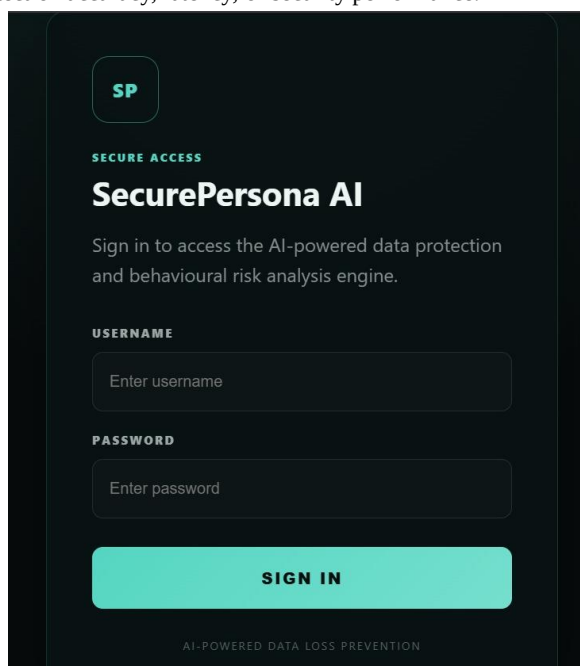


Fig. 7. Prototype sign-in interface.

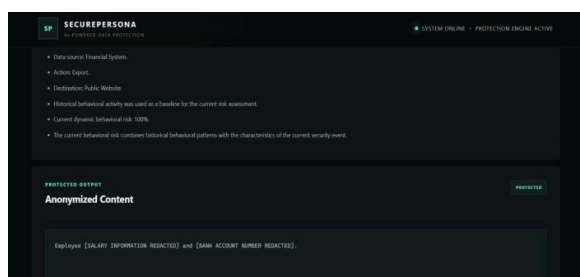


Fig. 8. Prototype message-analysis interface.

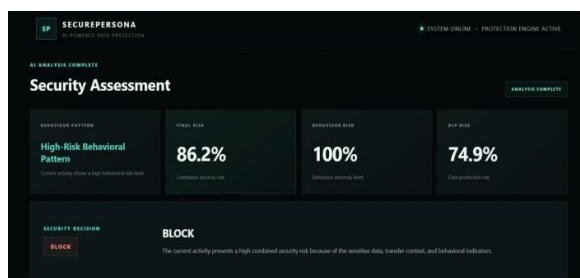


Fig. 9. Prototype security-assessment dashboard.

B. Analysis Plan

For each scenario, runs will be repeated and reported as mean, standard deviation, and, where sample size permits, a confidence interval. Persona and detection performance will be reported per class and per entity type. Because leakage is rare, precision–recall curves and false-positive rate at a fixed recall will be preferred over accuracy. Latency will be reported as distributions, including median and high percentiles, since worst-case delay matters for inline enforcement. Ablation comparisons will attribute differences to persona features, NER, adaptive thresholds, and explanations. Explanation quality will be analyzed through fidelity and rank agreement between SHAP and LIME.

C. Expected Limitations Affecting Performance

Persona labels are subjective and may transfer imperfectly from public corpora to workplace communication. Synthetic injection can overstate performance. Very short messages carry little persona signal. Obfuscation, images, and encrypted attachments can defeat text analysis. Role changes can resemble drift and cause false positives. Explanation methods are approximations and can be unstable under small perturbations. Failure cases should be reported alongside aggregate figures, and conclusions should not extend beyond the tested conditions.

X. PRIVACY, SECURITY AND LIMITATIONS

Privacy.

Monitoring employee communication is itself privacy-sensitive, and persona models learn from personal writing. The design therefore applies data minimization, masking of identifiers in stored text, role-based access to explanations, profiles, and logs, and defined retention and deletion periods. Differential privacy [21], [23] and federated learning [22] are candidate approaches for reducing centralization of raw text, at a possible utility cost [24]. Deployments should disclose monitoring in policy and obtain legal review under applicable regulations, including India's Digital Personal Data Protection Act, 2023 [28] and the General Data Protection Regulation [29].

Security of the system itself.

The profile store, models, and logs are high-value targets. Access should be authenticated and audited, models should be protected from tampering, and baselines should be updated only from events judged benign to resist poisoning. Adversaries who learn the system's behavior may try to mimic a benign persona or stay below thresholds, which is addressed in Scenarios 5–7.

Technical limitations.

Text-only analysis misses leakage in images, audio, and encrypted content. Transformer inference adds latency and cost. A false negative lets a leak pass, while a false positive interrupts legitimate work, so both are operationally significant. Persona taxonomies require maintenance as roles evolve. Explanation methods describe model behavior approximately and are not proof of intent.

Safety statement.

The system is a decision-support aid and not a replacement for security policy, access control, encryption, user training, or human judgment. It is not production-ready, and no guarantee of detection is made. Automated blocking of critical communication should be introduced gradually, with human review and an appeal path, and persona predictions must not be used as evidence of wrongdoing without independent investigation.

XI. FUTURE SCOPE

Planned improvements include multilingual and code-mixed support through multilingual distilled transformers and region-specific identifier patterns, including Indian languages; multi-format stream analysis covering images through optical character recognition, scanned documents, audio and video transcripts, and structured data exports; federated and differentially private training of persona models; adversarial robustness testing against evasion and baseline poisoning; richer behavioral signals from access logs and communication graphs; domain-adapted language models trained on organizational text; on-premises and edge deployment of compact models; improved uncertainty estimation and calibration of risk scores; automated policy recommendations from analyst feedback; and, subject to ethical approval, longitudinal studies with security analysts to measure alert fatigue and trust.

XII. CONCLUSION

This paper addressed the gap between content-centric DLP, which finds sensitive patterns but not the behavioral context of the sender, and behavior analytics, which models users but not content. We proposed an AI-based secure persona prediction and data leakage prevention system that fuses transformer and Bi-LSTM persona inference, hybrid NER and pattern-based sensitive data detection, channel context, and dynamic risk scoring through a priority-based decision policy with alert management, while attaching SHAP and LIME explanations to decisions. The responsible-design principles include data minimization, purpose limitation, human oversight, and fairness auditing. The manuscript distinguishes proposed components from prototype interface evidence and deliberately reports no performance figures until experiments are completed. An evaluation plan with metrics, ablations, and ten scenarios is defined. Future work will complete integration, execute the evaluation, extend language and format coverage, and, with appropriate approval, involve security practitioners in assessing usefulness.

ACKNOWLEDGMENT

The authors thank their project guide, Prof. Krupali Dhawale, and G H Raisoni College of Engineering, Nagpur, for guidance and support.

REFERENCES

[1] A. Shabtai, Y. Elovici, and L. Rokach, *A Survey of Data Leakage Detection and Prevention Solutions*, SpringerBriefs in Computer Science. New York, NY, USA: Springer, 2012.
 [2] S. Alneyadi, E. Sithirasanen, and V. Muthukkumarasamy, "A survey on data leakage prevention

systems," *J. Netw. Comput. Appl.*, vol. 62, pp. 137–152, Feb. 2016.
 [3] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *Proc. IEEE Security and Privacy Workshops (SPW)*, 2013, pp. 98–104.
 [4] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *Proc. AAAI Workshop on AI for Cyber Security*, 2017, arXiv:1710.00811.
 [5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.
 [6] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," *J. Artif. Intell. Res.*, vol. 30, pp. 457–500, 2007.
 [7] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artif. Intell. Rev.*, vol. 53, pp. 2313–2339, 2020.
 [8] A. Naz, H. U. Khan, T. Alsahfi, M. Alhajlah, B. Alshemaimri, and A. Daud, "Using transformers and Bi-LSTM with sentence embeddings for prediction of openness human personality trait," *PeerJ Comput. Sci.*, vol. 11, e2781, 2025, doi: 10.7717/peerj-cs.2781.
 [9] K. Fawaz, R. Yi, O. Suciu, R. Khandelwal, H. Harkous, N. Taft, and M. Gruteser, "Text-based personas for simulating user privacy decisions," arXiv:2603.19791, 2026.
 [10] S. Lim, S. Lee, D. Min, and Y. Yu, "Persona Dynamics: Unveiling the Impact of Persona Traits on Agents in Text-Based Games," in *Proc. 63rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 31360–31394, 2025, doi: 10.18653/v1/2025.acl-long.1515.
 [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
 [12] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
 [13] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.
 [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
 [15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019.
 [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
 [17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

- [18] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in Proc. NAACL-HLT, 2016, pp. 260–270.
- [19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 4765–4774.
- [20] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016, pp. 1135–1144.
- [21] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2016, pp. 308–318.
- [22] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. AISTATS, ser. PMLR, vol. 54, pp. 1273–1282, 2017.
- [23] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Found. Trends Theor. Comput. Sci., vol. 9, no. 3–4, pp. 211–407, 2014.
- [24] S. Meisenbacher, M. Chevli, J. Vladika, and F. Matthes, "DP-MLM: Differentially private text rewriting using masked language models," in Findings of the Association for Computational Linguistics: ACL 2024, pp. 9314–9328, 2024, doi: 10.18653/v1/2024.findings-acl.554.
- [25] B. Klimt and Y. Yang, "The Enron corpus: A new dataset for email classification research," in Machine Learning: ECML 2004, ser. LNCS, vol. 3201, pp. 217–226, 2004.
- [26] J. Lin, "Divergence measures based on the Shannon entropy," IEEE Trans. Inf. Theory, vol. 37, no. 1, pp. 145–151, Jan. 1991.
- [27] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP: System Demonstrations, 2020, pp. 38–45.
- [28] Government of India, "The Digital Personal Data Protection Act, 2023," Act No. 22 of 2023.
- [29] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Off. J. Eur. Union, L 119, 2016.
- [30] H. P. Luhn, "Computer for verifying numbers," U.S. Patent 2 950 048, Aug. 23, 1960.