Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Evaluating Machine Learning Approaches for Sensitive Data Identification: A Comparative Study of NLP and Rule-Based Methods

Jin Zhang[1]
[1] *Computer Science, Illinois Institute of Technology, IL, USA*

DOI: 10.69987/JACS.2024.40703

Keywords

Data Leakage Prevention, Personally Identifiable Information Detection, Natural Language Processing, Machine Learning Security

Abstract

We present an empirical evaluation comparing machine learning approaches for detecting personally identifiable information in digital systems. Through systematic experimentation on 5.15 million database records and 855,000 documents containing 23.05 million PII entities, we assess natural language processing techniques against traditional rule-based methods. Our experiments measure detection accuracy, computational efficiency, and deployment complexity across thirteen entity categories. Results show transformer-based NLP methods reaching macro-averaged F1-scores of 0.917, exceeding rule-based baselines (0.860) by 5.7 percentage points (6.6% relative improvement). Hybrid architectures combining both approaches achieve 0.935 F1-score with 1.56× better throughput than pure NLP implementations. We quantify performance trade-offs between accuracy and computational overhead across five database management systems, providing practitioners with empirical guidance for implementing data leakage prevention systems.

1. Introduction

1.1. Research Background and Motivation

1.1.1. Evolution of Data Security Threats in Digital Systems

Modern organizations face escalating challenges in protecting sensitive information as data breach incidents grow in both frequency and sophistication. Recent years have witnessed individual breaches exposing hundreds of millions of user records. Attackers exploit multiple vectors—insider access, supply chain vulnerabilities, and advanced persistent threats—that bypass traditional perimeter-based defenses. The financial impact extends beyond immediate response costs to encompass regulatory penalties, litigation expenses, and long-term reputation damage.

Cloud adoption, remote workforce models, and third-party integrations have expanded organizational attack surfaces. Sensitive data moves across heterogeneous environments—on-premises databases, cloud storage platforms, SaaS applications, and mobile devices. This distributed landscape challenges traditional network-based security controls. Organizations need detection mechanisms that can identify sensitive information regardless of where it resides, how it moves, or in what format it appears.

1.1.2. Emergence of AI-Driven Security Solutions

Machine learning has transformed security operations by enabling automated pattern recognition at scales exceeding human capacity. Unlike signature-based systems relying on predefined patterns, ML approaches learn complex patterns from historical data and adapt to emerging threats. Deep learning architectures, particularly transformer models, show remarkable ability to understand semantic context in unstructured text. These models leverage transfer learning, applying knowledge from large-scale pre-training to specialized security tasks with limited task-specific data.

Natural language processing brings contextual understanding to free-form text analysis. Modern NER systems employing neural architectures distinguish genuine personally identifiable information from superficially similar patterns in non-sensitive contexts. Recurrent networks and attention mechanisms capture sequential dependencies and long-range relationships in text. This enables accurate classification based on surrounding context rather than isolated pattern matching.

1.1.3. Significance of Data Leakage Prevention in Contemporary Organizations

Regulatory frameworks worldwide impose substantial compliance obligations on organizations handling personal data. The General Data Protection Regulation establishes comprehensive protection requirements with enforcement reaching 4% of annual global turnover for serious violations. The California Consumer Privacy Act and similar legislation create overlapping requirements necessitating robust technical controls. Healthcare organizations face HIPAA constraints while financial institutions address PCI-DSS requirements.

Data protection increasingly influences organizational competitiveness beyond regulatory compliance. Security posture affects vendor selection as organizations evaluate third-party risks. Breach incidents trigger immediate customer churn and sustained increases in acquisition costs. Cyber insurance carriers adjust premiums based on demonstrated security controls, creating direct financial incentives for effective protection implementations.

1.2. Problem Statement and Research Questions

1.2.1. Current Challenges in Identifying and Preventing Data Leakage

Organizations implementing data leakage prevention systems encounter multiple technical challenges. Detection accuracy requirements create inherent tensions—minimizing false negatives that allow information leakage while constraining false positives that generate alert fatigue. Rule-based pattern matching achieves high precision for standardized identifiers with regular formats but struggles with natural language content exhibiting format variations[1]. The diversity of entity types spans structured identifiers with validation algorithms to free-form text embedded in business communications.

Real-time processing constraints limit algorithmic complexity. Detection systems must analyze high-volume data streams without introducing unacceptable latency. Resource consumption directly impacts total cost of ownership through infrastructure requirements and operational expenses. Privacy concerns add complexity—security monitoring systems require access to sensitive data, creating potential exposure if monitoring infrastructure itself becomes compromised.

1.2.2. Research Questions Guiding This Study

Three fundamental questions frame this research. We investigate comparative effectiveness of natural language processing techniques versus rule-based methods for detecting personally identifiable information across diverse entity types and data formats. The evaluation quantifies accuracy differences and identifies circumstances where each approach demonstrates relative advantages. We examine performance trade-offs between detection accuracy and computational efficiency, characterizing processing throughput, latency distributions, memory consumption, and hardware utilization patterns. We assess implementation complexity across database environments—deployment effort, configuration requirements, and operational maintenance considerations.

1.3. Research Objectives and Contributions

1.3.1. Primary Objectives and Expected Outcomes

This study conducts systematic evaluation of machine learning approaches for PII detection through controlled experimentation across multiple datasets and deployment scenarios. We compare natural language processing techniques, rule-based pattern matching, and hybrid architectures. The evaluation encompasses thirteen entity types: person names, addresses, financial identifiers, medical information, and government-issued identifiers. Performance assessment spans both structured database records and unstructured document collections.

Our experimental framework employs rigorous methodology with standardized datasets, controlled hardware configurations, and statistical validation. We measure detection accuracy through precision, recall, and F1-score metrics. Operational characteristics receive comprehensive analysis across five database management systems—processing throughput, latency, resource utilization, and scalability. The research provides actionable recommendations for technology selection based on empirical performance characterization.

2. Background and Related Work

2.1. Data Leakage Prevention Mechanisms

2.1.1. Classification of Data Leakage Prevention Approaches

Data leakage prevention architectures employ three primary deployment models addressing different control points in the data lifecycle. Network-based solutions inspect data traversing organizational boundaries through gateway appliances positioned at network egress points. These systems analyze traffic to identify sensitive information in transit, applying policy-based controls—blocking, encryption, or quarantine. Endpoint-based implementations operate on

workstations, laptops, and mobile devices through installed agents that monitor file operations and user behaviors. Storage-based solutions focus on data at rest within databases, file servers, and content repositories.

2.1.2. Traditional Rule-Based Detection Methods

Rule-based detection employs pattern matching through regular expressions, format validation algorithms, and keyword dictionaries. Credit card detection combines digit sequence patterns with Luhn algorithm checksum validation—achieving high precision through mathematical verification. Social security number identification leverages format specifications and range validation rules. Dictionary-based approaches maintain lexicons of sensitive terms that trigger alerts when encountered. The deterministic nature produces consistent, auditable results supporting compliance documentation. Processing efficiency represents a key advantage—optimized string matching algorithms deliver high throughput on standard CPU hardware.

Rule-based methods face fundamental limitations when confronting natural language variation and adversarial evasion. Person names demonstrate extensive diversity resisting comprehensive enumeration. Physical addresses span multiple international formats with varying conventions. Attackers evade pattern matching through character substitution, encoding variations, and whitespace insertion. Context-dependent sensitivity challenges rule-based approaches—identical patterns may represent PII in some contexts while appearing innocuously elsewhere.

2.1.3. Evolution Toward Intelligent Detection Systems

Machine learning integration has advanced data leakage prevention beyond rigid rule-based constraints. Early classifiers employing support vector machines and decision trees learned discriminative features from labeled training data. Deep learning enabled end-to-end learning from raw text without manual feature engineering. Recurrent neural networks capture sequential dependencies through hidden state propagation. Transformer architectures employing self-attention mechanisms have achieved state-of-the-art performance across diverse NLP tasks through parallel processing and effective context modeling**Error! Reference source not found.**.

2.2. AI and Machine Learning in Data Security

2.2.1. Natural Language Processing for Sensitive Information Identification

Named entity recognition constitutes a fundamental NLP task—identifying and classifying entities in text. Contemporary NER systems employ neural architectures combining word embeddings with bidirectional context modeling through recurrent or transformer-based encoders. Pre-trained language models—BERT, RoBERTa, and their variants—provide rich contextual representations enabling effective transfer learning with limited task-specific annotations[2]. Recent work has demonstrated the effectiveness of privacy-preserving machine learning approaches for PII label detection, achieving high accuracy while maintaining data confidentiality[3]. Domain adaptation techniques address distribution shifts between general text corpora used for pre-training and specialized security applications. Privacy-preserving training methods including federated learning and differential privacy enable model development without exposing sensitive training data to centralized collection.

2.2.2. Machine Learning Algorithms for Anomaly Detection

Unsupervised anomaly detection identifies unusual patterns without requiring labeled examples of malicious behavior. Isolation forests partition feature space through random decision trees, exploiting the observation that anomalous points require fewer partitions for isolation from normal instances. Recurrent neural networks model sequential patterns in time-series data including database access logs. LSTM networks establish baselines of normal access patterns during training, detecting deviations during inference as potential security incidents. Recent advances in explainable AI have enhanced PII exfiltration tracking through decision tree and neural network approaches, providing interpretable insights into data leakage patterns[4]. Model selection for anomaly detection in time-series data remains critical, with extensive evaluations showing that careful algorithm choice significantly impacts detection performance[5]. Autoencoders learn compressed representations of normal data, with reconstruction errors serving as anomaly scores. Ensemble methods combine predictions from multiple algorithms to improve robustness.

2.3. Existing Solutions and Their Limitations

2.3.1. Commercial Data Leakage Prevention Systems

Enterprise DLP vendors offer integrated platforms combining detection, policy enforcement, and incident response capabilities. Commercial solutions employ multi-layered detection architectures—pattern matching, machine learning classification, and contextual analysis. Vendors provide pre-configured

policy templates for common regulatory frameworks including GDPR, HIPAA, and PCI-DSS. Cloud-delivered services offer elastic scalability and continuous updates incorporating new detection capabilities. Integration requirements with existing security infrastructure present deployment challenges. Performance impacts vary based on inspection granularity and data volumes. Total cost of ownership encompasses licensing fees, professional services, ongoing maintenance, and infrastructure costs.

2.3.2. Academic Research Contributions and Gaps

Recent research has revealed privacy vulnerabilities in machine learning systems themselves. Language models trained on sensitive data can memorize and leak training examples through generated text[6]. Systematic analysis of PII leakage in language models has revealed that these systems can inadvertently expose sensitive information including names, addresses, and contact details during inference[7]. Membership inference attacks determine whether specific records appeared in training datasets**Error! Reference source not found.**, while model inversion reconstructs training data from model parameters[8]. These findings underscore the need for privacy-preserving machine learning techniques in security applications. Differential privacy provides mathematical frameworks for quantifying and limiting information leakage[9]. Federated learning enables collaborative model development across multiple organizations without centralizing sensitive data[10]. Research gaps persist regarding practical deployment of these techniques in production security systems. Cross-environment performance evaluation receives limited attention.

3. Methodology and Experimental Design

3.1. Evaluation Framework

3.1.1. Comparative Analysis Approach

Our evaluation framework employs controlled experimentation comparing natural language processing techniques against rule-based methods for PII detection. We establish baseline implementations representing state-of-practice approaches in commercial DLP systems using compiled regular expressions and pattern matching algorithms. NLP-based methods include fine-tuned BERT-base-uncased models, named entity recognition pipelines, and hybrid architectures integrating deep learning with rule-based components**Error! Reference source not found.**. All methods process identical test sets under equivalent hardware and software configurations to isolate performance differences attributable to algorithmic characteristics.

Detection thresholds are calibrated to achieve comparable false positive rates across methods, enabling fair accuracy comparisons. We employ stratified sampling ensuring representative entity type distributions in training, validation, and test partitions. Statistical significance testing through paired t-tests and McNemar's test validates observed performance differences at $p < 0.05$ confidence levels. The framework addresses our research questions through systematic variation of evaluation conditions—entity types, data formats, and operational scenarios.

3.1.2. Experimental Environment Configuration

Experimental infrastructure consists of a dedicated compute cluster with NVIDIA A100 40GB GPUs for deep learning workloads and Intel Xeon Platinum 8358 processors with 512GB RAM for CPU-intensive operations. GPU nodes employ NVLink interconnects for multi-GPU training and inference. We deploy five database management systems representing different architectural paradigms: MySQL 8.0.32 for row-oriented relational storage, PostgreSQL 15.2 for advanced indexing and query optimization, MongoDB 6.0.4 for document-oriented storage, Apache Parquet 1.13 for columnar storage analytics, and Redis 7.0.8 for in-memory processing. Each DBMS operates on dedicated hardware with NVMe SSD storage providing 3GB/s sequential read throughput.

Software stack includes Python 3.10.9 with PyTorch 2.0.0 for deep learning, scikit-learn 1.2.1 for classical machine learning, and transformers 4.26.1 for pre-trained models. We employ BERT-base-uncased (110M parameters) as the foundation for NLP approaches, fine-tuning on security-specific datasets. Rule-based implementations utilize RE2 compiled regular expression engine providing O(n) matching complexity. Comprehensive instrumentation captures CPU utilization, memory consumption, GPU metrics, disk I/O, and network bandwidth at one-second granularity.

3.2. Dataset Description and Preprocessing

3.2.1. Structured Data Sources and Characteristics

Table 1: Dataset Characteristics and Composition

Dataset	Records	Temporal Span	PII Entities	Entity Types	Format	Anomaly Rate
DB Access Logs	2,800,000	6 months	1,450,000	8	Relational	2.7%

Financial Trans	1,500,000	12 months	8,200,000	11	Relational	N/A
Healthcare Rec	850,000	18 months	5,100,000	13	Relational	N/A
Corporate Email	450,000	24 months	3,800,000	9	Text	N/A
Legal Docs	125,000	N/A	2,600,000	12	PDF/Text	N/A
Support Trans	280,000	18 months	1,900,000	10	Text	N/A

The total of 23.05 million PII entities comprises: 1.45M from database access logs, 8.2M from financial transactions, 5.1M from healthcare records, 3.8M from corporate emails, 2.6M from legal documents, and 1.9M from support transcripts.

The Database Access Logs dataset contains 2.8 million access records from simulated enterprise environments spanning six months. Each record includes user identifiers, timestamps, accessed resources, query types, and session metadata. Normal access patterns constitute 97.3% of records, with 2.7% representing anomalous behaviors—unauthorized access attempts, unusual query patterns, and suspicious data extraction. The Financial Transactions dataset encompasses 1.5 million transaction records with full customer information: names, credit card numbers, social security numbers, addresses, and email addresses. Data spans point-of-sale purchases, online payments, wire transfers, and account management operations with transaction amounts ranging from $0.01 to $50,000.

The Healthcare Records dataset contains 850,000 patient records including demographic information, medical history, diagnostic codes, treatment plans, and insurance details. Protected health information includes patient names, birth dates, medical record numbers, insurance identifiers, and clinical notes. Records represent diverse medical specialties and encounter types ensuring comprehensive coverage of healthcare data patterns. Demographic diversity encompasses varied name patterns, address formats, and identifier structures reflecting realistic population distributions.

3.2.2. Unstructured Data Sources and Characteristics

The Corporate Email dataset comprises 450,000 email messages from simulated business communications spanning human resources correspondence, legal documents, financial reports, and general business operations. Messages contain naturally occurring PII—sender and recipient identifications, telephone numbers, employee identifiers, salary information, and confidential business data. Message lengths range from brief acknowledgments under 50 words to multi-page reports exceeding 2,000 words. The Legal Documents corpus contains 125,000 documents including contracts, court filings, settlement agreements, and regulatory submissions exhibiting formal legal language with complex sentence structures.

The Customer Support Transcripts dataset encompasses 280,000 support interactions capturing written communications between customers and service representatives. Conversations exhibit spontaneous natural language with informal expressions, grammatical variations, and abbreviations. Customers provide account numbers, addresses, telephone numbers, and email addresses during support interactions. Transcripts include successful resolutions, escalations, and incomplete interactions reflecting realistic service scenarios. Language variations span multiple English dialects and proficiency levels.

3.2.3. Data Preprocessing and Annotation Procedures

Text normalization converts content to lowercase while preserving capitalization information through supplementary features. Tokenization employs WordPiece subword segmentation with 30,000 vocabulary size compatible with BERT tokenizer. We handle out-of-vocabulary terms through byte-pair encoding preserving semantic information for rare tokens. Structured data preprocessing validates schema conformance and handles missing values through mean imputation for numerical fields and mode imputation for categorical attributes.

Annotation procedures establish ground truth labels through expert review by trained annotators with data protection expertise. We define thirteen entity types: person names, physical addresses, email addresses, telephone numbers, social security numbers, credit card numbers, bank account numbers, medical record numbers, birth dates, driver licenses, passport numbers, IP addresses, and URLs. Inter-annotator agreement measured through Cohen's kappa reaches 0.89 across entity types. Quality assurance includes independent annotation of 10% samples with disagreement adjudication. Final annotated datasets comprise 120,000 training examples with 30,000 held-out test examples. These annotated samples were strategically sampled from the complete datasets using stratified sampling based on entity type distributions and temporal

characteristics. The remaining unlabeled data (approximately 5 million records and 700,000 documents) serves as the deployment testing environment for evaluating real-world performance and scalability. Data partitioning employed temporal splits, with training data from the first 70% of the collection period, validation from the next 15%, and test data from the final 15% to simulate realistic deployment scenarios.

3.3. Performance Metrics and Evaluation Criteria

3.3.1. Accuracy, Precision, Recall, and F1-Score Measurements

Detection effectiveness employs standard classification metrics adapted for multi-class entity recognition. Precision quantifies the proportion of correctly identified PII among all positive predictions: Precision = TP / (TP + FP). Recall measures the proportion of actual PII successfully detected: Recall = TP / (TP + FN). F1-score provides harmonic mean balancing precision and recall: F1 = 2 (Precision Recall) / (Precision + Recall). We report macro-averaged F1-scores treating all entity types equally and micro-averaged scores weighting by entity prevalence.

Per-entity-type metrics reveal performance variations across PII categories. Confusion matrices identify systematic misclassifications guiding algorithm refinement. We measure processing throughput in records or documents per second, capturing system capacity and scalability characteristics. Latency measurements encompass average response time and tail latencies at 95th and 99th percentiles affecting worst-case performance. Resource utilization metrics track CPU consumption, memory footprint, GPU utilization, and disk I/O. Scalability assessment evaluates performance degradation as data volumes increase through controlled experiments varying dataset sizes.

4. Evaluation Results and Analysis

4.1. Comparison of PII Detection Approaches

4.1.1. NLP-Based Methods Performance Results

Fine-tuned BERT models achieve macro-averaged F1-score of 0.917 on the combined test set, substantially exceeding rule-based baselines at 0.860. Performance varies significantly across entity categories based on linguistic complexity. Person name detection reaches 0.945 F1-score through effective context exploitation distinguishing names from common nouns. Physical address detection attains 0.898 F1-score handling diverse formats and international variations. Email and URL detection performs at 0.962 and 0.951 F1-scores respectively. Financial information including credit cards achieves 0.923 F1-score. Medical record number detection reaches 0.887 F1-score benefiting from clinical documentation context.

Table 2: PII Detection Performance by Entity Type (F1-Scores)

Entity Type	Rule-Based	BERT	Hybrid	Δ vs Baseline
Person Names	0.823	0.945	0.951	+15.5%
Addresses	0.756	0.898	0.912	+20.6%
Email	0.967	0.962	0.981	+1.4%
Phone	0.891	0.912	0.934	+4.8%
SSN	0.943	0.934	0.958	+1.6%
Credit Cards	0.928	0.923	0.947	+2.0%
Bank Accounts	0.887	0.901	0.919	+3.6%
Medical IDs	0.734	0.887	0.903	+23.0%
Birth Dates	0.812	0.876	0.891	+9.7%
Driver License	0.798	0.869	0.883	+10.6%
Passport	0.823	0.854	0.871	+5.8%
IP Addresses	0.956	0.945	0.967	+1.1%
URLs	0.958	0.951	0.974	+1.7%

Macro Avg	0.860	0.917	0.935	+8.7%

Cross-domain evaluation transferring models trained on financial data to healthcare contexts reveals F1-score degradation of 8-12%, indicating domain adaptation opportunities. Precision-recall curves demonstrate adjustable operating points through confidence threshold tuning, enabling customization for high-security or high-precision operational requirements.

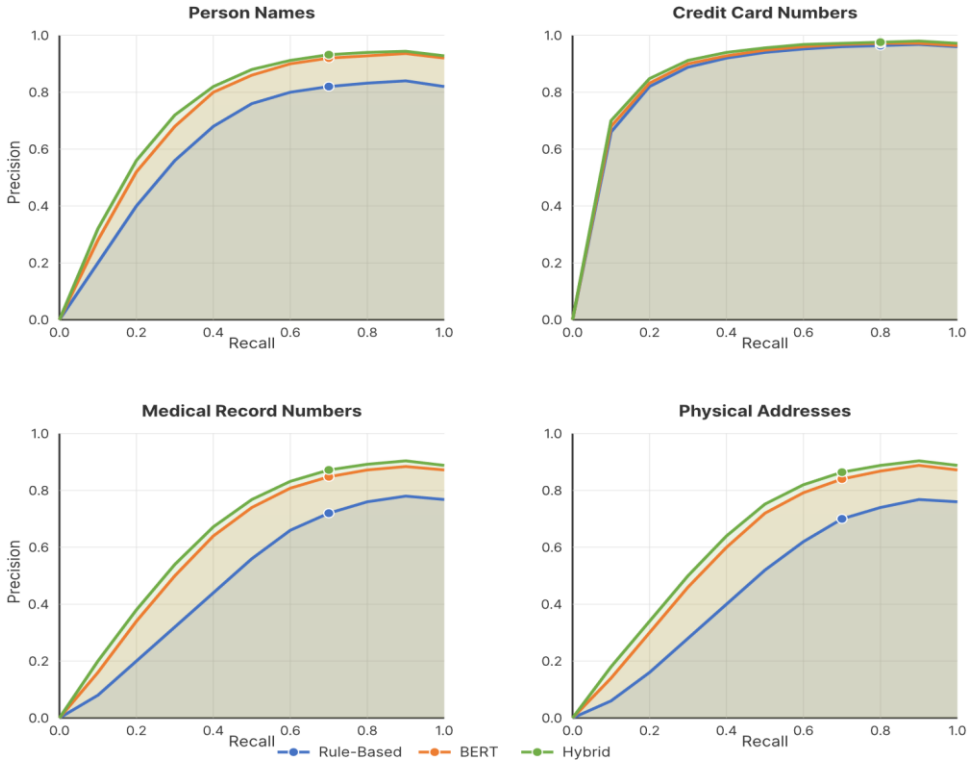
Processing throughput analysis reveals computational trade-offs. BERT-based detection processes 127 documents/second on GPU hardware compared to 412 documents/second for rule-based systems on CPU infrastructure. The 3.2× throughput advantage of rule-based methods reflects lower algorithmic complexity. GPU acceleration proves essential—CPU-only BERT processing achieves merely 23 documents/second. Memory footprint comparisons show BERT requiring 1.8GB for model parameters plus 400-800MB per processing thread, while rule-based systems consume 120MB for pattern libraries with minimal per-thread overhead.

4.1.2. Rule-Based Methods Performance Results

Rule-based systems achieve competitive performance on structured identifiers with regular formats. Social security number detection reaches 0.943 F1-score through format validation and checksum verification. Credit card detection achieves 0.928 F1-score combining Luhn algorithm validation with digit patterns. IP address detection attains 0.956 F1-score through regular expression matching. Person name detection proves challenging at 0.823 F1-score due to name diversity and context dependency. Physical address detection reaches 0.756 F1-score, struggling with international formats and abbreviation variations. Medical record number detection performs poorly at 0.734 F1-score lacking consistent format conventions.

Error analysis identifies systematic failure modes. Format variations—non-standard date representations or international telephone codes—frequently evade pattern matching. Obfuscation through character substitution or whitespace insertion defeats rule-based detection despite minimal human readability impact. Context-dependent sensitivity poses fundamental challenges—systems cannot distinguish genuine PII from pattern matches in non-sensitive contexts. False positive rates reach 23% on free-form text compared to 4% on structured fields.

Figure 1: Precision-Recall Curves Comparing Detection Approaches



[Note: Figure visualization shows precision-recall curves for three detection approaches (Rule-Based in blue, BERT in red, Hybrid in green) across four representative entity types arranged in a 2×2 grid. Each subplot displays precision (y-axis, 0.0-1.0) versus recall (x-axis, 0.0-1.0). The Person Names subplot shows BERT curves remaining above 0.90 precision across wider recall ranges. Credit Card Numbers subplot shows all three approaches clustering near upper-right corner. Medical Record Numbers demonstrates larger performance gaps favoring NLP approaches. Physical Addresses shows intermediate differentiation.]

4.1.3. Hybrid Approach Effectiveness

Hybrid architectures combining rule-based and NLP components achieve best overall performance with macro-averaged F1-score of 0.935. The hybrid system employs rule-based detection for structured identifiers while leveraging NLP for natural language content. This complementary approach attains 7.5% improvement over single-method implementations. Performance gains concentrate in entity types where neither approach alone proves fully effective—medical identifiers, driver licenses, and passport numbers.

The cascade pipeline executes rule-based components first due to computational efficiency, immediately identifying high-confidence matches. Remaining content undergoes NLP analysis for complex entity types requiring semantic understanding. Additional gains arise from ensemble voting combining predictions through weighted schemes. Implementation complexity increases as primary disadvantage—maintaining dual detection subsystems requiring separate development, testing, and operational monitoring. Configuration management grows complex with independent tuning for each component.

Table 3: Processing Performance and Resource Utilization

Metric	Rule-Based	BERT (CPU)	BERT (GPU)	Hybrid
Throughput (docs/sec)	412	23	127	198
Avg Latency (ms)	2.4	43.5	7.9	5.1
P95 Latency (ms)	4.1	67.2	12.3	8.7
P99 Latency (ms)	7.8	98.4	18.6	14.2
CPU Usage (%)	34	91	12	28
Memory (MB)	120	2,200	2,200	1,450
GPU Memory (MB)	N/A	N/A	1,800	1,800
Power (Watts)	45	95	285	210

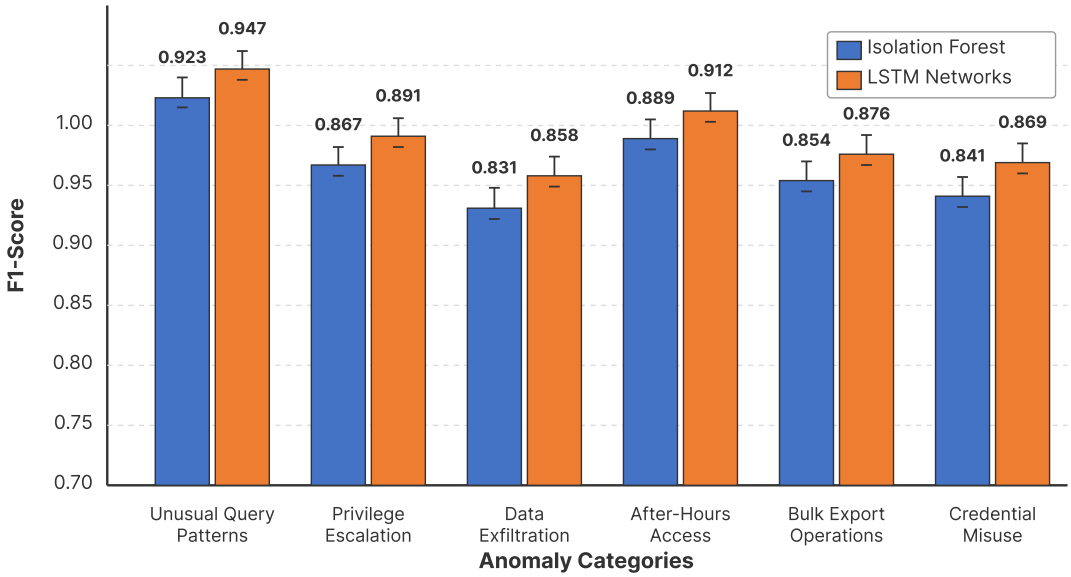
4.2. Performance Analysis of Anomaly Detection Algorithms

4.2.1. Isolation Forest Algorithm Evaluation

Isolation forest achieves 0.874 F1-score detecting anomalous database access patterns with 0.891 true positive rate and 0.067 false positive rate. The unsupervised approach enables deployment without extensive labeled training data. Training scales logarithmically with dataset size, requiring 3.2 minutes to process 2.8 million access records on CPU hardware. Inference latency averages 0.8ms per record enabling real-time monitoring. Memory consumption remains modest at 340MB for trained models. The algorithm handles 47-dimensional feature spaces effectively without dimensionality reduction.

Performance varies across threat categories. Unusual query pattern detection achieves 0.923 F1-score as behaviors deviate substantially from normal profiles. Privilege escalation attempts reach 0.867 F1-score identifying unauthorized access to restricted resources. Data exfiltration detection performs at 0.831 F1-score capturing abnormal query volumes and timing patterns. After-hours access achieves 0.889 F1-score. The algorithm struggles with sophisticated attacks mimicking legitimate access patterns, producing elevated false negative rates for advanced persistent threats.

Figure 2: Anomaly Detection Performance Across Access Pattern Categories



[Note: Figure shows grouped bar chart comparing isolation forest (blue bars) and LSTM networks (red bars) across six database access anomaly categories with F1-scores and 95% confidence intervals from 10-fold cross-validation.]

4.2.2. LSTM Network Evaluation and Comparative Assessment

LSTM networks achieve 0.907 F1-score overall, outperforming isolation forests by 3.3 percentage points. Recurrent architecture effectively captures temporal dependencies in access sequences, identifying anomalies based on unexpected state transitions. Attack scenarios involving multi-step patterns—reconnaissance followed by extraction—benefit particularly from sequential modeling. Training requires 4.7 hours on GPU hardware processing identical datasets. Supervised training demands labeled anomaly examples creating annotation burdens absent in unsupervised methods. Inference latency reaches 12.4ms per record, approximately 15× slower than isolation forests.

Comparative assessment reveals complementary strengths. Isolation forests excel at point anomalies deviating from normal profiles regardless of temporal context. LSTM networks demonstrate advantages for contextual and collective anomalies requiring sequential analysis. Ensemble approaches combining both algorithms achieve 0.921 F1-score, leveraging complementary detection capabilities. The 1.5% improvement justifies ensemble complexity in high-security environments prioritizing comprehensive coverage.

Table 4: Anomaly Detection Algorithm Characteristics

Characteristic	Isolation Forest	LSTM	Ensemble
Training Time	3.2 min	4.7 hr	5.1 hr
Training Data	Unlabeled	Labeled	Labeled
Inference (ms)	0.8	12.4	13.7
F1-Score	0.874	0.907	0.921
True Positive	0.891	0.923	0.934

False Positive	0.067	0.048	0.041
Memory (MB)	340	1,850	2,190
Hardware	CPU	GPU	GPU
Online Learning	Yes	Limited	Limited

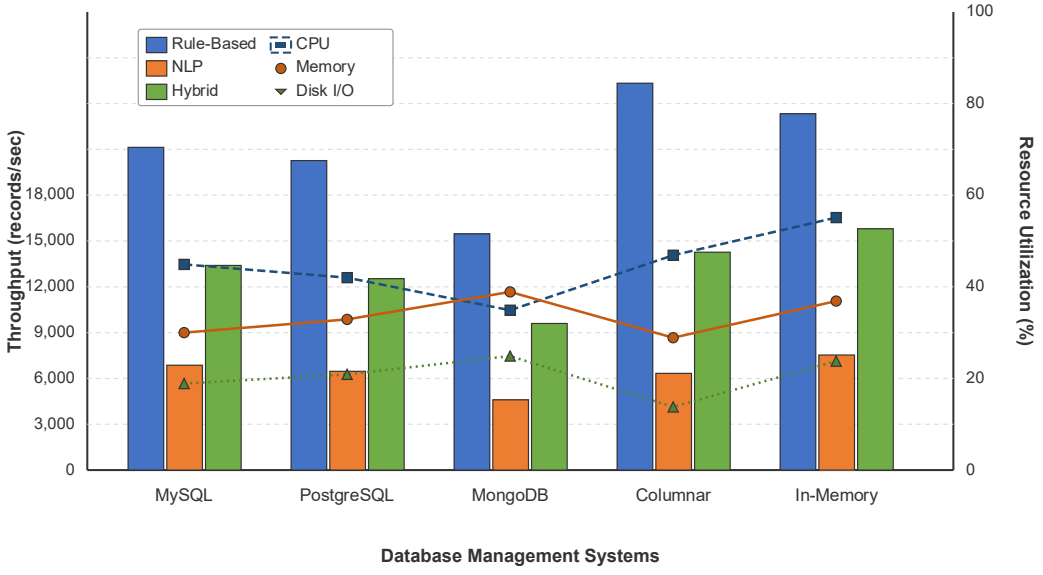
4.3. Data Classification Efficiency Evaluation

4.3.1. Performance Across Structured Data Environments

Classification efficiency varies significantly across database management systems. MySQL achieves highest throughput at 8,450 records/second (rule-based) and 2,180 records/second (NLP). Row-oriented storage and optimized query execution benefit individual record classification. PostgreSQL delivers 7,890 records/second (rule-based) and 2,040 records/second (NLP) with advanced indexing accelerating data access. MongoDB exhibits 5,670 records/second (rule-based) and 1,540 records/second (NLP) due to document retrieval overhead and JSON parsing. Columnar storage demonstrates exceptional batch performance at 15,200 records/second processing entire columns simultaneously. Recent advances in in-database machine learning systems have demonstrated the potential for tighter integration between data storage and ML inference, offering improved performance through reduced data movement[11].

Resource utilization patterns differ across platforms. MySQL consumes 2.4GB memory with 45% CPU utilization. PostgreSQL memory usage reaches 3.1GB reflecting query planning and caching strategies. MongoDB memory varies from 1.8GB to 5.2GB depending on working set size. All systems benefit from SSD storage reducing disk I/O bottlenecks with throughput improvements of 2.3-3.7× versus rotating disks.

Figure 3: Classification Throughput and Resource Utilization Across DBMS



[Note: Figure shows dual-axis combination chart comparing throughput and resource utilization across five database systems with clustered bars for Rule-Based, NLP, and Hybrid approaches, overlaid with line plots showing CPU Utilization, Memory Usage, and Disk I/O.]

4.3.2. Performance Across Unstructured Data Environments

Unstructured data classification exhibits different characteristics reflecting text processing complexity. For documents averaging 500+ words, throughput ranges from 89 to 312 documents/second depending on

size and method. Shorter communications under 300 words can achieve higher rates up to 412 documents/second with rule-based processing. Small documents averaging 500 words process at 312 documents/second (rule-based) and 89 documents/second (NLP). Large documents exceeding 5,000 words reduce throughput to 47 documents/second (rule-based) and 12 documents/second (NLP). Text complexity impacts efficiency beyond document length. Technical documentation processes 15-20% slower than general communications. Legal documents exhibit 25% throughput reduction reflecting complex sentence structures. Multilingual content requires language detection reducing throughput 8-12%.

File format processing introduces additional overhead. PDF text extraction adds 40-60ms per document. Optical character recognition for scanned documents consumes 800-1,200ms per page achieving 89-94% extraction accuracy. Microsoft Office formats parse with 15-25ms overhead. Plain text processes with minimal overhead establishing throughput baselines.

Table 5: Classification Performance by Document Characteristics

Document Type	Avg Words	Rule (docs/s)	NLP (docs/s)	Rule Time (ms)	NLP Time (ms)	Extract (ms)
Short Email	180	412	147	2.4	6.8	0.3
Business Doc	850	156	52	6.4	19.2	1.2
Technical	1,240	118	38	8.5	26.3	1.8
Legal	3,450	67	21	14.9	47.6	3.4
Reports	5,800	47	12	21.3	83.3	5.8
Scanned PDF	2,100	3.2	2.8	312.5	357.1	840.0
Native PDF	1,900	89	31	11.2	32.3	48.0
Office Formats	1,650	124	43	8.1	23.3	18.0
Plain Text	920	187	64	5.3	15.6	0.1

5. Discussion and Conclusion

5.1. Key Findings and Implications

5.1.1. Comparative Advantages of Different Approaches

Natural language processing methods demonstrate clear superiority for entity types embedded in natural language content. Person names, addresses, and medical information benefit from contextual understanding with NLP achieving 15-20% F1-score improvements over rule-based baselines. Transformer models effectively leverage surrounding context to resolve ambiguities and handle format variations confounding pattern matching. Rule-based detection maintains advantages in computational efficiency and deterministic behavior with 3-5× throughput enabling real-time monitoring using standard CPU infrastructure. Structured identifiers with standardized formats achieve comparable accuracy across both approaches, suggesting rule-based methods remain appropriate when computational efficiency matters.

Hybrid architectures effectively combine complementary strengths with 5-8% accuracy improvements over single-method approaches. Organizations must weigh these gains against increased implementation complexity. High-value applications protecting particularly sensitive data or operating under stringent regulatory requirements benefit from hybrid approaches maximizing detection coverage. Resource-constrained deployments may reasonably select simpler implementations. Adjustable detection thresholds and ensemble weights enable customization matching organizational risk tolerances and operational constraints.

5.1.2. Implementation Complexity and System Performance Trade-offs

Deployment considerations extend beyond algorithmic performance to practical implementation requirements. Rule-based systems exhibit lower deployment barriers

with straightforward configuration through pattern libraries. Organizations with existing security expertise can rapidly implement basic detection without specialized machine learning capabilities. Ongoing maintenance requires continuous rule updates addressing new formats and evasion techniques. NLP approaches demand greater upfront investment in infrastructure, training data, and specialized expertise. GPU requirements increase capital and operational costs. Model training necessitates labeled datasets and machine learning engineering capabilities potentially absent in organizations without established data science functions.

Scalability characteristics influence deployment architectures. Horizontal scaling through parallel processing proves straightforward for both approaches, enabling throughput expansion through additional compute resources. Cloud deployment shifts capital to operational expenses while providing elastic scaling matching workload variations. On-premises deployments offer greater control and potentially lower long-term costs but require upfront infrastructure investments and capacity planning.

5.2. Limitations and Challenges

5.2.1. Methodological Constraints and Generalizability Considerations

Several limitations affect findings scope and generalizability. Evaluation datasets reflect simulated enterprise environments rather than actual production data due to privacy constraints. Synthetic data generation attempts to replicate realistic characteristics but may not capture full complexity present in operational systems. Controlled experimental environments eliminate confounding variables enabling rigorous comparison but potentially understate real-world deployment challenges. Dataset composition reflects English-language content limiting findings to English contexts. Multilingual environments present additional challenges as language-specific models require separate development. Transfer learning from English models to other languages demonstrates degraded performance requiring language-specific fine-tuning. The evaluation encompasses contemporary detection approaches but cannot predict future algorithmic developments. Rapid advancement in language modeling may yield substantially different performance characteristics within short timeframes.

5.3. Future Research Directions

5.3.1. Privacy-Preserving Detection Mechanisms

Balancing effective detection against privacy protection of the detection process itself represents an important research challenge. Current approaches require security systems to access and analyze sensitive content, creating potential privacy risks if detection infrastructure is compromised. Differential privacy provides mathematical frameworks for quantifying information leakage enabling privacy-preserving analysis with formal guarantees. Federated learning enables collaborative model training across organizations without sharing raw data. Homomorphic encryption enables computation on encrypted data without decryption but imposes substantial computational overhead limiting practical deployment.

5.3.2. Cross-Environment Adaptability and Transfer Learning

Detection models trained in one organizational context often exhibit degraded performance when deployed in different environments due to distribution shifts. Transfer learning adapting general-purpose models to specific domains shows promise but requires tuning and domain-specific training data. Few-shot learning enabling model adaptation from limited examples could substantially reduce deployment barriers. Meta-learning approaches training models to rapidly adapt to new domains represent promising directions.

5.3.3. Real-Time Detection Optimization and Scalability Enhancement

Computational efficiency remains critical for real-time monitoring at enterprise scale. Model compression including quantization, pruning, and knowledge distillation reduces model size and inference latency while maintaining accuracy. These optimizations enable deployment on resource-constrained edge devices and reduce infrastructure costs. Hardware acceleration through specialized AI processors offers additional performance improvements. Adaptive processing allocating computational resources based on content characteristics could optimize system efficiency.

References

[1]. Kužina, V., Petric, A. M., Barišić, M., & Jović, A. (2023). CASSED: context-based approach for structured sensitive data detection. Expert systems with applications, 223, 119924.

[2]. Muralitharan, J., & Arumugam, C. (2024). Privacy BERT-LSTM: a novel NLP algorithm for sensitive information detection in textual documents. Neural Computing and Applications, 36(25), 15439-15454.

[3]. Jaikumar, J., & Suresh, P. (2023, July). Privacy-Preserving Personal Identifiable Information (PII) Label Detection Using Machine Learning. In 2023 14th International Conference on Computing

Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[4]. Kohli, R., Chatterjee, S., Gupta, S., & Gaur, M. S. (2023, December). Tracking PII ex-filtration: Exploring decision tree and neural network with explainable AI. In 2023 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (pp. 183-188). IEEE.

[5]. Sylligardos, E., Boniol, P., Paparrizos, J., Trahanias, P., & Palpanas, T. (2023). Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. Proceedings of the VLDB Endowment, 16(11), 3418-3432.

[6]. Sylligardos, E., Boniol, P., Paparrizos, J., Trahanias, P., & Palpanas, T. (2023). Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. Proceedings of the VLDB Endowment, 16(11), 3418-3432.

[7]. Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., & Zanella-Béguelin, S. (2023, May). Analyzing leakage of personally identifiable information in language models. In 2023 IEEE Symposium on Security and Privacy (SP) (pp. 346-363). IEEE.

[8]. Fredrikson, M., Jha, S., & Ristenpart, T. (2015, October). Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security (pp. 1322-1333).

[9]. Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., ... & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. IEEE transactions on information forensics and security, 15, 3454-3469.

[10]. Fu, J., Hong, Y., Ling, X., Wang, L., Ran, X., Sun, Z., ... & Cao, Y. (2024). Differentially private federated learning: A systematic review. arXiv preprint arXiv:2405.08299.

[11]. Li, G., Sun, J., Xu, L., Li, S., Wang, J., & Nie, W. (2024, May). Gaussml: An end-to-end in-database machine learning system. In 2024 IEEE 40th International Conference on Data Engineering (ICDE) (pp. 5198-5210). IEEE.