Modern strategies for data leak detection and prevention in corporate networks

Anatoliy Sachenko[1,2], Petro Vizhevskyi[3,*], Oleg Savenko[3], Viktor Ostroverkhov[2], Bogdan Maslyyak[2]

[1] *Casimir Pulaski Radom University, 26-600 Radom, Poland*
[2] *West Ukrainian National Unversity, Ternopil, 46009, Ukraine*
[3] *Khmelnytskyi National University, Khmelnytskyi, 29016, Ukraine*

Abstract

As companies handle ever-growing stores of sensitive information, from proprietary research to customer data, the threat of unauthorized disclosure escalates. Traditional Data Loss Prevention (DLP) measures, relying on static content matching and signature-based detection, have proven inadequate in detecting transformed or obfuscated sensitive information, particularly in environments that embrace remote work, Bring Your Own Device (BYOD) policies, and third-party integrations. This paper surveys the limitations of such conventional DLP systems and examines novel detection methodologies, including graph-based semantic analysis, probabilistic bigraph models, and context-aware anomaly detection, each addressing distinct facets of modern data leakage scenarios. Furthermore, the paper reviews prevention strategies that involve multi-layered defenses, robust encryption, secure file systems, and dynamic deception techniques to broaden the scope of adversarial deterrence.

A primary contribution of this study is a genetic-algorithm-driven method for detecting data leaks. Experiments on real data-leak datasets show that the method matches or surpasses the performance of standard baselines, including Naive Bayes and SVM, while maintaining low computational overhead. Future research should explore a dynamic ensemble in which the genetic algorithm assigns weights to multiple detection modules, thereby reducing false positives and keeping pace with evolving threat landscapes and corporate data practices. The paper concludes by underscoring the necessity of a multi-layered, continuously evolving DLP architecture, arguing that only through integrated and adaptive solutions can enterprises effectively safeguard their critical assets in an increasingly interconnected digital landscape.

Keywords

Data Loss Prevention, Anomaly Detection, Secure File Systems, Cloud Security, Dynamic Deception, Genetic Algorithms

1. Introduction

In today's interconnected digital landscape, protecting sensitive corporate data has become increasingly critical. Organizations now manage enormous volumes of both structured and unstructured data, ranging from emails and internal reports to intellectual property and customer records. This surge in data generation has not only heightened operational efficiencies but has also expanded the potential avenues for unauthorized data disclosure. Data leakage, whether through inadvertent mistakes by insiders or deliberate malicious actions, poses severe risks, including significant financial losses, reputational damage, and non-compliance with stringent regulatory frameworks.

[1] *MoDaST 2025: Modern Data Science Technologies Doctoral Consortium, June, 15, 2025, Lviv, Ukraine*
[1,*] Corresponding author.
[†] These authors contributed equally.

as@wunu.edu.ua (A. Sachenko); vizhevskyipv@khmnu.edu.ua (P.Vizhevskyi); savenko_oleg_st@ukr.net (O. Savenko); v.ostroverkhov@wunu.edu.ua (V. Ostroverkhov); b.maslyjak@wunu.edu.ua (B. Maslyyak);
0000-0002-0907-3682 (A. Sachenko); 0009-0009-4851-0839 (P. Vizhevskyi); 0000-0002-4104-745X (O. Savenko); 0000-0002-3818-0604 (V. Ostroverkhov); 0000-0002-9671-7617 (B. Maslyyak)

Recent research highlights that traditional security mechanisms, which are predominantly designed to defend against external cyber threats, are often insufficient when it comes to monitoring internal data flows. Many conventional Data Loss Prevention (DLP) systems rely on static content matching or predetermined patterns, which can falter when sensitive data undergoes transformations such as editing, reformatting, or partial redaction. For example, one study demonstrated that by representing documents as weighted graphs, it is possible to capture contextual sensitivity and detect modified data that would otherwise bypass standard detection methods [1].

In parallel, probabilistic models using bigraph representations have been introduced to statistically assess how sensitive data is distributed among various entities within an organization. These models underscore the importance of statistical analysis in tracking subtle changes in data flow that traditional DLP techniques might miss [2]. A comprehensive review of existing DLP methodologies further points out that approaches such as watermarking and content fingerprinting, while useful in certain scenarios, often struggle with the complexity introduced by insider threats and the dynamic nature of modern data formats [3].

Moreover, as organizations embrace modern workplace practices like BYOD (Bring Your Own Device) and remote work, the security perimeter becomes increasingly porous. Advanced DLP architectures are now required to monitor a heterogeneous mix of devices and endpoints without disrupting everyday operations [4]. Complementing these technical challenges, studies employing anomaly detection in relational databases have illustrated that monitoring the behavioral patterns of applications can offer an effective second layer of defense, further reinforcing the need for integrated, multi-dimensional approaches to data leak prevention [5].

Another emerging perspective is the concept of contextual integrity, which shifts the focus from merely detecting static content to evaluating the appropriateness of information flows between entities. This approach considers the relationships between senders, recipients, and the underlying data attributes, offering a more nuanced method to differentiate between legitimate and suspicious data exchanges [6]. In environments where language complexity and document transformations present additional hurdles, techniques based on morphological analysis have also been explored, particularly for languages with intricate grammatical structures [7].

Beyond software-centric solutions, research into physical and network-level vulnerabilities, including electromagnetic leakage from hardware, emphasizes that comprehensive data protection requires both digital and physical security measures [8]. Meanwhile, broader vulnerability assessments and comparative analyses of DLP systems reveal that an effective data protection strategy must combine technical innovations, such as big data analytics and machine learning [9], with cost-effective, low-intrusive solutions tailored to the operational realities of modern enterprises [10].

Recent advancements have also seen the integration of dynamic deception techniques, where the system deliberately alters or obfuscates data to expand the perceived attack surface, thus increasing the difficulty for attackers to extract genuine information. Such strategies complement conventional DLP mechanisms and provide an additional layer of resilience against both external and insider threats [11].

Finally, secure file system architectures designed specifically to address insider threats have been proposed, aiming to offer transparent protection without impeding user productivity. These systems leverage virtual file system techniques and are evaluated based on their ability to encrypt and monitor data flows without introducing significant overhead [12]. Additionally, tagging mechanisms that transform unstructured data into managed content repositories have emerged as a promising method to control information dissemination within an organization [13].

This paper aims to comprehensively survey the strengths and weaknesses of existing data leak detection and prevention strategies and then propose a novel method for data leak detection based on genetic algorithm that can be integrated into adaptive framework that unifies the most promising techniques within an ensemble approach guided by genetic algorithms. By dynamically weighting and combining modules, including morphological analysis, context-aware anomaly detection, time stamp-based classification, and moving target defenses, organizations can more effectively handle the complex mix of modern threats without continuous manual tuning.

2. Understanding Data Leaks

Data leakage involves the unintended or unauthorized dissemination of confidential or sensitive information outside the boundaries of an organization. This phenomenon may result from both inadvertent mistakes by employees and deliberate actions by insiders or external adversaries. The concept encompasses a range of incidents, including simple errors like accidental file sharing and sophisticated cyberattacks that exploit system vulnerabilities [1].

Sensitive information within organizations can be categorized into different states, each presenting unique risks. **Data at Rest** is information stored on servers, databases, or external storage devices. Breaches in this category often occur when unauthorized individuals gain physical or remote access to these storage systems [3]. **Data in Motion**, which is actively transmitted across networks via emails, file transfers, or cloud synchronization, is vulnerable to interception. Effective protection in this state typically relies on secure transmission protocols and encryption [9]. **Data in Use** is actively processed or accessed by applications or users. Leakage at this stage is frequently associated with insider threats or the exploitation of application-level vulnerabilities, which may not be adequately addressed by traditional perimeter-based security measures [5].

Various factors contribute to data leakage. Individuals within an organization, whether through negligence or malice, represent a significant threat. Studies show that a considerable percentage of breaches can be traced back to insiders who inadvertently expose sensitive information [14]. Many traditional DLP systems focus on static data patterns and keyword matching. These methods may fail when data is modified, for example, by reformatting or partial redaction, before it is exfiltrated. Advanced detection techniques are needed to accommodate these transformations [1]. The increasing use of cloud services, mobile devices, and remote work arrangements expands the potential leakage points. This diversity creates challenges in monitoring data consistently across various platforms and endpoints [4].

The consequences of data leakage can lead to substantial direct costs, including regulatory fines, litigation expenses, and remediation costs, along with indirect losses resulting from operational disruptions and reduced market confidence [15]. Exposure of sensitive information can severely damage an organization's reputation, leading to a loss of customer trust and competitive edge. The detection and mitigation process can strain organizational resources, particularly when security systems generate excessive false positives that interfere with normal business operations [9].

3. Threat Landscape and Attack Vectors

Modern corporate networks face a dynamic and diverse threat landscape in which both internal and external actors exploit various vulnerabilities to cause data leakage. Insider threats remain one of the most challenging aspects of data security. These threats emerge when employees or trusted individuals, either through carelessness or malicious intent, expose or intentionally leak confidential data. Several studies emphasize that insiders are often responsible for a significant portion of data breaches due to their extensive access rights and familiarity with internal systems. For instance, research on data leakage detection has shown that traditional methods often struggle to accurately monitor insider behavior, especially when the leaked data is intentionally altered to evade standard controls [3]. Additionally, survey results on machine learning-based DLP approaches indicate that insider actions, whether accidental or deliberate, require more adaptive detection techniques to effectively distinguish between normal operational patterns and suspicious behavior [14].

External adversaries continuously evolve their tactics to breach corporate defenses. Attackers exploit vulnerabilities in network protocols, unpatched software, and misconfigured systems to gain unauthorized access [5]. In particular, ransomware and phishing campaigns have emerged as prevalent forms of external attacks. Advanced systems that monitor application behavior and data flow anomalies have been shown to detect such threats with higher accuracy, yet the rapid evolution of malware strains often presents new challenges that traditional signature-based methods cannot address [11].

Another emerging vector in the data leakage external threat landscape is the exploitation of botnet networks. Botnets, which consist of numerous compromised devices coordinated through a centralized command-and-control infrastructure, are increasingly being used not only for distributed denial-of-service attacks but also for exfiltrating sensitive data. Botnets are organized in multiple tiers, with a command-and-control center directing intermediate control nodes and basic bot elements. This hierarchical structure enables attackers to remotely control a vast number of endpoints, aggregating small amounts of leaked data in a stealthy, distributed manner that can evade traditional data leakage prevention systems [16, 17]. The dynamic and decentralized nature of botnets makes it especially challenging for conventional security measures, which are typically designed to detect static or predictable data flows, to identify and mitigate such threats. As botnets continue to evolve, integrating specialized detection mechanisms that focus on identifying botnet behavior and its associated data exfiltration patterns becomes critical for robust corporate data security [18, 19].

In today's interconnected IT environment, organizations increasingly rely on third-party services, cloud platforms, and external vendors. This reliance creates additional vectors for data leakage, as vulnerabilities in supply chains or partner networks can serve as conduits for sensitive information to be exfiltrated. Research into BYOD policies and cloud-based DLP systems highlights that gaps in third-party security controls can lead to unmonitored data flows, making it imperative for enterprises to incorporate comprehensive risk assessments and stringent access controls across all external interfaces [4]. Furthermore, cost-effective strategies for cloud data protection are critical, particularly for small and medium-sized enterprises, as they face unique challenges in balancing security needs with limited resources [20].

A significant challenge in detecting data leaks arises from attackers deliberately transforming or obfuscating data to evade traditional DLP systems. Techniques such as content modification, insertion of benign text, or even partial redaction are used to mask sensitive information. Emerging detection models, including adaptive graph-based methods and contextual integrity frameworks, address these challenges by focusing on the underlying semantics and relationships within the data rather than relying solely on fixed patterns [21]. Such methods are especially effective in environments where data undergoes frequent transformations during routine operations, ensuring that even altered data is subject to robust monitoring.

Beyond purely digital threats, physical and side-channel attacks also contribute to the data leakage landscape. These attacks exploit non-traditional vectors such as electromagnetic emissions or hardware vulnerabilities to capture information without directly breaching network security. Investigations into the security of computer systems have demonstrated that electromagnetic leakage from displays and peripheral devices can inadvertently expose sensitive information, underscoring the importance of considering physical security measures alongside digital defenses [22].

The diversity of attack vectors, including insider mishaps, external cyberattacks, supply chain breaches, and physical side-channel exploits, underscores the complexity of the modern data leakage threat landscape. A successful defense strategy requires a multi-layered approach that integrates behavioral analysis, adaptive detection techniques, and comprehensive monitoring of both digital and physical environments. By understanding the interplay of these factors, organizations can design DLP solutions that are both resilient and responsive to the evolving nature of cyber threats [23].

4. Survey on Data Leak Detection

Detecting unauthorized disclosure of sensitive information in corporate environments requires a multifaceted approach. Modern detection techniques have evolved to address not only static data content but also transformed and obfuscated data, user behavior anomalies, and contextual irregularities.

One line of research involves representing documents as weighted graphs to capture both the significance of key terms and their contextual relationships. In these approaches, documents are converted into graphs where nodes represent sensitive keywords and edges capture their contextual dependencies. By applying an adaptive weighted graph walk model, systems can effectively identify cases where data has been altered, for example through partial modifications or inserted noise, to evade traditional detection methods [1]. In parallel, probabilistic models that leverage bigraph representations have been developed to statistically assess the likelihood of data leakage events by mapping the distribution of sensitive data among entities [2]. Both techniques focus on overcoming the limitations of fixed-pattern matching by integrating contextual and statistical analysis into the detection process.

Another detection strategy centers on identifying deviations from established behavioral norms. Systems employing anomaly detection techniques monitor sequences of operations, including database queries or file access patterns, and compare them against profiles of normal application behavior. For example, a detection system based on Hidden Markov Models (HMM) creates profiles from normal program traces, and deviations from these profiles may indicate data leakage attempts via application misuse [5]. This approach is especially useful for detecting subtle insider threats where an authorized user may perform atypical actions that could result in data leakage.

Detection techniques grounded in the concept of contextual integrity focus on evaluating whether information flows adhere to the expected norms within a given environment. Instead of simply scanning for sensitive keywords, these methods extract semantic flows by employing advanced natural language processing to verify that data exchange patterns comply with organizational policies and privacy regulations. By comparing observed communication sequences against a set of declaratively defined privacy rules, these systems can flag potentially non-compliant data transfers that may signal a leakage [6].

Advanced machine learning techniques have been employed to enhance detection accuracy, particularly when data is unstructured or when it undergoes transformation. Methods based on morphological analysis decompose text into its constituent parts (e.g., roots, stems, suffixes) to better capture the semantic content even when superficial changes are made. Combined with classification algorithms, these techniques help differentiate between benign modifications and genuine leakage of sensitive information [7]. Furthermore, surveys of machine learning approaches in DLP indicate that integrating both supervised and unsupervised learning models can significantly improve detection precision while reducing false positives [24].

Some detection systems incorporate temporal information as an additional layer of analysis. For instance, time stamp-based methods involve clustering documents and assigning temporal labels during a learning phase. During detection, if the document's time stamp falls within a critical period (e.g., before a scheduled public release), the system assigns a higher risk score, potentially flagging it as confidential [15]. In a complementary approach, content tagging methods organize data into controlled repositories. By tagging data with predefined labels, organizations can more easily monitor and restrict the flow of sensitive information across internal networks, thereby reducing the risk of inadvertent leakage [13].

Methods that integrate data transformation with moving target defense strategies dynamically alter the appearance of data, making it more difficult for adversaries to identify and exfiltrate genuine information. In these systems, deceptive data is generated based on both historical user behavior and current operational context, thereby increasing the attack cost for adversaries while preserving data usability for legitimate purposes [11].

Static content matching and fixed-pattern detection, often falter when sensitive information is disguised through reformatting, morphological changes, or partial redactions. Even more adaptive models that leverage anomaly detection or context-aware analysis still struggle to handle the heterogeneous mix of data flows brought by modern workforce practices and diverse endpoint devices. In large-scale corporate environments, high false-positive rates can overwhelm security teams, while purely signature-based systems prove ill-equipped against novel threats or insider misuse. Table 1 is summarizing advantages and shortcomings of selected existing detection methods.

Table 1
Comparison of data leak detection techniques

Method	Strengths	Weaknesses	Type of Data Handled
Adaptive Graph Walk [1]	Detects leaks after heavy text modification, efficient on long documents	Graph-build overhead, text only	Unstructured text
Probabilistic Bigraph [2]	Identifies likely leaker without watermarking, simple to audit	Accuracy drops with broad sharing, no collusion detection	Structured files or DB rows
AD-PROM HMM Anomaly Detector [5]	Very low false positives, light runtime impact	Needs wide training coverage, mimicry may evade	Application and DB behaviour
Contextual Integrity [6]	Flags semantic policy breaches, supports rich GDPR-style norms	Rule maintenance effort, NLP errors raise false alerts	Email and text messages
Timestamp-Based Sensitivity Scoring [15]	Protection expires automatically when data is no longer sensitive, accurate on fully confidential files	Misses partial snippets, requires correct expiry	Time-sensitive documents
Content-Tag Repository Control [13]	Central hub simplifies auditing and uniform policy enforcement, works across multiple channels routed through the repository	Users can bypass the repository, mis-tagging undermines protection	Any file stored or sent through the CMS
SVM Text Classifier [14]	High precision with moderate training data, fast inference once trained	Requires labelled corpus, vulnerable to newly obfuscated terms	Emails, documents, chats

Deep Autoencoder Anomaly Detection [14]	Detects previously unseen leak patterns, works without labelled data	Computationally intensive, benign anomalies may trigger false flags	Network traffic, system logs, mixed telemetry

5. Data Leak Prevention Strategies

Preventing data leakage requires a proactive, multi-layered approach that combines robust policies, technical safeguards, and adaptive monitoring. A strong foundation for data protection begins with comprehensive policies that define what constitutes sensitive data and set clear rules for its handling. Organizations should implement governance frameworks that enforce regulatory compliance, including adherence to GDPR and other data protection laws, and ensure that employees are well-trained in data security practices. These frameworks are essential for establishing accountability and promoting a security-aware culture throughout the enterprise [16].

To mitigate consequences of possible data leak deploying strong encryption for data at rest, in transit, and in use is critical. Advanced encryption techniques, along with rigorous access control policies, restrict unauthorized users from accessing or extracting sensitive information. Several studies highlight the importance of integrating these measures into corporate IT environments to both secure data and provide traceability in case of a breach [3]. Developing secure file systems that incorporate on-the-fly encryption and controlled access can significantly mitigate internal leakage risks. By creating virtual file systems that mirror actual file operations and enforce encryption/decryption during read and write operations, organizations can transparently protect data without hampering user productivity [12]. Some approaches classify data based on critical time windows by assigning temporal labels during a learning phase and enforcing access restrictions when documents fall within these sensitive periods. This strategy helps ensure that information remains confidential until it is meant to be released [15].

DLP solution can be grouped by deployment scheme as endpoint, network-wide or mixed [8]. Network deployed tools continuously monitor data flows across the organization's network, identifying and blocking unauthorized transmission of sensitive information. They can inspect content in real time and enforce policies that prevent leakage over unsecured channels [9]. At the device level, endpoint solutions that monitor user activity are critical for detecting anomalous actions that may signal insider threats. By comparing current user behavior against established baselines, these systems help detect and prevent data leakage before it occurs [14]. Mixed ones combine some of all attributes of both endpoint and network DLP tools.

Emerging prevention techniques leverage context and deception to add a proactive layer of defenses. Rather than relying solely on static rules, context-aware strategies assess whether information flows adhere to predefined privacy norms. By analyzing the roles of data senders, recipients, and the nature of the data exchanged, these systems can dynamically enforce policies that reflect real-world expectations, reducing the risk of unintentional leakage [6]. To further complicate efforts by adversaries, some systems dynamically generate deceptive data. This method alters the appearance of sensitive data to create a larger, misleading attack surface. Such techniques not only increase the difficulty for attackers to isolate genuine information but also trigger alarms when deceptive elements are manipulated, providing early warnings of potential breaches [11].

The widespread adoption of cloud services and BYOD policies requires specialized strategies. As data migrates to cloud environments, integrated DLP systems that monitor both cloud storage and transmission channels become vital. Hybrid strategies that combine on-premise controls with cloud-based monitoring enable organizations to maintain visibility over data regardless of its location, while ensuring compliance with evolving regulatory demands [16]. In environments where employees use personal devices for work, DLP strategies must extend to managing these endpoints. Tailored solutions include enforcing secure access policies, monitoring data flows on mobile devices, and segregating personal from corporate data to reduce the risk of accidental leaks [4].

Another approach involves the use of content tagging and the formation of controlled content repositories. By assigning metadata or labels to sensitive data as soon as it is created or modified, organizations can track the movement of critical information across systems. This tagging allows for the automated application of security policies and facilitates quick identification of data that should not leave a secure repository [13]. Restricting data movement to specific, monitored portals or repositories minimizes the exposure of sensitive information. These systems act as gatekeepers, ensuring that data is only transferred through secure channels and only to authorized destinations.

6. Data Leak Detection Using Genetic Algorithm

DLP systems typically classify data in two ways: by formal attributes (metadata such as "confidential" labels, document type, author, etc.) and by analyzing the actual content (file text, presence of specific patterns, keywords). The best results are achieved by combining both approaches, so the proposed method considers both file metadata and content to determine the information's sensitivity level.

We propose a classification method built on a genetic algorithm [24, 25]. During the tuning (training) phase, the system receives as input a set of data examples $D=\{d_1,\ldots,d_n\}$ labeled as confidential or non-confidential, $y_i \in \{0,1\}$. Each document in the input data array can be represented as a feature-presence vector:

$$x_j=(x_{j1},x_{j2},\ldots,x_{jm}), \tag{1}$$

where the element $x_{ji} \in \{0;1\}$ indicates the presence or absence of features taken from a predefined dictionary $T=\{t_1,\ldots,t_m\}$; each t_j may correspond to a keyword, a metadata field, or a match to a specific pattern. The GA module gradually evolves a set of rules or a model capable of classifying new data, and the resulting classifier is integrated into the distributed DLP system as local agent [26]. It inspects the content of files, messages, or other objects, together with their attributes, in order to determine whether they contain confidential information.

Each chromosome in the genetic algorithm encodes a candidate solution to the classification problem:

$$C=[r_1|r_2|\ldots|r_k], \tag{2}$$

where C denotes a set of k IF-THEN rules. Each rule is characterised by two subsequences: a positive template p_q that requires certain features to be present and a negative template n_q that requires certain features to be absent.

A chromosome can therefore be written as:

$$r_q=\begin{cases} 1, if\,(p_{q\cdot}x \geq 1) \wedge (n_{q\cdot}x=0) \\ \quad 0 \end{cases} \tag{3}$$

A document is classified as confidential if at least one rule is triggered:

$$\hat{y}(x)=max_{q=1,k}r_q. \tag{4}$$

The chromosome is therefore represented as a bit sequence with total length $L=2\cdot k\cdot m$ where k is the number of rules and m is the size of the feature dictionary.

During evolution each chromosome is evaluated on the training set. The evaluation measures how well the encoded rules identify confidential data (true detections) and how well they avoid confusing ordinary data with confidential data (minimising false alarms). A fitness function that reflects overall classification accuracy is used to score every candidate model. For this purpose, the counts of true positives *TP* , false positives *FP*, true negatives *TN*, and false negatives *FN* are calculated:

$$\begin{gathered} TP = \Sigma\left[\hat{y}(d) = 1 \wedge y = 1\right], \\ FP = \Sigma\left[\hat{y}(d) = 1 \wedge y = 0\right], \\ FN = \Sigma\left[\hat{y}(d) = 0 \wedge y = 1\right], \\ TN = \Sigma\left[\hat{y}(d) = 0 \wedge y = 0\right]. \end{gathered} \quad (5)$$

The fitness function to be maximised is defined as a combination of the Precision P and Recall R metrics:

$$\begin{gathered} P = \frac{TP}{TP + FP + \varepsilon}, \\ R = \frac{TP}{TP + FN + \varepsilon}, \\ F = \frac{2PR}{P + R + \varepsilon}, \\ Fit(C) = \alpha \cdot F - \beta \cdot \frac{L}{L_{max}}, \end{gathered} \quad (6)$$

where $\varepsilon \ll 1$ prevents division by zero, and $\alpha = 0.9$ and $\beta = 0.1$ are penalty coefficients that control the influence of rule length. At every iteration of the genetic algorithm the fittest individuals are selected for reproduction using tournament selection [27]. After selection, each pair of parents is combined by single-point crossover. Let $s \in \{1, \ldots, L-1\}$ be a randomly chosen cut position; the offspring is

$$child = (parent_1[1 \ldots s], parent_2[s+1 \ldots L]). \quad (7)$$

To maintain population diversity every bit in the chromosome is inverted with probability $p_{mut} = \frac{1}{L}$. The evolutionary process stops when either the predefined maximum number of generations is reached or the improvement over the last ten generations falls below tolerance $\delta \ll 1$:

$$\left|Fit_{best}^{g} - Fit_{best}^{g-10}\right| \leq \delta. \quad (8)$$

The proposed method derives its classification rules automatically from real data, whereas conventional DLP systems usually depend on hand-crafted templates and static policies. This data-driven process reduces reliance on domain experts and enables the system to adapt quickly when new formats or code words for sensitive information appear. The method produces an explicit set of IF–THEN rules that security specialists can read and verify, avoiding the opacity typical of black-box models such as many neural networks [28]. Because the logic is transparent, analysts can explain why a document was marked confidential, which fosters trust and simplifies forensic investigations; when requirements change, the rules can be edited directly instead of retraining an entire model.

Proposed method can be utilized in endpoint-based agent data leak detection through three main phases shown in Figure 1.

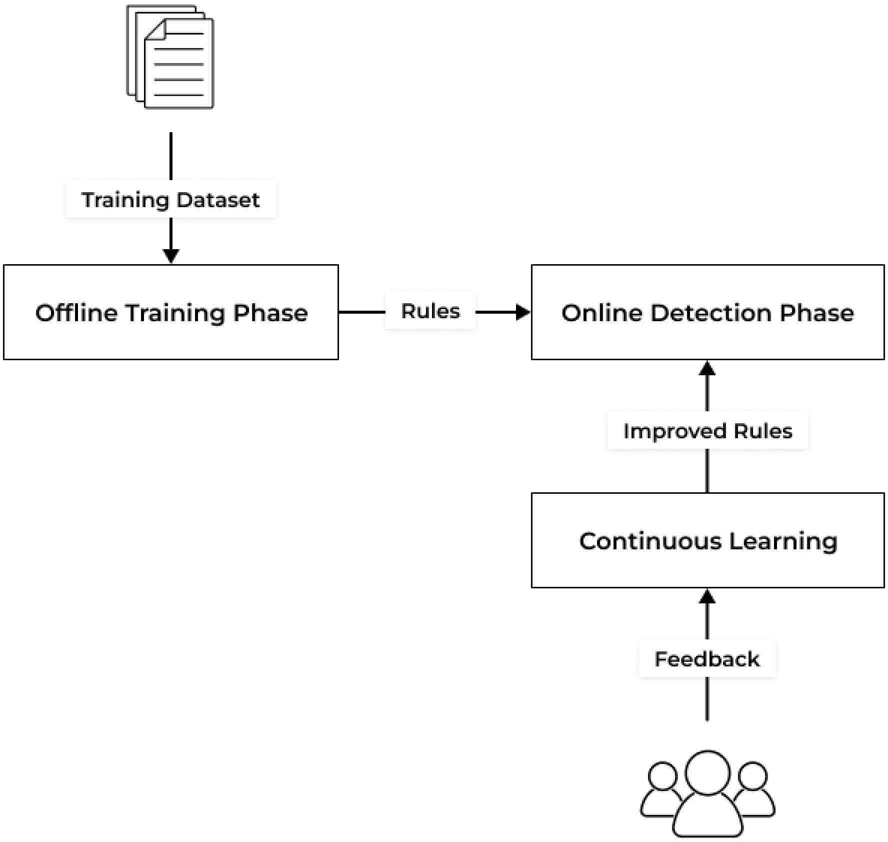

Figure 1: Data leak detection agent operation phases.

For Offline Training Phase given historical leak data (emails, documents, media files) as input following steps are executed:

1. Feature extraction (text patterns, metadata).
2. Rules sets population initialization.
3. Fitness evaluation.
4. Selection of new population based on fitness.
5. Crossover.
6. Mutation.

Steps 3-6 are repeated until equation (8) is satisfied. Result is evolved set of IF-THEN rules for classifying data as confidential or not that is used by Online Detection Phase.

For Online Detection Phase initial input is set of rules, execution phase consists of steps:

1. Data (document, email etc.) interaction identified.
2. Feature extraction for identified data similar to Offline Training Phase.
3. IF-THEN rules applied
4. Appropriate action performed (Allow, Block, Log etc.)

Continuous Learning phase generates new rules that reflect the most recent data landscape, so the classifier keeps pace with emerging document structures. By evaluating both textual features and metadata, the system gains a broader inspection context and lowers the likelihood of missing a leak.

7. Experiments

We validate our method across three real-world datasets listed in Table 2 and compare its performance against established machine learning algorithms. Each dataset underwent extensive preprocessing to ensure consistent feature extraction and fair comparison across methods.

Table 2
Dataset characteristics

Dataset	Documents	Estimated Positive Class	Domain
Enron Email Corpus [29]	500 000+	15-25%	Corporate Email
AI4Privacy PII-300k [30]	220 000+	40-50%	PII data
USA Government FOIA [31]	70 000+	10-40%	Goverment documents

The preprocessing pipeline involved dataset-specific stages to ensure high-quality feature extraction for both genetic algorithm and baseline methods. For the Enron Email Corpus, email body text and metadata were extracted from raw files with headers parsed to separate content from routing information. Signatures, quoted replies, and forwarding chains were removed, followed by text normalization including lowercasing, contraction expansion, and special character removal while preserving meaningful punctuation. A domain-specific stop word list retained security-relevant terms like "confidential" and "restricted." Email addresses were tokenized as EMAIL_TOKEN with internal/external domain preservation, while attachments were processed separately with filenames and extensions as additional features.

The AI4Privacy PII-300k dataset required custom regular expressions to detect and tokenize PII patterns including SSN (XXX-XX-XXXX), credit cards, phone numbers, and addresses. Each PII type was replaced with corresponding tokens (SSN_TOKEN, CREDITCARD_TOKEN) while preserving presence information. Special cases like partial redactions ("SSN: XXX-XX-1234") and age-revealing date formats were handled, with PII density features calculating the ratio of PII tokens to total tokens per document.

For the Government FOIA dataset, classification markings (e.g., "CONFIDENTIAL//NOFORN") were extracted as separate features before text removal. Redacted sections ([REDACTED] or 'X' blocks) were replaced with REDACTION_TOKEN while counting redaction frequency and length. Paragraph structure was preserved due to section-specific classification levels, with page numbers, form numbers, and reference codes becoming metadata features.

TF-IDF vectorization used dataset-specific parameters: Enron (1,000 features, 0.001-0.95 document frequency), AI4Privacy (500 features), and FOIA (750 features). Metadata-derived features included sender-recipient patterns and time indicators for emails, PII co-occurrence statistics for privacy data, and classification/redaction patterns for government documents.

Data splitting employed stratified sampling to maintain class distribution across 80-20 train-test splits with fixed random seeds for reproducibility. A validation set (10% of training data) was created for baseline hyperparameter tuning, while the genetic algorithm used the full training set with fixed parameters. For temporal datasets (Enron and FOIA), we verified that random splitting avoided temporal leakage and ensured documents from the same conversation threads or document families remained within the same split to prevent information leakage.

The genetic algorithm used a population of 100 individuals with maximum 200 generations and early stopping when fitness improvement over 20 consecutive generations fell below 0.001. The fitness function combined F_1-score with a rule length penalty ($\lambda=0.01$) to balance accuracy and interpretability. Tournament selection (size 3), single-point crossover (probability 0.7), and bit-flip mutation (probability 0.05 per gene) maintained diversity while preserving good solutions.

Five baseline methods were compared using standard parameters: Multinomial Naive Bayes with Laplace smoothing ($\alpha=1.0$), SVM with RBF kernel ($C=1.0, \gamma=\frac{1}{n_{features}}$), Decision Tree and Random Forest (100 estimators) both with maximum depth 10, and Logistic Regression with L2 regularization (C = 1.0). All methods used identical preprocessed features and train-test splits.

Evaluation used macro-averaged F_1-score as the primary metric to handle class imbalance, with precision and recall providing additional insight into false positive and false negative rates critical for security applications. For interpretable models, rule complexity was measured as the number of unique features in the final rule set. Training times were recorded on identical hardware (Apple M3 Max, 36 GB RAM) to assess computational requirements.

Table 3 presents the F_1-scores achieved by each method across all datasets, demonstrating the genetic algorithm's consistent high performance across diverse document types and classification challenges. The genetic algorithm achieved the highest average F_1-score (0.877) across all datasets, demonstrating robust performance in diverse document classification scenarios. While SVM slightly outperformed GA on the AI4Privacy dataset (0.921 vs 0.913), this dataset's relatively simple PII patterns favored SVM's ability to find optimal separating hyperplanes. GA showed superior performance on datasets with more complex decision boundaries, particularly the Enron corpus where subtle contextual patterns determine document sensitivity.

The genetic algorithm's evolved rules demonstrated clear domain-specific patterns that align with human understanding of document sensitivity. For the Enron Email dataset, the algorithm identified a core set of required features including "confidential" "internal", "restricted", "employee", and "salary" combined with forbidden features such as "public", "press", "announcement", and "release". This rule effectively captures the intuitive notion that documents discussing internal employee matters with confidentiality markers, but lacking public dissemination indicators, likely contain sensitive information.

On the AI4Privacy dataset, evolved rules centered on PII patterns, requiring presence of tokens like "ssn", "credit_card", "address", "phone", and "email" while forbidding synthetic data indicators such as "example", "test", "sample", and "demo". This demonstrates the algorithm's ability to distinguish real PII from training examples or documentation, a critical capability for practical deployment.

Government FOIA document rules revealed hierarchical classification patterns, with required features including official classification markings ("classified", "secret", "redacted", "official_use") and forbidden features representing public release indicators ("unclassified", "public_release", "approved"). The algorithm successfully learned the bureaucratic language patterns that distinguish classified from publicly releasable government documents.

The evolutionary process showed consistent convergence within 100 generations across all datasets. Enron converged at generation 87 (F_1-score 0.872), AI4Privacy reached convergence fastest at generation 62 (F_1-score 0.913) due to simpler pattern structure, and Government FOIA required 95 generations (F_1-score 0.847) reflecting diverse terminology across agencies. Early stopping prevented unnecessary computation and overfitting as fitness improvements plateaued upon discovering optimal feature combinations, with consistent convergence behavior suggesting robust algorithm design adapting naturally to different problem complexities.

Table 3

F_1-Score Comparison Across Datasets

Method	Enron Email	AI4Privacy PII	FOIA	Average
GA (Ours)	0,872	0,913	0,847	0,877
Naive Bayes	0,823	0,887	0,792	0,834
Decision Tree	0,798	0,854	0,773	0,808
Random Forest	0,841	0,903	0,818	0,854
Logistic Reg.	0,834	0,895	0,809	0,846
SVM	0,856	0,921	0,831	0,869

The genetic algorithm demonstrated notable robustness to class imbalance, a critical property for security applications where sensitive documents typically comprise a minority class. At the most balanced ratio of 1:1.3 in the AI4Privacy dataset, SVM slightly outperformed GA with an F_1-score of 0.921 versus 0.913. However, as imbalance increased to 1:3.2 and 1:4.5 in the Government FOIA and Enron datasets respectively, GA showed consistent advantages of approximately 0.016 in F_1-score over both SVM and Naive Bayes. This robustness stems from the fitness function's use of F_1-score, which inherently balances precision and recall, combined with the evolutionary search's ability to discover feature combinations that reliably identify minority class instances even when positive examples are scarce.

The interpretability comparison reveals fundamental differences between methods in terms of human comprehension and practical deployment. The GA approach produces rules using between 15 and 35 features across different datasets, compared to decision trees requiring 45 to 89 features to achieve lower accuracy. Black-box methods like SVM, Neural Networks, and Naive Bayes utilize the entire feature space of over 1,000 features, making interpretation practically impossible without additional explanation techniques. GA rules can be directly expressed in natural language that security analysts understand, such as "Document is sensitive if it contains 'confidential' and 'internal' but not 'public' or 'press release'." This interpretability enables security teams to validate rules against organizational policies, adjust them based on domain knowledge, and explain decisions to stakeholders or during audits.

The experimental results validate our hypothesis that evolutionary search can effectively explore the vast space of feature combinations to discover accurate yet interpretable classification rules. The genetic algorithm achieved the highest average F_1-score across diverse datasets while producing rules an order of magnitude simpler than decision trees. This demonstrates that the global search capability of evolutionary algorithms can identify compact feature sets that capture essential patterns for sensitive document detection. The method's consistent performance across datasets with varying characteristics indicates robust generalization. Unlike black-box methods that may learn dataset-specific quirks, the evolved IF-THEN rules capture fundamental patterns that transfer well across domains.

The interpretability of evolved rules extends beyond mere feature counting. The rules express logical relationships that align with human intuition about document sensitivity, combining positive indicators with negative evidence in a natural way. This bi-directional reasoning mirrors how human analysts approach document classification, checking both for presence of sensitive markers and absence of public dissemination indicators. The compact rule sets can be directly implemented in existing security infrastructure without specialized machine learning frameworks, using simple pattern matching engines. Security analysts can inspect and understand the rules, building trust in automated decisions and enabling manual overrides when organizational policies change.

The method's high precision reduces false positive rates that plague many automated security systems, preventing alert fatigue among security teams. Meanwhile, competitive recall ensures most sensitive documents are caught, with the interpretable rules helping analysts understand any misclassifications and refine detection patterns. The evolutionary approach also supports incremental improvement as new types of sensitive documents emerge. Rather than retraining from scratch, the existing rule population can seed a new evolutionary run, allowing rapid adaptation to evolving threats while preserving proven detection patterns.

While we evaluated on diverse real-world datasets, highly specialized document types may exhibit different characteristics. Technical documents with extensive code snippets or mathematical formulas might require adapted preprocessing. However, the genetic algorithm's flexibility to incorporate domain-specific features suggests it would adapt well to such scenarios. The selection of F_1-score as the primary metric appropriately balances the competing demands of precision and recall in security applications where both false positives and false negatives carry significant costs.

Our comprehensive experimental evaluation demonstrates that genetic algorithm-based document classification successfully achieves its dual objectives of high accuracy and interpretability. Across three diverse datasets representing different sensitive document detection scenarios, the evolutionary approach discovered compact IF-THEN rules that achieved superior average performance while using significantly fewer features than decision trees. The method showed particular strength on challenging real-world datasets like Enron emails, where complex contextual patterns determine sensitivity. Its robustness to class imbalance and ability to produce human-understandable rules make it especially suitable for security applications where both performance and explainability are critical. These results validate evolutionary search as an effective approach for exploring the combinatorial space of features in document classification, finding globally optimal solutions that balance multiple objectives. The success of this method opens possibilities for applying evolutionary techniques to other security tasks requiring interpretable models, such as intrusion detection, fraud identification, and regulatory compliance monitoring.

8. Future Directions

Existing data leak detection and prevention solutions exhibit several critical shortcomings discussed in section 4. Attempts to integrate encryption, tagging, or behavioral analysis sometimes lack holistic coordination, leading to visibility gaps that sophisticated adversaries readily exploit [8, 14]. Furthermore, although dynamic deception and multi-layered defense can reduce false negatives, their efficacy depends heavily on precise calibration for each organization’s operational context, which can be labor-intensive to maintain.

Way to overcome these limitations is to take the most promising features of existing detection techniques:

- **Comprehensive Policy Frameworks and Employee Training:** Organizations should establish clear data handling policies and conduct regular training sessions to ensure that all employees understand the importance of data security. This practice forms the backbone of any effective Data Loss Prevention (DLP) strategy by aligning technical measures with organizational culture [16].
- **Robust Encryption and Access Controls:** Implementing end-to-end encryption for data at rest, in transit, and in use is critical. Coupled with strict access control mechanisms, these

technical safeguards help restrict unauthorized access and ensure that sensitive data remains protected even if other layers of defense are breached [3].

- **Context-Aware Monitoring and Anomaly Detection:** Modern DLP systems benefit from integrating behavioral analytics and context-aware detection techniques. By continuously comparing user actions against established baselines, these systems can quickly identify deviations that may indicate insider threats or other anomalies. This layered approach not only reduces false positives but also provides a more nuanced understanding of data movement within the network [5].
- **Time Stamp and Content Tagging Strategies:** Employing methods that incorporate temporal metadata and content tagging can significantly enhance data classification. By marking data with time-sensitive attributes and specific labels, organizations can automate policy enforcement and restrict data exposure during critical periods. This is especially useful in environments where the confidentiality status of data changes over time [13, 15].
- **Layered Defense with Network and Endpoint Integration:** Best practices recommend deploying a blend of network-wide monitoring alongside endpoint-specific controls. This ensures that even if data is accessed or modified at a local level, broader network policies and real-time monitoring can detect and mitigate unauthorized data flows [32, 33]. Employing local and network-wide hardware based security modules can significantly speed up analysis and decrease operational costs [31].
- **Dynamic Deception and Moving Target Defense:** Approaches such as generating deceptive data or dynamically altering the data attack surface add a proactive dimension to DLP strategies. These methods complicate an attacker's efforts by increasing uncertainty and raising the cost of data exfiltration, thereby acting as an additional safeguard against both internal and external threats [11, 17].

And unite them as modules into single adaptive framework guided by a genetic algorithm. ach detection module offers partial "scores" or indications of potential leakage. These outputs then become the genetic algorithm's raw material. The system starts with multiple candidate configurations, each specifying how to weight or combine modules' outputs under different network conditions, data types, and time constraints. Based on feedback from both real-time and historical data leak events, including false positives and missed detections, an evolutionary process evaluates the fitness of each configuration, eliminating underperforming combinations and promoting or mutating more successful ones.

Over time, this iterative process hones in on configurations that maximize true positives and minimize false alarms, continually rebalancing priorities among modules. Such an ensemble not only becomes more robust against varied threats but also circumvents the need for constant manual tuning, a known pain point in large-scale DLP deployments [8].

9. Conclusion

Effective protection against data leakage demands a layered, adaptive strategy that integrates complementary detection and prevention techniques. Experiments on three real-world datasets confirm that the proposed genetic-algorithm classifier delivers the highest average F_1-score while producing concise, human-readable rules. Because these rules explicitly combine textual cues and metadata, analysts can readily verify decisions and refine policies without retraining opaque models. The method's robustness to class imbalance and its modest computational overhead make it practical for large-scale corporate environments in which sensitive documents form only a small fraction of overall traffic.

Beyond a single classifier future work should be aimed at developing an ensemble architecture in which the genetic algorithm continually adjusts the weight of diverse modules such as anomaly detection, contextual integrity checks, time-aware sensitivity scoring and dynamic deception. By treating each module's output as an input feature and evolving optimal weightings, the ensemble reduces false positives and adapts as new workflows, devices and cloud services emerge. This evolutionary coordination also lessens the manual effort that traditionally accompanies rule maintenance in complex distributed systems.

Future research should also focus on extending the framework to additional data states, including encrypted streams and multimedia content, and on tightening resistance to adversarial transformations. Investigating hardware-level telemetry for corroborating evidence and integrating privacy-preserving learning techniques will further enhance resilience. As corporate networks grow more heterogeneous and regulations more stringent, the genetic-algorithm-guided ensemble offers a promising foundation for DLP solutions that must remain accurate, transparent and agile.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] X. Huang, Y. Lu, D. Li and M. Ma, A Novel Mechanism for Fast Detection of Transformed Data Leakage, IEEE Access, vol. 6, 2018, pp. 35926-35936. doi:10.1109/ACCESS.2018.2851228.

[2] I. Gupta, A. Singh, A Probability based Model for Data Leakage Detection using Bigraph, in: Proceedings of the 2017 7th International Conference on Communication and Network Security (ICCNS '17). Association for Computing Machinery, New York, NY, USA, 2017, pp.1-5. doi:1-5. 10.1145/3163058.3163060.

[3] K. Gupta, A. Kush, A Review on Data Leakage Detection for Secure Communication, NTERNATIONAL JOURNAL ENGINEERING AND APPLIED TECHNOLOGY (IJEAT), Vol. 7, 2017, pp.153-159.

[4] S. E. Calias, B. Caoli, R. Padilla, J. Tum-en, K. C. Bacilio, I. Lyn, G.S. Guaki, The Impact of BYOD (Bring Your Own Device) On Network Security: A Literature Review, in: Southeast Asian Journal of Science and Technology, Vol. 9, 2024.

[5] D. Fadolalkarim, E. Bertino, and A. Sallam, An Anomaly Detection System for the Protection of Relational Database Systems against Data Leakage by Application Programs, Purdue University, in: Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE*)*, Dallas, TX, USA, 2020, pp. 265-276. doi:10.1109/ICDE48307.2020.00030.

[6] Y. Shvartzshnaider, Z. Pavlinovic, A. Balashankar, T. Wies, L. Subramanian, H. Nissenbaum, P. Mittal, VACCINE: Using Contextual Integrity For Data Leakage Detection, in: Proceedings of the The World Wide Web Conference (WWW '19). Association for Computing Machinery, New York, NY, USA, 2019. doi:10.1145/3308558.3313655.

[7] M. Hart, P. Manadhata, R. Johnson, Text Classification for Data Loss Prevention, 2011. doi:10.1007/978-3-642-22263-4_2.

[8] S. Syarova, S. Toleva, A. Kirkov, S. Petkov, K. Traykov , Data Leakage Prevention and Detection in Digital Configurations: A Survey, in: Proceedings of the 15th International Scientific and Practical Conference, Vol. 2, 2024. doi:10.17770/etr2024vol2.8045

[9] J.K. Periasamy, A. Cindy Catherine, R. Elamathi, S. Subhiksha, Data Leakage Vulnerability Assessment, in: Proceedings of the 2023 Intelligent Computing and Control for Engineering and Business Systems, 2023. doi:10.1109/ICCEBS58601.2023.10448949.

[10] L. Cheng, F. Liu, D. D. Yao, Enterprise data breach: causes, challenges, prevention, and future directions, in: WIREs Data Mining Knowl Discov, 2017. doi:10.1002/widm.1211.

[11] K. Chen, Q. Yang, J. Wang, D. Ma, L. Wang, and Z. Xu, What You See Is The Tip Of The Iceberg: A Novel Technique For Data Leakage Prevention, in: Proceedings of the 27th International Conference on Computer Supported Cooperative Work in Design, 2024. doi:0.1109/CSCWD61410.2024.10580487.

[12] I. Herrera Montano, I. de la Torre Díez, J. J. García Aranda, J. Ramos Diaz, S. Molina Cardín, and J. J. Guerrero López, Secure File Systems for the Development of a Data Leak Protection (DLP) Tool Against Internal Threats, in: Proceedings of the 17th Iberian Conference on Information Systems and Technologies, 2022. doi:10.23919/CISTI54924.2022.9820170.

[13] M. H. Matthee, Tagging Data to Prevent Data Leakage (Forming Content Repositories), 2016. doi:99.9999/woot07-S422.

[14] G. Agrawal, S. J. Goyal, Survey on Data Leakage Prevention through Machine Learning Algorithms, in: Proceedings of the 2022 International Mobile and Embedded Technology Conference, 2022. doi:10.1109/MECON53876.2022.9752047

[15] S. Peneti, B. P. Rani, Data Leakage Prevention System with Time Stamp, in: Proceedings of the International Conference on Information Communication and Embedded Systems, 2016. doi:10.1109/ICICES.2016.7518934

[16] O. Savenko, A. Sachenko, S. Lysenko, G. Markowsky, N. Vasylkiv, BOTNET DETECTION APPROACH BASED ON THE DISTRIBUTED SYSTEMS, in: International Journal of Computing, Vol. 19(2), 2020, pp. 190-198. doi:10.47839/ijc.19.2.1761.

[17] S. Lysenko, O. Savenko, K. Bobrovnikova, DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering, CEUR-WS, Vol.2104, 2018, pp. 688-695.

[18] S. Lysenko, O. Savenko, K. Bobrovnikova, A. Kryshchuk, B. Savenko. Information technology for botnets detection based on their behaviour in the corporate area network, Communications in Computer and Information Science, Vol. 718, 2017, pp. 166–181.

[19] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova. A Technique for the Botnet Detection Based on DNS-Traffic Analysis. Communications in Computer and Information Science. Vol. 522, 2015, pp.127-138.

[20] V. Valleru, COST-EFFECTIVE CLOUD DATA LOSS PREVENTION STRATEGIES FOR SMALL AND MEDIUM-SIZED ENTERPRISES, International Research Journal of Engineering and Technology, Vol. 11, Iss. 05, 2024.

[21] S. D. Gupta, R. Kumar, waRLOCK: Countering Ransomware and Data Leak, in: Proceedings of the 2024 IEEE International Conference on Contemporary Computing and Communications, 2024. doi:10.1109/INC460750.2024.10649292.

[22] A. Kashtalian, S. Lysenko, O. Savenko, A. Nicheporuk, T. Sochor, V. Avsiyevych, Multi-computer malware detection systems with metamorphic functionality, Radioelectronic and Computer Systems, 2024, pp. 152-175. doi:10.32620/reks.2024.1.13.

[23] M. Alojo, Innovative Approaches in Data Management and Cybersecurity: Insights from Recent Studies, World Journal of Advanced Research and Reviews, Vol. 23, Iss. 03, 2024, pp. 2410–2425. doi:10.30574/wjarr.2024.23.3.2897.

[24] M. M. R. Mazumder, C. Phillips, PARTITIONING KNOWN ENVIRONMENTS FOR MULTI-ROBOT TASK ALLOCATION USING GENETIC ALGORITHMS, International Journal of Computing, Vol. 19(3), 2020, pp. 480-490. doi:10.47839/ijc.19.3.1897

[25] P. Bykovyy, V. Kochan, A. Sachenko, G. Markowsky, Genetic Algorithm Implementation for Perimeter Security Systems CAD, in: Proceedings of the 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, 2007, pp. 634-638, doi: 10.1109/IDAACS.2007.4488498.

[26] S. Obadan, Z. Wang, A MULTI-AGENT APPROACH TO POMDPS USING OFF-POLICY REINFORCEMENT LEARNING AND GENETIC ALGORITHMS. International Journal of Computing, Vol. 19(3), 2020, pp.377-386. doi:10.47839/ijc.19.3.1887

[27] P. Bykovyy, Y. Pigovsky, V. Kochan, A. Sachenko, G. Markowsky, S. Aksoy, Genetic algorithm implementation for distributed security systems optimization, in: Proceedings of the 2008 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2008, pp. 120-124, doi:10.1109/CIMSA.2008.4595845.

[28] R. Lynnyk, V. Vysotska, Y. Matseliukh, Y. Burov, L. Demkiv, A. Zaverbnyj, A. Sachenko, I. Shylinska, I. Yevseyeva, O. Bihun, DDOS Attacks Analysis Based on Machine Learning in Challenges of Global Changes, in: CEUR Workshop Proceedings (CEUR-WS.org) MoMLeT+DS

2020 Modern Machine Learning Technologies and Data Science Workshop 2020, pp. 159-171. ISSN 1613-0073.

[29] The Enron Email Dataset. URL: https://www.kaggle.com/datasets/wcukierski/enron-email-dataset.

[30] AI4Privacy PII-300k. URL: https://huggingface.co/datasets/ai4privacy/pii-masking-300k.

[31] Government FOIA. URL: https://www.foia.gov/foia-dataset-download.html.

[32] A. Sachenko, V. Kochan, V. Turchenko, Instrumentation for gathering data [DAQ systems], IEEE Instrumentation & Measurement Magazine, Vol. 6, 2003, pp. 34-40. doi:10.1109/MIM.2003.1238339.

[33] V. Hamolia, V. Melnyk , P. Zhezhnych, A. Shilinh, INTRUSION DETECTION IN COMPUTER NETWORKS USING LATENT SPACE REPRESENTATION AND MACHINE LEARNING. International Journal of Computing, Vol. 19(3), 2020, pp. 442-448. doi:10.47839/ijc.19.3.1893.

[34] O. Kehret, A. Walz, A. Sikora, INTEGRATION OF HARDWARE SECURITY MODULES INTO A DEEPLY EMBEDDED TLS STACK, International Journal of Computing, Vol. 15(1), 2016, pp. 22-30. doi:10.47839/ijc.15.1.827