International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Machine Learning Based Data Leakage Detection and Prevention

Harshita Jain, Vishakha Belge, Vivek Kurve, Tanmay Gharde, Savi Sawwalakhe, and Vedant Matte
Department of Computer Technology
Kavikulguru Institute of Technology and Science, Ramtek, Nagpur, Maharashtra, India

Abstract **- Data leakage refers to the unauthorized disclosure or unintended transfer of sensitive information and represents a significant security concern for organizations. This paper presents SentinelDLP, a machine-learning-based data leakage detection and prevention system designed for Windows endpoints. The system combines rule-based techniques with TF-IDF and Logistic Regression for content classification into different sensitivity levels. It also includes an Isolation Forest component intended for behavioural anomaly detection.**

SentinelDLP provides monitoring and detection capabilities across multiple endpoint channels, including filesystem activity, USB devices, clipboard operations, screenshots, printing, MTP devices, and network shares. The system generates alerts and incident records to support administrative monitoring and investigation. It also incorporates risk-scoring and policy-management components.

Print prevention is implemented for detected sensitive printing activities. USB transfer blocking is activated through a separate command that must be executed with administrator privileges from the project directory. Once activated, the current mechanism blocks all file transfers to USB devices rather than selectively blocking files according to their sensitivity. The Windows Shell Copy Hook is present in the source code but has not been implemented successfully as a functioning selective file-copy prevention mechanism.

The content-classification model was evaluated using separate training and testing datasets. Further verification is required to establish the integration of behavioural anomaly detection, the reported evaluation metrics, and the effectiveness of prevention across all monitored channels. The project demonstrates an approach to combining content classification, endpoint monitoring, and selected prevention mechanisms for data leakage detection and prevention.

Keywords - **Data Leakage Detection, Data Loss Prevention, Machine Learning, TF-IDF, Logistic Regression, Isolation Forest, Endpoint Security, Risk Scoring.**

I. INTRODUCTION

The rapid growth of digital data usage has increased the importance of protecting sensitive information from unauthorized disclosure and transfer. This research focuses on the combined application of data security, machine learning, data loss prevention (DLP), and endpoint and behavioral monitoring**.** Organizations routinely exchange sensitive information through USB storage devices, cloud platforms, web-based email services, messaging applications, printers, and mobile devices. As a result, the number of possible channels through which information can be unintentionally or deliberately exposed continues to increase. Previous studies indicate that human error is a major contributor to data leakage incidents, often exceeding incidents caused by intentionally malicious insiders [1]. This highlights the need for security mechanisms that can actively prevent unauthorized data transfers rather than simply recording or reporting them.

Existing research has addressed data leakage from several perspectives, particularly forensic analysis and post-incident attribution. Some approaches use allocation-based techniques to estimate the likelihood that a particular recipient is responsible for a leaked dataset, while other methods incorporate fabricated objects, markers, or watermarks into data so that a leaked copy can subsequently be associated with its source [2], [5], [9], [10]. Although these approaches can assist in identifying the origin of a leakage incident, they generally operate after the data has already been exposed and therefore do not directly prevent the unauthorized transfer itself.

Machine learning has also been investigated for identifying potential leakage risks. For example, deep-learning techniques have been applied to source-code analysis and taint tracking to identify potentially unsafe data flows [3]. Other studies have examined cloud-storage access patterns and user behavior to identify abnormal activities associated with possible data leakage [4]. These approaches demonstrate the usefulness of machine learning for detecting suspicious behavior; however, there remains a need for practical endpoint-oriented systems capable of evaluating the actual content being transferred and taking an appropriate preventive action before the transfer is completed.

Commercial DLP platforms, including Microsoft Purview DLP, Symantec DLP, and Forcepoint DLP, provide integrated capabilities for monitoring, detecting, and preventing sensitive-data transfers. However, such solutions are generally proprietary and may require paid licensing, enterprise infrastructure, and specialized deployment environments. These requirements can make them difficult to adopt in

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

academic projects and resource-constrained environments.

To address this gap, this research proposes SentinelDLP, a machine-learning-based endpoint data leakage detection and prevention system designed for Windows environments. The proposed approach combines content-based classification, behavioral analysis, risk scoring, and policy-based decision making to evaluate potential data transfers. Instead of allowing a machine-learning model to independently make security decisions, the system uses machine-learning outputs as bounded inputs to an explainable risk-scoring mechanism and policy engine. This enables the system to provide transparent reasons for its decisions while supporting preventive actions on supported endpoint channels.

The primary research question addressed by this work is whether a machine-learning-based endpoint agent developed using freely available tools can accurately identify sensitive information and provide practical, explainable prevention of unauthorized data transfers without relying on proprietary enterprise infrastructure. The proposed system therefore aims to demonstrate an affordable and practical approach to endpoint data loss prevention while maintaining transparency in the security decisions generated by the system.

II. LITERATURE SURVEY

The literature review was conducted to identify recent research relevant to data leakage detection and prevention, examine foundational approaches, compare existing methodologies, and determine the research gap addressed by the proposed system. Ten representative studies were considered across major research directions, including machine-learning-based detection, allocation-based guilt assessment, fake-object techniques, deep-learning-based source-code analysis, cloud-based monitoring, insider-threat detection, and organizational security guidelines. This broader selection enables comparison across different approaches rather than limiting the review to a single category of data leakage research.

Ramachandiran *et al.* [1] investigated the application of machine learning to data leakage detection, covering techniques such as anomaly detection, user profiling, risk assessment, network-traffic analysis, and natural language processing of communication data. Their proposed insider-threat detection model reported an accuracy of 85% with a false-positive rate of 0.78% and employed SMOTE to address class imbalance. However, the work primarily focuses on identifying anomalous insider behavior and does not provide an endpoint mechanism for preventing a transfer before its completion.

Verma *et al.* [2] reviewed data leakage detection and prevention approaches based on the distributor–agent model. Their study examined techniques such as minimum-overlap data allocation, insertion of fake objects, and guilt-probability estimation. These approaches are mainly concerned with determining the probable source of a leaked dataset after distribution. Consequently, they provide valuable support for forensic investigation and accountability but do not directly intercept an unauthorized transfer while it is occurring.

Guan and Chen [3] proposed a Private Data Leak Analysis (PDLA) technique that combines taint analysis with a convolutional neural network (CNN) classifier to track private-information variables within source code. Their approach achieved a recall of 95.04%, compared with 82.64% obtained using the Find Security Bugs static analysis tool. Although the results demonstrate the potential of deep learning for privacy-leakage analysis, the approach is designed for source-code analysis rather than monitoring end-user activities such as file transfers, clipboard operations, or uploads.

Komal *et al.* [4] developed a cloud-oriented data leakage detection approach that integrates machine-learning classification, role-based access control, and behavioral monitoring. The approach demonstrates the usefulness of combining access control with behavioral analysis in cloud environments. However, its primary focus is cloud storage, and it does not address endpoint-level leakage channels such as USB devices, clipboard operations, screenshots, printing, or mobile-device transfers.

Papadimitriou and Garcia-Molina [5] introduced a foundational model for data leakage detection based on guilt probability and fake-object distribution. The approach provides a theoretical basis for determining the probability that a particular data recipient is responsible for a subsequent leak. Despite its importance in the development of allocation-based leakage detection, the method is primarily intended for post-leak attribution and does not perform real-time content-based interception.

Al-Shehari and Alsowail [6] investigated insider data leakage detection using machine-learning classifiers together with One-Hot Encoding and SMOTE. Their work demonstrates the importance of appropriate preprocessing and class-balancing techniques when developing machine-learning models for insider-threat detection. These observations are relevant to the present research, particularly in the preparation of synthetic training data and handling of imbalanced security-related classes.

Silowash *et al.* [7], through the *Common Sense Guide to Mitigating Insider Threats* developed by CERT and the U.S. Secret Service, presented organizational practices for reducing risks associated with insider threats. The recommendations emphasize preventive controls, monitoring, access management, and graduated responses to suspicious activities. These principles influenced the policy-oriented design of the proposed system, in which detected events can result in different responses such **as** ALLOW, WARN, BLOCK, QUARANTINE, and ADMIN_ALERT**,**
depending on the evaluated risk.

Le, Zincir-Heywood, and Heywood [8] examined the effect of

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

data granularity on insider-threat detection performance. Their work highlights the importance of the characteristics and level of detail present in the training data when evaluating machine-learning-based detection systems. This finding is particularly relevant to the use of synthetic datasets in the present research, where performance obtained from controlled data must not automatically be interpreted as equivalent to performance in a real-world environment.

Yin Fan *et al.* [9] investigated a data distribution strategy based on minimizing overlap between data records assigned to different agents. The approach aims to reduce the possibility of identifying a complete dataset from any individual recipient and thereby limits the impact of a subsequent leakage. However, the prevention mechanism is primarily structural, as it determines how data is distributed before access occurs, rather than inspecting and controlling an active transfer at the endpoint.

Holambe, Shinde, and Bhosale [10] investigated guilt-detection techniques involving sample-based and explicit fake-object algorithms for improving the identification of a probable data leaker. Such techniques can strengthen post-incident investigation by providing additional evidence linking a leaked dataset to its potential source. However, similar to other fake-object-based approaches, the primary objective is attribution after distribution rather than real-time prevention of the transfer itself.

The reviewed studies demonstrate that existing data leakage research can broadly be divided into several categories. Allocation-based and fake-object approaches [2], [5], [9], [10] primarily concentrate on identifying or attributing responsibility for a leakage after data has been distributed. Machine-learning and behavioral-analysis approaches [1], [6], [8] improve the ability to identify suspicious activities and assign risk, but commonly focus on detection rather than direct intervention. Domain-specific approaches, such as source-code analysis [3] and cloud-based monitoring [4], address particular environments and therefore do not provide comprehensive endpoint-level protection.

A further limitation is the separation between detection and enforcement**.** Existing approaches may identify suspicious behavior or sensitive information, but relatively fewer academic implementations combine content classification, behavioral anomaly detection, transparent risk assessment, and configurable policy enforcement within a single endpoint agent. In particular, there is limited work demonstrating pre-action, content-based intervention at the operating-system level for common endpoint activities such as copying files to USB devices or network shares.

The proposed SentinelDLP system integrates rule-based content analysis, TF-IDF and Logistic Regression-based classification, endpoint monitoring, risk assessment, and policy-management components. The machine-learning components are intended to contribute to the identification of sensitive content and potentially unusual activities. A risk-scoring component and configurable policy engine are included to support security decisions and administrative review. The system generates alerts and records incidents when relevant activities are detected. Print prevention is implemented for sensitive printing activities, while USB transfer blocking requires a separate administrator command and currently applies to all files. The Windows Shell Copy Hook is present in the source code but does not currently provide the intended selective, pre-transfer file-copy prevention functionality.

III. RELATED WORK

The literature review reveals several gaps that are relevant to the proposed system. First, many machine-learning-based approaches [1], [6]–[8] focus mainly on classification or anomaly-detection performance and do not provide an enforcement mechanism to prevent an identified transfer. Second, allocation-based, guilt-probability, and fake-object techniques [2], [5], [9], [10] are primarily post-incident approaches that identify the probable source of a leak rather than stopping the transfer in real time.

Third, cloud-based approaches [4] mainly focus on cloud storage and access control and provide limited coverage of endpoint channels such as USB, clipboard, screenshots, printing, and mobile-device transfers**.** Similarly, source-code-based analysis [3] does not address real-time monitoring of user file operations.

Another limitation identified in existing research is the separation between detection and enforcement. The proposed SentinelDLP system addresses this challenge by integrating rule-based content analysis, machine-learning-based classification, endpoint monitoring, risk assessment, and policy-management components. The implementation supports alert generation and incident recording, while print prevention provides a working prevention mechanism for detected sensitive printing activities. USB transfer blocking is available through a separate administrator command; however, it currently blocks all USB file transfers rather than selectively enforcing content-based policies. The intended selective file-copy prevention mechanism through the Windows Shell Copy Hook has not been implemented successfully. These limitations identify areas requiring further development and evaluation.

IV. PROPOSED APPROACH

The proposed Machine Learning-Based Data Leakage Detection System follows a unified processing pipeline to monitor user activities across seven different channels. Each event is captured by the corresponding channel monitor and analyzed to determine its sensitivity level. The system then evaluates the user's surrounding activities to identify unusual behavioral patterns. These two analyses are combined to generate a transparent risk score, which is used by the policy engine to determine an appropriate response. Depending on the risk level and the capabilities of the monitored channel,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

the system may allow the activity, generate a warning, notify the administrator, or block the operation. On supported Windows endpoints, the system can also intercept and prevent certain operations before they are completed.

The process begins when the user accesses the SentinelDLP application and its supported monitoring components. The system monitors configured endpoint activities, including filesystem operations, USB connections, clipboard operations, screenshots, printing, MTP devices, and network shares. Captured activities are analysed using rule-based techniques and machine-learning-based content classification to identify potentially sensitive information. The system can generate alerts and incident records for administrative review. Print prevention is available for sensitive printing activities. USB transfer blocking requires a separate command to be executed with administrator privileges and currently blocks all USB file transfers after activation. Other channel-specific prevention mechanisms must be evaluated according to their actual implementation and supported functionality.

Based on the results of these analyses, the system calculates a risk score and evaluates the activity against predefined security policies. If the activity is classified as normal and its risk level is acceptable, it is allowed to proceed. If suspicious or potentially unsafe behavior is detected, the system takes an appropriate action according to the severity of the risk and the interception capabilities of the respective channel. These actions may include issuing warnings, notifying the administrator, recording the incident, or blocking supported file transfer and data-sharing operations.

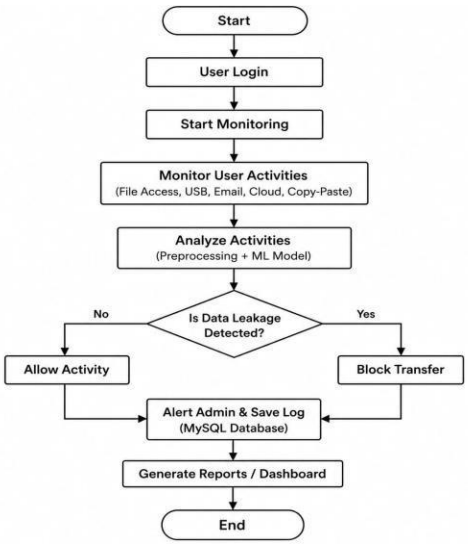


Fig. 1 Proposed Approach

All relevant events, risk assessments, and security actions are recorded in the database to support auditing and incident investigation. The system also provides reports and dashboards that help administrators review user activities, identify potential threats, and assess the overall security status of the monitored endpoints. This integrated approach combines content-based detection, behavioral analysis, risk assessment, and policy enforcement to improve the identification and prevention of potential data leakage incidents.

This section presents the proposed methodology through architecture, data-flow, class, and sequence diagrams. Each diagram illustrates a different aspect of the system, including its major components, the movement of data, the relationships between classes, and the sequence of interactions involved in detecting and responding to potential data leakage events.

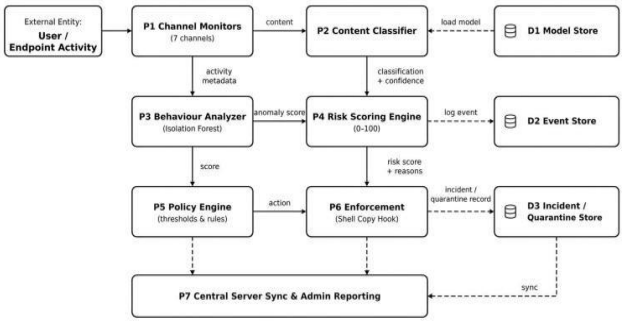


Fig. 2 Data Flow Diagram of the Detection and Prevention Pipeline

Figure 5.2 illustrates the flow of a single user activity event through the proposed system using a Data Flow Diagram (DFD). The process begins when endpoint activities are captured by the seven channel monitors (P1). The collected data is then divided into two parallel processing streams: content analysis and behavioral analysis.

In the first stream, the extracted content is sent to the Content Classifier (P2). This component uses a trained Term Frequency–Inverse Document Frequency (TF-IDF) vectorizer and Logistic Regression model, loaded from the Model Store (D1), to classify the sensitivity of the content and generate a corresponding confidence score.

In the second processing stream, activity metadata may be analysed by the Isolation Forest component, which is intended to identify unusual behavioural patterns. However, the integration of the model's output into the complete application pipeline requires further verification. Therefore, end-to-end behavioural anomaly detection should not be considered conclusively established until the model's output and its use by the application have been confirmed.

The outputs of both analysis components are forwarded to the Risk Scoring Engine (P4), which combines the content sensitivity and behavioral anomaly results to calculate a unified risk score ranging from 0 to 100. The engine also generates human-readable explanations for the assigned score. The resulting risk assessment and associated event details are recorded in the Event Store (D2).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

The policy-management component is intended to evaluate detected activities and determine an appropriate security response. The implemented system generates alerts and records incidents, and print prevention blocks detected sensitive printing activities. USB transfer blocking is activated separately through an administrator command and blocks all USB file transfers once activated. Although the

Windows Shell Copy Hook is included in the source code, it does not currently provide the intended selective, pre-transfer blocking functionality. The effectiveness of other channel-specific prevention actions requires verification against the deployed implementation.

Finally, the Central Server Synchronization and Admin Reporting process (P7) coordinates the exchange of events, incident records, and policy updates between the monitored endpoint and the Admin Console. This enables administrators to review detected threats, monitor security events, and manage applicable policies through a centralized interface.

V. SYSTEM ARCHITECTURE

SentinelDLP follows a three-tier architecture, as illustrated in Figure 5.1. The architecture is divided into the Endpoint Tier, Application Tier, and Data Tier, with each tier performing specific functions to support data leakage detection, risk assessment, policy enforcement, and centralized administration.



Fig. 3 System Architecture

The Endpoint Tier consists of the protected Windows workstation or laptop on which the SentinelDLP monitoring components operate. The system includes channel monitors, a content-classification component, an Isolation Forest component intended for behavioural analysis, a risk-scoring component, policy-management functionality, and the Windows Shell Copy Hook implementation. Rule-based techniques and the content-classification model support the identification of sensitive information. Print prevention is implemented for detected sensitive printing activities. USB transfer blocking requires a separate administrator command and currently blocks all USB file transfers after activation. The Shell Copy Hook has not been implemented successfully as a functioning selective file-copy prevention mechanism. The effectiveness and integration of the behavioural-analysis component also require further verification.

The Application Tier comprises two major components: the Central Server and the Admin Console. The Central Server is developed using FastAPI and is responsible for device authentication, event collection, incident management, and the distribution of security policies. The Admin Console is a

desktop application developed using PySide6 that provides administrators with an interface to manage registered devices, security incidents, quarantined files, policy exceptions, security policies, and reports. Together, these components enable centralized monitoring and administration of the protected endpoints.

The Data Tier uses an SQLite database to maintain persistent records required for the operation of the system. The database consists of nine tables that store information related to registered devices, monitored events, security incidents, policies, quarantine records, policy exceptions, and user accounts, along with other supporting records. This structured storage supports event tracking, incident investigation, policy management, and administrative reporting.

Overall, the three-tier architecture separates endpoint protection, application-level management, and persistent data storage. This separation improves the modularity and maintainability of SentinelDLP while supporting local security enforcement and centralized administration.

VI. WORKING OF THE SENTINELDLP ADMIN CONSOLE

The SentinelDLP Admin Console is a desktop application developed using PySide6 that enables administrators to monitor endpoint security, investigate potential data leakage incidents, and manage system-wide security policies. The console consists of eight tabs: Dashboard, Devices, Incidents, Quarantine, Exceptions, Policies, Reports, and Settings. Each tab provides specific administrative functions and communicates with the Central Server through its REST API. Access to the console is restricted through a login interface to prevent unauthorized administrative access.

1. Admin Login: The administrator opens the SentinelDLP Admin Console and enters the required login credentials. After successful authentication, access is granted to the administrative interface and its available security management features.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026



Fig. 4 SentinelDLP Admin Console login screen

2. Dashboard: The Dashboard tab provides a consolidated overview of the system's security status. It displays key metrics, including the total number of events processed, blocked events, open incidents, and online devices. It also presents a breakdown of events by monitoring channel and classified content by sensitivity level. This information helps administrators quickly identify unusual activity and assess the current security status of monitored endpoints.

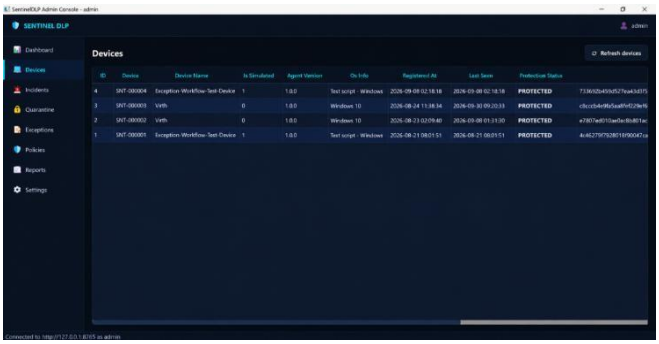


Fig. 5 Devices tab, listing registered and protected endpoints

3. Device Management: The Devices tab displays information about registered endpoints, including the agent version, operating system, registration time, last-seen time, and current protection status. This allows administrators to identify which devices are actively reporting to the Central Server. Devices that fail to send heartbeats within the configured interval can be visually distinguished, helping administrators investigate disconnected endpoints or agents that may have stopped functioning.



Fig. 6 Dashboard tab showing the security overview

4. Incident Monitoring: The Incidents tab displays security events that trigger policy actions beyond the ALLOW category. Each record includes relevant details such as the device, monitoring channel, file name, content classification, calculated risk score, action taken, and destination. Administrators can review these records to investigate suspicious activities and manage incidents according to the available system functionality. The incident records also support auditing and security analysis by maintaining a history of detected and actioned events.

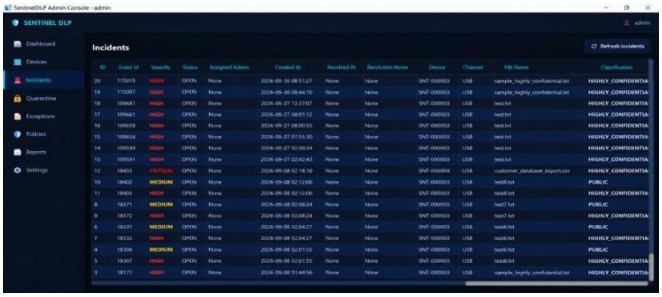


Fig. 7 Incidents tab, listing risk-scored and actioned events

5. Quarantine Management: The Quarantine tab provides an interface for quarantine-related administration. However, the complete file-quarantine functionality has not been implemented successfully. The current system should therefore not be considered capable of reliably moving detected files into a restricted location. Implementing and validating secure file quarantine remains a future development task.

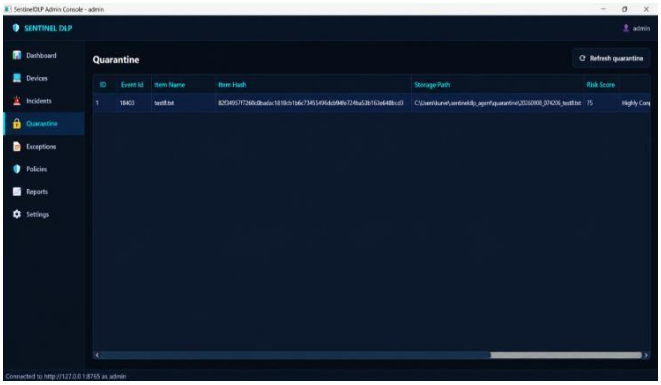


Fig. 8 Quarantine tab

6. Exception Management: The Exceptions tab provides information about administrator-approved exceptions to existing security policies. These exceptions may apply to specific files or destinations that would otherwise be restricted by the system. The tab displays the relevant exception details, approval status, and expiry information, allowing administrators to review and manage authorized deviations from standard security rules.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

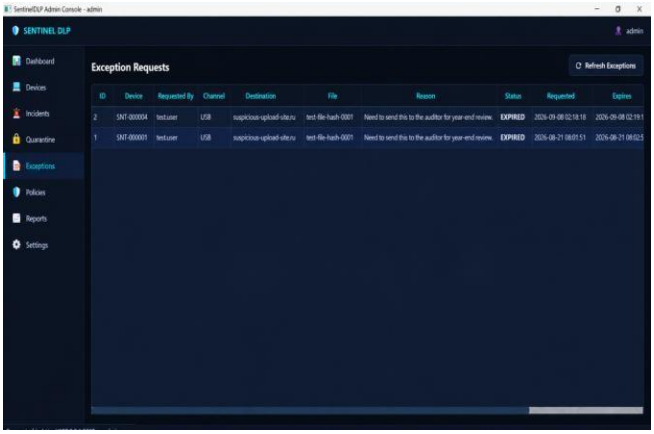


Fig. 9 Exception Requests tab

7. Security Policy Configuration: The Policies tab provides an interface for configuring security policies and related administrative settings. The available functionality should be interpreted according to the actions supported by the current implementation. Print prevention is implemented for detected sensitive printing activities, while USB transfer blocking requires a separate administrator command and currently blocks all USB file transfers after activation. Quarantine and selective file-copy prevention through the Windows Shell Copy Hook are not currently functioning successfully. The implementation of other policy actions should be verified before they are described as operational.

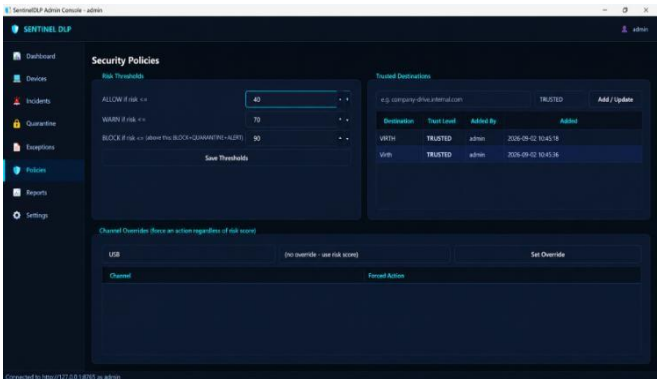


Fig. 10 Security Policies configuration tab

8. Reports and Analytics: The Reports tab presents security-related statistics for a selected reporting period. It provides information about total events, blocked events, and the distribution of content classifications. These reports help administrators review system activity, identify recurring security concerns, and evaluate the effectiveness of the configured data leakage prevention policies.



Fig. 11 Analytics and Reports tab

9. System Settings: The Settings tab provides access to basic administrative configuration options. It allows the logged-in administrator to view the connected Central Server URL and change the account password, according to the implemented functionality. These settings help maintain the administrative account and provide visibility into the server connection configuration.

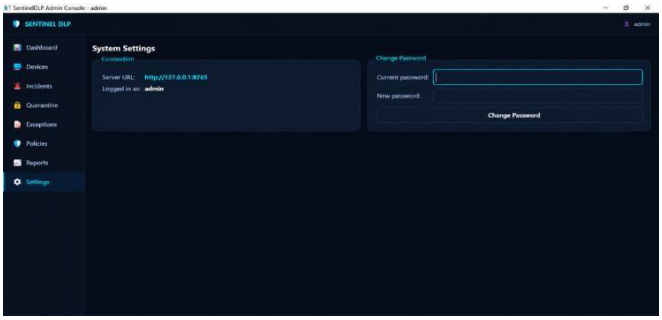


Fig. 12 System Settings tab

VII. TECHNOLOGY STACK AND IMPLEMENTATION

SentinelDLP is developed using Python and machine learning technologies to detect and prevent data leakage on Windows endpoints.

Programming Language: Python is used to develop the Endpoint Agent, Central Server, and Admin Console.

Machine Learning: Scikit-learn is used for the TF-IDF and Logistic Regression-based content-classification model. Isolation Forest is also included in the project for behavioural anomaly detection. However, the integration of Isolation Forest's output into the complete application pipeline requires further verification. Joblib is used to store trained models where applicable.

Endpoint Monitoring: Watchdog, WMI, and pywin32 monitor file activities, USB connections, clipboard events, and other supported channels.

Backend and Database: FastAPI and Uvicorn handle server communication, while SQLite stores events, incidents, policies, devices, and user records.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Admin Console: PySide6 provides a desktop interface for device monitoring, incident management, policy configuration, and report generation.

Implementation: The system includes risk-scoring and policy-management components intended to support security decisions. The application also generates alerts and records incidents. Print prevention is implemented for detected sensitive printing activities. USB transfer blocking is activated through a separate administrator command and currently blocks all USB file transfers after activation. The intended selective file-copy prevention mechanism through the Windows Shell Copy Hook and the complete file-quarantine functionality are not currently functioning successfully.

VIII. MATCHING AND RETRIEVAL TECHNIQUES

SentinelDLP uses a combination of pattern matching, keyword matching, and machine learning to identify sensitive information in monitored activities. Regular expressions detect predefined patterns, while keyword matching identifies terms associated with confidential data. These techniques are combined with TF-IDF and Logistic Regression to classify content into different sensitivity levels. This hybrid approach improves the identification of potentially sensitive information.

The system includes an Isolation Forest component intended to analyse activity data and identify unusual data-handling behaviour. However, the integration of the model's output into the complete application pipeline has not yet been verified. Consequently, the effectiveness of end-to-end behavioural anomaly detection cannot be conclusively established. The application also includes a risk-scoring component intended to support security decisions. Further verification is required to establish how behavioural-analysis results contribute to the final risk score and how the resulting score influences policy enforcement.

IX. RESULTS AND DISCUSSION

The results demonstrate that SentinelDLP combines rule-based techniques and machine-learning-based content classification to identify potentially sensitive information. The system also includes an Isolation Forest component intended for behavioural anomaly detection; however, the integration of its output into the complete application pipeline requires further verification. The application provides alerts and incident records for administrative review, and print prevention is implemented for detected sensitive printing activities. USB transfer blocking requires a separate administrator command and currently blocks all USB file transfers after activation.

Based on the calculated risk and configured policies, the system can allow an activity, display a warning, block an operation, or trigger quarantine and administrator alerts. The Admin Console provides facilities for monitoring devices, reviewing incidents, managing policies, and generating reports.

The evaluation results show strong performance on the synthetic datasets used for testing. However, these results do not guarantee the same level of accuracy in real-world environments. The system's effectiveness also depends on the monitored channel and the prevention mechanisms supported by the endpoint. Overall, SentinelDLP demonstrates the potential of combining machine learning and rule-based techniques to support data leakage detection and prevention.

X. FUTURE SCOPE

The future scope of SentinelDLP includes improving the accuracy of data leakage detection by training machine learning models on larger and more diverse datasets. Advanced classification techniques can be explored to improve the identification of sensitive information and reduce false alerts. The system can also be extended to support additional data transfer channels and stronger prevention mechanisms.

Further development may include improved behavioural analysis, enhanced monitoring capabilities, and more detailed reporting through the Admin Console. Cloud-based deployment and integration with organizational security systems could also be explored to support larger environments. These improvements would help make SentinelDLP more adaptable, scalable, and effective in detecting and preventing unauthorized data leakage.

Further development of SentinelDLP will focus on improving the reliability of prevention mechanisms and integrating machine-learning outputs into the complete detection pipeline. The Windows Shell Copy Hook requires further development and testing to achieve selective file-copy prevention based on content sensitivity. The current USB prevention mechanism should also be improved to distinguish between sensitive and non-sensitive files rather than blocking all transfers after activation.

Additional work will include implementing and validating secure file quarantine, verifying the integration of Isolation Forest into risk assessment, and evaluating the risk-scoring mechanism. Further testing will be required to establish the effectiveness of prevention policies across supported monitoring channels. These improvements will help establish the reliability of SentinelDLP under practical endpoint conditions.

XI. CONCLUSION

This research presented SentinelDLP, a machine-learning-based data leakage detection and prevention system designed for Windows endpoints. The system combines rule-based techniques with TF-IDF and Logistic Regression for content classification into different sensitivity levels. It also incorporates endpoint monitoring, an administrative

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

dashboard, incident recording, and components intended to support behavioural analysis and risk-based security decisions.

The implemented system provides content classification and monitoring capabilities across multiple endpoint channels. Print prevention blocks detected sensitive printing activities and generates corresponding alerts and incident records. USB transfer blocking is available through a separate command that requires administrator privileges. However, the current mechanism blocks all file transfers to USB devices after activation rather than selectively enforcing policies according to file sensitivity.

The Windows Shell Copy Hook is present in the source code but has not been implemented successfully as a functioning selective file-copy prevention mechanism. In addition, the integration of Isolation Forest into the complete detection pipeline and the secure file-quarantine functionality require further verification or development. The content-classification model was evaluated using separate training and testing datasets, and its reported performance should be interpreted in the context of the synthetic data used for evaluation.

Overall, SentinelDLP provides a foundation for combining machine-learning-based content classification, endpoint monitoring, and selected prevention mechanisms. Further development and testing are necessary to improve selective file-transfer prevention, validate behavioural anomaly detection, and establish the reliability of the complete system in practical environments.

REFERENCES

[1] P. Papadimitriou and H. Garcia-Molina, “A Model for Data Leakage Detection,” IEEE Transactions on Knowledge and Data Engineering, 2011.
[2] S. V. Shaj and K. P. Kaliyamurthie, “A Review on Data Leakage Detection,” International Journal of Computer Science and Mobile Computing (IJCSMC), 2013.
[3] P. Papadimitriou and H. Garcia-Molina, “Data Leakage Detection,” IEEE Transactions on Knowledge and Data Engineering, 2013.
[4] J. Sawant, H. Kadam, and S. Menghani, “Data Leakage Detection in an Organization,” International Journal of Scientific & Engineering Research, 2013.
[5] X. Shu, J. Zhang, D. D. Yao, and W.-C. Feng, “Fast Detection of Transformed Data Leaks,” IEEE Transactions on Information Forensics and Security, 2016.
[6] S. Praveen Kumar, Y. Srinivas, D. Suba Rao, and A. Kumar, “A Novel Model for Data Leakage Detection and Prevention in Distributed Environment,” International Journal of Engineering and Technical Research, 2016.
[7] J. Sawant, H. Kadam, and S. Menghani, “Data Leakage Detection in an Organization,” International Journal of Scientific & Engineering Research, 2017.
[8] I. Gupta and A. K. Singh, “A Framework for Malicious Agent Detection in Cloud Computing Environment,” International Journal of Advanced Science and Technology (IJAST), 2017.
[9] I. Gupta and A. K. Singh, “A Probability-Based Model for Data Leakage Detection Using Bigraph,” Seventh International Conference on Communication and Network Security (ICCNS), 2017.
[10] K. Kaur, I. Gupta, and A. K. Singh, “A Comparative Evaluation of Data Leakage/Loss Prevention Systems (DLPS),” Fourth International Conference on Computer Science & Information Technology (CS & IT-CSCP), 2017.
[11] I. Gupta and A. K. Singh, “Dynamic Threshold-Based Information Leaker Identification Scheme,” Information Processing Letters, 2017.
[12] U. Arora, S. Verma, I. Gupta, and A. K. Singh, “Implementing Privacy Using Modified Tree and Map Technique,” Third International Conference on Advances in Computing, Communication & Automation (ICACCA), 2017.
[13] G. Raghavendra Rao, D. D. Doye, and S. Niranjan, “A Watermarking Technique for Securing Multimedia Data Against Digital Data Leakage.” Publication year and venue not provided.
[14] K. Kaur, I. Gupta, and A. K. Singh, “Data Leakage Prevention: Email Protection via Gateway,” Journal of Physics: Conference Series, IOP Publishing, 2018.
[15] I. Gupta and A. K. Singh, “A Confidentiality Preserving Data Leaker Detection Model for Secure Sharing of Cloud Data Using Integrated Techniques,” Seventh International Conference on Smart Computing and Communication Systems (ICSCC), 2019.
[16] A. Nag, A. Kesharwani, A. Tiwari, I. Gupta, B. Sharma, and A. K. Singh, “Potential and Extension of Comparative Internet of Things,” Second International Conference on Computer Networks and Inventive Communication Technologies, 2019.
[17] Y. G. B. Yadav, P. C. Bhaskar, and R. K. Kamat, “Assessing the Guilt Probability in Intentional Data Leakage,” International Journal of Computer Science and Information Technologies, 2023.
[18] C. Mercy Praba and G. Satyavathy, “A Technical Review on Data Leakage Detection and Prevention Approaches,” Journal of Network Communications and Emerging Technologies (JNCET), 2023.
[19] Dr. A. R. Pon Periyasamy and E. Thenmozhi, “Data Leakage Detection and Data Prevention Using Algorithm,” International Journal of Advanced Research in Computer Science and Software Engineering, 2024.
[20] Y. Fan, W. Lina, Y. Rongwei, and M. Xiaoyan, “A Distribution Model for Data Leakage Prevention,” IEEE International Conference on Mechatronic Sciences, Electric Engineering and Computer, Shenyang, China, 2025.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)