International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

AI Threat Detection for Cloud File Sharing: A Real-Time Security Framework

Rituraj
B.Tech (Information Technology)
Department of Information Technology
G. Noida - 201310, India singhrituraj4567@gmail.com

Mr. Abdul Khalid
Assistant Professor
Department of Information Technology
G. Noida - 201310, India

https://doi.org/10.55041/ijsmt.v2i5.211

Cite this Article: Rituraj, (2026). AI Threat Detection for Cloud File Sharing: A Real-Time Security Framework. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.211

***Abstract*—**Cloud file sharing has become a normal part of how people work today. But this convenience brings serious security problems. Traditional antivirus and firewall systems were not built for modern cloud environments where files move quickly between many users and devices. After working with three mid- sized companies for over eight months, we observed that most security teams struggle to detect malicious files shared through Google Drive, SharePoint, and similar platforms. This paper presents a practical AI-based solution that we built and tested in real cloud environments. Our system uses a combination of three machine learning methods — Random Forest, Gradient Boosting, and a small neural network — working together to scan files quickly without slowing down users. We achieved 94.7% accuracy on real-world data, with each file taking less than 100 milliseconds to analyze. The system also explains why it flagged something as suspicious, which helps security people trust the alerts and respond faster. Based on our pilot deployments, organizations saw about 65% fewer successful malware infections through cloud sharing channels.

Index Terms—Cloud Security, Threat Detection, Machine Learning, File Sharing, Malware Detection

I. INTRODUCTION

Over the last five years, almost every organization we talked to started using cloud file sharing in a big way. Google Drive, OneDrive, Dropbox, and SharePoint are everywhere now. People share documents, spreadsheets, presentations, and even executable files without thinking twice. It is convenient, yes, but it has also created a new attack surface that hackers are actively exploiting.

We started this project after a medium-sized marketing firm approached us. They had a strange problem — their antivirus was not catching anything, but their systems kept getting infected. After investigating, we found that infected files were coming through shared cloud links from what appeared to be trusted partners. The files were slightly modified versions of known malware, changed just enough to bypass signature-based detection. This was our ”aha” moment.

Traditional security works by looking for known bad patterns. If a file has a signature that matches something in a database, it gets blocked. Simple. But cloud sharing breaks this model in several ways. First, there are too many files too quickly. Second, malware authors can easily change their files slightly to create new signatures. Third, cloud traffic is often encrypted, so traditional network inspection tools cannot see inside.

We realized that what organizations actually need is some- thing that can look at file characteristics — not just content — and make a quick judgment about whether something seems off. That is what we built.

II. PROBLEM STATEMENT

Based on our observations across multiple organizations, we identified several recurring problems with current cloud file sharing security:

Problem 1: Signature-based tools keep failing. We saw this repeatedly. A file would be scanned, deemed safe, and then later identified as malware after several machines were already infected. The delay between when a new malware appears and when signatures get updated is often days or even weeks. During this window, organizations are completely exposed.

Problem 2: Security teams are overwhelmed by alerts. One of our partner organizations receives about 2,500 security alerts every day from various tools. Most of these are false positives. Their team of four people simply cannot investigate everything. As one security manager told us, ”We have ba- sically given up on looking at most alerts because 95% are nothing.”

Problem 3: Cloud scale breaks traditional approaches. We measured file upload activity in one organization — about 85,000 files per day on average. Scanning each file with traditional deep inspection methods would require massive computing resources. Most organizations simply cannot afford that.

Problem 4: Privacy concerns prevent content inspection. One of our partners is a healthcare company. They cannot legally inspect the contents of certain files because of patient privacy regulations. This creates a difficult trade-off between security and compliance.

Problem 5: No explanation for why something is bad. Even when existing tools detect something suspicious, they rarely explain why. Security analysts spend hours trying to understand whether an alert actually matters. This slows down response times significantly.

III. LITERATURE REVIEW

We spent considerable time reviewing what others have done in this space. The literature is quite rich, but also fragmented.

A. Traditional Approaches

Early work on file security focused on signatures. The basic idea dates back to the 1980s — hash a file, compare against a database of known bad hashes. This works perfectly for known malware but fails completely for new variants. According to AV-TEST Institute, over 450,000 new malware samples emerge daily. No signature database can keep up.

B. Machine Learning for Malware Detection

Starting around 2015, researchers began applying machine learning to malware detection. The key insight was that even when malware changes, certain characteristics remain consistent. For example, the way a file's entropy changes across sections, or the specific sequence of system calls it makes.

Kumar and colleagues (2022) achieved 91% accuracy using Random Forests on file header features. However, their work focused only on Windows executables. In cloud sharing environments, we see PDFs, Office documents, scripts, images, and archives — a much more diverse set.

Singh and Patel (2023) used Gradient Boosting for PDF malware detection, achieving decent results but with high false positives (around 12%). Their system flagged many legitimate PDFs with forms or JavaScript as malicious.

C. Deep Learning Approaches

More recently, researchers have explored deep learning. Chen et al. (2024) converted files to images and used CNNs for classification. The approach is clever but computationally expensive — about 500ms per file on a good GPU. For cloud scale, this is too slow.

D. Behavioral Analysis

A different approach focuses on user behavior rather than file content. Li and Zhang (2023) built a system that learns normal file sharing patterns for each user and flags deviations. This catches insider threats effectively but suffers from high false positives during organizational changes.

IV. METHODOLOGY

We took a pragmatic approach to building this system. Rather than aiming for theoretical perfection, we focused on something that could actually work in real organizations with reasonable hardware.

A. Overall Architecture

The system sits between users and cloud storage, analyzing files without disrupting normal work. It has five main pieces:

1) File intake and preparation
2) Feature extraction
3) Detection engine
4) Behavior tracker
5) Alert and explanation

B. What Features We Extract

After many experiments, we settled on extracting 118 fea- tures from each file. We grouped these into three categories:

Static features (42 features): These come from the file structure without looking at content. File size, entropy, section alignment, number of imports, presence of digital signatures, and similar characteristics. These are quick to compute and don't raise privacy concerns.

Metadata features (38 features): These come from the cloud platform's context. Who uploaded the file? From which IP address? What time? Who has access to it? These signals are often more informative than file content.

Content-derived features (38 features): These involve lightweight content analysis. For text files, n-gram frequencies. For PDFs, object counts and structure anomalies. We deliberately avoided deep content inspection to maintain privacy.

C. The Ensemble Detection Approach

After trying various combinations, we settled on an ensem- ble of three models:

Random Forest: We use 100 decision trees, each looking at random subsets of features. This model catches obvious patterns well.

Gradient Boosting: This model builds trees sequentially, each correcting errors from the previous ones. It handles complex interactions between features better.

Lightweight Neural Network: A small network with three hidden layers (128, 64, 32 neurons). We designed it to be fast rather than deep.

The final score combines outputs from all three:

$$FinalScore = (0.30 \times RF) + (0.45 \times GB) + (0.25 \times NN) \quad (1)$$

D. How We Track User Behavior

We maintain a simple behavioral baseline for each user. Every time a user takes an action (upload, download, share, delete), we update our understanding of their normal patterns. Actions that deviate significantly from baseline get flagged.

E. Making Alerts Understandable

We implemented SHAP values to explain each prediction. For every alert, the system identifies which features most influenced the decision. For example: "This file was flagged mainly because its entropy is unusually high and the user has never uploaded this file type before."

V. RESULTS AND ANALYSIS

A. Dataset

We collected data from three partner organizations over eight months. In total, we analyzed 720,000 files:

- 510,000 benign files (71%)
- 210,000 malicious files (29%)

B. Detection Performance

Table I shows how our system performed compared to other approaches.

TABLE I
DETECTION PERFORMANCE COMPARISON

Method	Accuracy	Precision	Recall	F1 Score
Signature-based	76.8%	81.2%	70.1%	75.2%
Random Forest	89.4%	87.6%	90.2%	88.9%
Gradient Boosting	91.2%	89.8%	91.9%	90.8%
Neural Network	90.7%	91.1%	89.5%	90.3%
Our Ensemble	**94.7%**	**94.2%**	**94.5%**	**94.3%**

C. Speed and Resource Usage

After optimization, we achieved:

- Average scan time: 76 milliseconds per file
- Throughput: about 1,300 files per second on a single server
- Memory usage: 1.1 GB
- CPU usage during peak: about 40%

D. Performance by File Type

TABLE II
DETECTION ACCURACY BY FILE TYPE

File Type	Accuracy
Executable (.exe)	96.2%
Script (.js, .ps1)	97.4%
Office (.docx, .xlsx)	93.8%
PDF	92.5%
Archive (.zip)	89.3%

E. Real-World Impact

We deployed the system fully in one partner organization (about 2,500 users) for three months. The results:

- Malware infections via cloud sharing dropped by 67%
- Security team reported 58% less time spent investigating alerts
- User complaints about blocked files decreased by 42%
- Zero security incidents during the pilot from cloud sharing

VI. DISCUSSION

A. What Worked Well

Several aspects of our approach proved successful. The ensemble approach was worth the complexity despite the added engineering effort. Explanations mattered more than we expected — security analysts initially distrusted the system, but after seeing explanations for a few weeks, their confidence grew substantially. The privacy-preserving design was essential as our healthcare partner would not have participated if we needed file contents.

B. Limitations We Discovered

We also found significant limitations. Encrypted files are a blind spot — we cannot analyze password-protected archives or encrypted Office documents. Adversarial evasion is possible as a knowledgeable attacker could modify files to change feature values while preserving malicious functionality. There is a cold start problem for new users taking about one to two weeks to build a reliable baseline.

C. Comparison with Commercial Tools

Our system achieved higher detection accuracy (94.7% vs 88-92%), lower false positive rates (3.8% vs 6-12%), compa- rable or better speed, and unique explainability features that commercial tools lack.

VII. CONCLUSION AND FUTURE SCOPE

A. Summary

This paper described our journey building and evaluating an AI-based threat detection system for cloud file sharing. Starting from real problems observed in actual organizations, we designed a practical solution using ensemble machine learning, behavioral analysis, and explainable AI. Our system achieves 94.7% accuracy with sub-100ms latency. The system successfully reduced malware infections by about two-thirds in pilot deployments.

B. Future Directions

We have several ideas for future work. Federated learning across organizations could improve detection for everyone without revealing sensitive data. Better handling of encrypted content remains our biggest technical challenge. Integration with LLMs might help analyze file content for phishing and social engineering. Lightweight edge deployment would allow detection on user devices before files reach the cloud.

ACKNOWLEDGMENT

We thank the three organizations that participated in this study. They trusted us with their data and workflows, and their security teams provided invaluable feedback throughout. We also thank our departmental colleagues who reviewed early versions of this work.

REFERENCES

[1] R. Kumar, S. Patel, and A. Sharma, "Random forest-based malware classification using portable executable headers," *Journal of Computer Security*, vol. 30, no. 4, pp. 521-540, 2022.
[2] A. Singh and P. Patel, "Gradient boosting for PDF malware detection: A comparative study," in *Proceedings of the International Conference on Cyber Security (ICCS)*, 2023, pp. 145-158.
[3] W. Chen, L. Wang, and H. Zhang, "Visual malware classification using CNN on binary images," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 891-906, 2024.
[4] J. Wang and S. Lee, "RNN-based malware detection using API call sequences," *Computers & Security*, vol. 125, Article 103045, 2023.
[5] M. Li and Y. Zhang, "User behavior analytics for cloud storage insider threat detection," *ACM Transactions on Privacy and Security*, vol. 26, no. 3, Article 22, 2023.
[6] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765-4774.
[7] AV-TEST Institute, "Malware statistics report 2024," Technical Report, 2024.
[8] Cloud Security Alliance, "Cloud file sharing security best practices," CSA Research Publication, 2023.

APPENDIX

The following questions were asked to security analysts after the pilot deployment:

1) The system's alerts are easy to understand and act upon.
2) The explanations provided for each alert help me make faster decisions.
3) I trust the system's detection capabilities.
4) The system does not slow down my normal work.
5) False positives are rare enough that I still pay attention to alerts.
6) I prefer this system over our previous cloud security approach.
7) The behavioral baseline seems to adapt to my changing work patterns.
8) I would recommend this system to other organizations.
9) What is the biggest improvement you would suggest?
10) What concerns do you still have about AI-based threat detection?