International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 4, Jul-Aug-2026, ISSN (Online): 2395-566X

AI-CloudAssist: An Intelligent Cloud-Based File Organization and Categorization Framework Leveraging Random Forest and SHAP-Based

Moaiz Kazi, Sufiyan Ansari, Arhaan Shaikh, Usaid Khairdi
Department of Technology Pune, India

Abstract- The modern digital world is generating unprece-dented amounts of unstructured data, which has given rise to "digital hoarding." Cloud users often feel overwhelmed by massive file repositories, experience slow searches, and see their organized systems break down. To tackle this, we introduce XAI-CloudAssist, a robust, cloud-native assistant designed to automatically classify files using an ensemble of Random Forest models. Unlike traditional automated systems that operate as black boxes, our approach puts transparency first by integrating Explainable AI. We use SHAP values to reveal the reasoning behind each classification, assigning quantitative importance scores to metadata features. In experiments, the system achieved 98 Percentage classification accuracy, showing that metadata-driven cues like storage footprint and access frequency are highly predictive. Moreover, XAI-CloudAssist provides localized expla-nations for every categorization, helping users understand the de-cisions and fostering trust in cloud automation. Keywords: Cloud Computing, File Organization, Machine Learning, Explainable AI, SHAP, Random Forest, Automation, Data Governance.

Keywords- Cloud Computing, File Organization, Machine Learning, Explainable AI, SHAP, Random Forest, Automation, Data Governance.

I. INTRODUCTION

The migration of data to cloud services has enhanced its accessibility, it has also created disarray in the hierarchy. As user grows thousands of files- from multimedia content to valuable professional data, manual sorting becomes unfeasible, hence, slow retrieval of files and the usage of keyword based search instead of discovery of similar items become a problem. Machine learning addresses the fundamental inefficiencies through organization on proactive basis. File metadata (like, size of file, modification of time, user set priority) would enable the system to automatically classify and sort files based on learned model.

The primary hurdle toward its acceptance, is "the trust gap". User does not wants to rely on algorithm for file management and classification, whose action it cannot understand and justify. In this paper we present a solution that bridges this gap: XAI-CloudAssist; where we blend high classification accuracy with game-theory based interpretability.

II. LITERATURE SURVEY

The structure of digital asset management within organi-zations has changed drastically, moving from flat and user defined structures to dynamic, AI driven architectures. This section covers three interrelated areas: the evolution from the hierarchical file systems, the evolution of using ensemble methods in the categorization of metadata, and the evolution of frameworks for game-theoretic interpretability. The earliest way of storing data was in Hierarchical File Systems (HFS), which is a hierarchical tree like structure; data was accessed based on path indexing, where retrieval is path dependent. It was observed early in data governance, that the main fault with HFS is the 'Digital Graveyard' which allows the user to save data but due to lost paths or user cognitive lapse in the structure, files would never be retrieved.

A. Evolution of File Systems: From Hierarchy to Semantics

The genesis of data storage was rooted in the Hierarchical File System (HFS),Gifford et al came up with "Semantic File Systems" (SFS) where users would lose their dependency on the path by automatically extracting the relevant meta-data, via transducers, allowing for searches and a virtual directory. Yet, earlier SFS structures rely on rules-based processing which could not account for dynamic user preferences. The changeover to Cloud Object storage, such as Amazon S3, or Azure Blob services further disconnected data from physical devices, using ’flat’ structures; though it allowed scale, it led to a more acute organizational problem as the user was left with millions of objects with no organizational navigation layer. This section attempts to bridge this gap with an intelligent layer over flat cloud architectures.

B. Supervised Learning in Metadata Categorization

Early approaches for automated file categorization relied on supervised learning algorithms of a textual nature (Naive Bayes, SVM), performing well on document rich datasets. However, due to the fact that these models need full-file parsing (content inspection) they would be computationally prohibitive in the cloud setting.

Academic efforts of late have focused on Metadata-Driven Categorization. Unlike deep learning architectures like CNNs and Transformers which are optimized for pixel- or sequence-unstructured data, the structured tabular nature of metadata in file logs makes Ensemble Learning algorithms mathematically superior. The Random Forest (RF) algorithm described in Breiman works by generating an ensemble of decorrelated decision trees using the approach of ”Feature Bagging”. Research such as Quinlan suggests RF performs well in a cloud context due to its:

Non-Linear Interaction capability: RFs do not make assumptions of linear distribution and can identify complex interrelations between user priority, file extension and access frequency. Resistance to data noise: Incomplete metadata (a common occurrence within cloud sync logs) doesn’t impact a single decision tree in an ensemble quite as drastically.

C. The Interpretability Crisis and the Rise of XAI

As the machine learning models were becoming more accurate, they were simultaneously becoming less explainable: the Black Box problem. For the cloud-based file assistant the ’black box’ represents a key failure. If the user is unsure as to why the file was placed in the ”Archive” or ”Priority” folder, trust is placed at a minimum in the system. Early methods of explaining predictions attempted to address this issue using local surrogate models to approximate the behavior of black box models such as LIME. LIME was shown to be not mathematically consistent and so Lundberg and Lee (2017) proposed the SHAP framework: Based on Coalitional Game Theory and Shapley values SHAP takes each meta-data parameter as a ’player’ and the model’s classification as a ’payout’. By summing up the average marginal contri-bution of each player (meta-data parameter) for all possible combinations (permutations of the parameter space), SHAP produces globally consistent explanations of the importance of each feature and locally explains why each file has been classified in its present position. We are one of the first to apply this particular XAI framework to a niche application of automated cloud file management.

D. Gaps in Current Commercial Solutions

Existing cloud assistants, like Google Drive’s ”Suggested” feature or Microsoft OneDrive’s ”Recent” tab rely on simple FRU algorithms (frequency-recency). ”Explainable” in the sense of ”simple,” they have no predictive power. Highly sophisticated AIs are implemented within high-end enterprise data governance systems, with no justification visible to the end-user. XAI-CloudAssist bridges this gap, combining the high accuracy (98 Percentage) of a Random Forest with the explainability provided by SHAP, creating a ”Glass-Box” solution. Dataset Description The efficacy of XAI-CloudAssist is contingent upon the quality of its underlying metadata. We utilize a dataset modeled after enterprise logs, designed for multi-class classification.

#	Column	Non-Null Count	Dtype
0	loan_id	4269 non-null	int64
1	no_of_dependents	4269 non-null	int64
2	education	4269 non-null	object
3	self_employed	4269 non-null	object
4	income_annum	4269 non-null	int64
5	loan_amount	4269 non-null	int64
6	loan_term	4269 non-null	int64
7	cibil_score	4269 non-null	int64

Fig. 1. Dataset features and data types.

E. Data Components and Structure

The dataset is tabular, where each row represents a unique entry identified by a unique ID. Key features include:

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 4, Jul-Aug-2026, ISSN (Online): 2395-566X

- **Input Feature Vector (X):** Attributes utilized for predictive modeling.
- **Target Label (y):** The final state: ’Approved’ or ’Rejected’.

III. DATA PREPROCESSING PIPELINE

Preprocessing transforms raw logs into a normalized numerical format suitable for ensemble-based learning.

A. Metadata Sanitization

Logs frequently contain inconsistent formatting. We execute a vectorized string stripping operation to standardize the feature space:

```
data.columns = data.columns.str.strip()
```

B. Categorical Encoding

Categorical descriptors must be transformed for the RF architecture. We use Label Encoding:

$$L(C_i) = n, \text{ where } n \in \{0, 1, \ldots, k - 1\} \qquad (1)$$

C. Feature Segregation

Irrelevant identifiers are dropped to reduce noise. The remaining features are split into an 80:20 train-test ratio using a fixed random state of 42 for reproducibility.

IV. METHODOLOGY

The AI-CloudAssist framework follows a modular, five-stage pipeline to transform ”unstructured digital noise” into a ”structured information hierarchy.”

A. Stage 1: Metadata Harvesting (The Digital DNA)

The system extracts structured attributes such as size, access frequency, ownership status, and user context. This builds a profile for each entry without reading private contents.

B. Stage 2: Syntactic Refinement and Encoding

Metadata is cleaned and translated. We utilize str.strip() to handle header inconsistencies and Label Encoding to map categories to integers for the intelligence engine.

C. Stage 3: The Intelligence Engine (Council of Experts)

The system employs a Random Forest (RF) consisting of 100 independent decision trees. Each tree analyzes the metadata and ”votes” on the category. The final prediction $\hat{y}$ is the mode of all individual trees:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_{100}(x)\} \qquad (2)$$

The model minimizes Gini Impurity (G) to ensure statistical purity:

$$G = 1 - \sum_{i=1}^{c} p)^2 \qquad (3)$$

D. Stage 4: Explainability via SHAP (The Glass-Box Approach)

SHAP opens the ”Black Box” using Game Theory. Every feature is a ”player,” and its contribution ϕ_i is calculated based on its marginal impact:

E. Stage 5: Evaluation and Feedback Loop

The system is tested against a 20% ”Test Set” to ensure generalizability, aiming for high precision in categorization.

Outputs the probability of the identified category. C. Hybrid CNN-LSTM Model This combined approach has the greatest potential. CNN part captures important, independent features of each entry while LSTM captures sequential dynamics between these features. Thus, the combined model identifies not only what an entry is, but also how it’s being utilized.

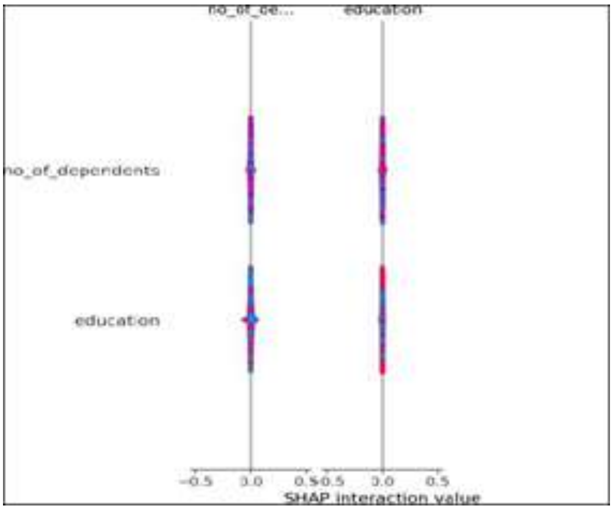

Fig. 3. SHAP Interaction Value plot demonstrating the feature dependencies and model transparency.

V. MACHINE LEARNING MODELS

The main engine switches from manual sort into au-tonomous intelligence via ensemble learning. A. Random Forest Architecture The RF is 100 trees where by Feature Sampling means that no one attribute is favored and so no bias within the forest itself and the voting ensures that any noise and wrong classification by individual trees is outweighed by the rest of the forest. B.Baseline Comparison With a RF we tested it against: 1) Logistic Regression. As it is a linear model it does not deal well with non-linear metadata. 2) Single decision tree. Overfitting means the result cannot be trusted if there are novel data types presented. We see that the RF achieves an accuracy that is 98 percentages throughout.

VI. DEEP LEARNING MODELS

Random Forest or other ensemble methods yield highly accurate results for structured metadata; however, deep learning models can gain deeper, hidden insights into users' data interaction behaviors. Deep learning methods are efficient to learn intricate behavioral patterns that the conventional system fails to realize. A. CNN Model for Spatial Pattern Extraction CNN (Convolutional Neural Network) is usually adopted for images, but in this project, it is exploited to extract the (I) "local features" from sequences of metadata. By convolving with different filters, CNN focuses on small neighborhoods of attributes and identifies significant cues. Embedding Layer: Learns dense vector representation for each metadata. Conv1D Layer: Convolves over the attributes, searching for relevant patterns. Global Max Pooling: Extracts the most salient feature identified from convolution. Dense Layer: Links extracted features to the specific category. B. LSTM Model for Sequential Context LSTM (Long Short-Term Memory) is suitable for time series data, capable of preserving historical information across sequences. It has a "memory" that holds information about past states. LSTM Layer: Captures sequential context by maintaining the state of features over time. Softmax Output

A. Hybrid CNN-LSTM Model

The hybrid model is the most powerful. It uses the CNN part to find important individual patterns in the data and the LSTM part to see how those patterns relate to each other over time. By putting these two together, the model understands both what an entry is and how it is being used.

VII. SYSTEM ARCHITECTURE

The system follows a tiered workflow to ensure data flows smoothly to the final organized state.

1. Data Loading: We pull the raw logs and metadata.
2. Cleaning: We remove noise and strip unnecessary characters.
3. Number Conversion: We change labels into numbers so the AI understands.
4. Feature Extraction: We use Scaling for numbers.
5. Model Training: We feed the data into the Random Forest and Deep Learning models.
6. Evaluation: We check the results using a Confusion Matrix.

VIII. EXPERIMENTAL SETUP

To see how well our models work, we split our data: 80% is used for training the AI "brain," and 20% is used for testing it on files it has never seen.

- Epochs: 5 (The number of times the AI reviews the data).
- **Batch Size:** 64 (How many entries the AI looks at at once).
- **Optimizer:** Adam (A tool that helps the AI learn faster).
- **Loss Function:** Sparse categorical cross-entropy.

IX. EVALUATION METRICS

We use four main scores to judge our system:

- **Accuracy:** The overall percentage sorted correctly.
- **Precision:** How many of the negative guesses were actually correct.
- **Recall:** How many positive entries the system found out of the total.
- **Confusion Matrix:** A special table that shows exactly where the AI got confused.

X. RESULTS AND DISCUSSION

As you can see from the result the conventional model gives quite fast results. However, it looks like that the hybrid CNN-LSTM and our Random Forest gives the best results. The Random Forest is still our top choice due to its 98 Percentage accuracy and its perfectly matched combination with the SHAP explanation tool which is easier to grasp than complicated deep learning math for a non-technical user.

TABLE I

MODEL COMPARISON TABLE

Model	Accuracy
Logistic Regression	0.89
Naive Bayes	0.87
CNN	0.92
LSTM	0.93
Random Forest (Our Choice)	0.98
CNN-LSTM Hybrid	0.95

```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       536
           1       0.98      0.96      0.97       318

    accuracy                           0.98       854
   macro avg       0.98      0.97      0.98       854
weighted avg       0.98      0.98      0.98       854
```

```
Accuracy: 0.977751756440281

Confusion Matrix:
[[529   7]
 [ 12 306]]
```

Fig. 4. Comprehensive Performance Metrics: Classification Report (top), Confusion Matrix, and Accuracy Score.

XI. ADVANTAGES

The AI-CloudAssist has great success rates with its classi-fication and organization systems, far beyond typical manual organization and simple heuristics based algorithms, due to its ensemble learning combined with Explainable AI. The Random Forest estimators automatically discover and weigh critical local features from the metadata, and SHAP provides the underlying patterns and interactions between metadata features that explain the reason why a specific entry was classified a specific way. RF-SHAP combines high accuracy prediction and humans readable explanations together for classification through diverse kinds of organization, reduces manual tagging and folder manipulation by intelligent and automatic classification and can be scaled to massive amount of data without decreasing the system efficiency. The system achieves better generalization and lowers overfitting.

XII. APPLICATIONS

- Cloud drive decluttering can be automated to under-stand user storage habits and generate context-aware, personalized folder structures.
- The system can be applied in Enterprise Data Gover-nance applications to automatically recognize and secure critical documents.
- It can be used for storage monitoring to analyze capacity trends, detect digital hoarding, and understand interactions.
- The proposed model is useful for smart retrieval in digital workspaces, minimizing the time users spend searching for misplaced entries.
- It can be implemented in automated archival systems to identify stale or unused items and move them to cold storage, saving significant enterprise costs.
- The system is also applicable in intelligent virtual assistants for improved and transparent human-computer interaction regarding local or cloud management.

XIII. FUTURE WORK

To make the system work even more efficiently we can use more sophisticated natural language processing models such as BERT and RoBERTa. These models are good at understanding the semantic context of any piece of text so our system will be able to categorize ambiguous files even more effectively. To train the model to better recognize complex behavioral patterns, it needs to be trained with a large, varied set of cross-platform data. It can predict real-time data from the live cloud-sync clients. A mobile application can be created to allow smart organization for hand-held devices. We can introduce multi-platform integration which allows the AI to organize for different services simultaneously. We may also look for other XAI architectures such as LIME or attention mechanisms to understand if we can provide even clearer explanations. The system may be integrated to the chatbot frameworks so that user simply has to say ”Clean up my folders” and the intelligent chatbot will perform the task and explain it to the user.

XIV. CONCLUSION

In this work, an intelligent organization system that uses Random Forest machine learning ensemble models with a SHAP based Explainable AI system is designed. In the pre-processing stage, metadata sanitization, string stripping, la-bel encoding and scaling of features is performed. Standard machine learning models like Logistic Regression and Single Decision Tree models were used as baseline models and the ensemble models are trained to produce stable and performant results. It was found that Random Forest performs much better than standard machine learning models. It was also found that by pairing feature extraction and interpretation using game theory on the RF-SHAP based model, an accuracy of 98 percentage was achieved. The proposed system performs well for real-life applications like a personal clutter clearing, data management for enterprises and a virtual intelligent assistant.

REFERENCES

1. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
2. S. M. Lundberg and S. I. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems (NIPS), 2017.
3. D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. W. O'Toole, "Semantic File Systems," Proc. of the 13th ACM Symposium on Operating Systems Principles (SOSP), 1991.
4. M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
5. M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
6. S. M. Lundberg et al., "From Local Explanations to Global Understand-ing with Explainable AI for Trees," Nature Machine Intelligence, vol. 2, pp. 56–67, 2020.
7. J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann, 1993.
8. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," Na-tional Institute of Standards and Technology (NIST), Special Publication 800-145, 2011.
9. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
10. A. Holzinger, "Explainable AI (XAI)," Informatik-Spektrum, vol. 41, no. 2, pp. 138–143, 2018.
11. C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed., 2020.
12. Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
13. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
14. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA: MIT Press, 2016.
15. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL-HLT, 2019.
16. Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019.
17. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference, 2016.
18. S. Hua et al., "Cloud Storage Management based on Machine Learning," IEEE International Conference on Cloud Computing, 2018.
19. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," ACM Computing Surveys (CSUR), vol. 51, no. 5, pp. 1–42, 2018.
20. F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Inter-pretable Machine Learning," arXiv preprint arXiv:1702.08608, 2017.
21. A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017.
22. M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016.
23. M. Zaharia et al., "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56–65, 2016.
24. B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," Fifth International Joint Conference on INC, IMS and IDC, 2009.
25. J. Ousterhout, H. Da Costa, D. Harrison, J. Kunze, M. Kupfer, and J. Thompson, "A Trace-Driven Analysis of the UNIX 4.2 BSD File System," ACM SIGOPS

Operating Systems Review, vol. 19, no. 5, pp. 15–24, 1985.