Malicious PDF Detection Using Autoencoders Trained on Metadata

Maria Diaz Alba
MS in Applied Artificial Intelligence IoT
Florida International University

***Abstract*—This paper explores the use of deep autoencoders for anomaly detection in PDF documents, with a focus on identifying malicious files through metadata analysis. The study investigates how reconstruction error can serve as a signal for anomalous behavior when the model is trained exclusively on benign samples. In addition, it examines the role of threshold selection in distinguishing between normal and suspicious documents. The approach is motivated by the need for lightweight, interpretable solutions to detect evasive threats in document-based attack scenarios.**

***Index Terms*—Autoencoder, anomaly detection, PDF malware detection, reconstruction error, metadata, threshold**

I. Motivation

Document-based cyberattacks, such as PDF exploits, are an increasing threat to organizational security. Traditional signature-based detection methods often fail to generalize to novel or evasive threats. This motivates the use of anomaly detection methods that learn benign patterns and detect deviations without relying on labeled malicious examples.

Autoencoders are well-suited for this task due to their ability to reconstruct known input distributions and highlight unusual deviations through elevated reconstruction error. This is especially relevant for metadata analysis, where threats are not easily detectable through fixed rules or manual inspection. Deep learning enables the discovery of latent patterns in structured data, offering a robust mechanism for identifying hidden anomalies in PDF metadata.

II. Literature Review

Anomaly detection using autoencoders has shown promising results in various forensic and cybersecurity applications. In [1], deep autoencoders are used to analyze forensic timelines and detect cyberattacks by identifying unusual patterns in log data. This approach reduces the need for exhaustive manual log inspection and enables earlier intervention in incident response.

Similarly, in the field of audio forensics, [2] introduces a benchmark dataset to detect rare and contextually inconsistent events, such as gunshots on a beach or screams in an office environment. Autoencoders are employed to model normal environmental audio patterns and detect anomalies that deviate from expected acoustic profiles.

In the domain of document security, the authors of [3] provide a comprehensive survey of malicious PDF detection techniques, emphasizing the effectiveness of static analysis methods based on metadata. Another structural approach is presented in [4], which models the hierarchical document composition of PDFs to detect deviations caused by embedded malicious objects.

The present work also aligns with the broader exploration of autoencoders applied to structured tabular data. As reviewed in [6], autoencoders are capable of learning compact representations of multivariate numerical data and detecting outliers based on reconstruction error. This reinforces their applicability in scenarios where feature relationships are complex but well-structured.

Finally, design guidelines and training strategies for autoencoders in anomaly detection tasks are extensively discussed in [7], with a focus on real-world use cases such as fraud detection. This reference was instrumental in guiding the architectural decisions and pre-processing pipeline of the current study, particularly regarding training exclusively on normal examples and evaluating performance in a semi-supervised context.

III. Dataset Information

A. Source

The dataset used in this project is the *CIC-Evasive-PDFMal2022* corpus, made publicly available by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick [8]. It is designed to support research in malicious document detection and contains both benign and evasive malicious PDF samples. The dataset provides not only the raw PDF files but also a preprocessed set of extracted metadata features to facilitate rapid experimentation and model development.

The metadata were extracted from over 10,000 PDF documents using a custom feature extraction pipeline designed to capture characteristics relevant to structural and behavioral analysis. These include 33 features such as file size, presence of JavaScript, object counts, use of encryption, and various embedded content indicators. Each record in the dataset corresponds to a single PDF file and includes a set of numerical and categorical features describing its composition, along with a target label indicating whether the file is `Benign` or `Malicious`. An overview of the features and their data types can be seen in Table I.

B. Class Distribution and Learning Paradigm

Before preprocessing, the dataset consisted of 4468 benign and 5555 malicious samples. The presence of the `class` label enables a semi-supervised evaluation framework, where

TABLE I: Feature Names and Data Types in the Dataset

Feature	Data Type
FileName	category
PdfSize	float32
MetadataSize	float32
Pages	float32
XrefLength	float32
TitleCharacters	float32
isEncrypted	float32
EmbeddedFiles	float32
Images	category
Text	category
Header	category
Obj	category
Endobj	category
Stream	float32
Endstream	category
Xref	category
Trailer	float32
StartXref	category
PageNo	category
Encrypt	float32
ObjStm	float32
JS	category
Javascript	category
AA	category
OpenAction	category
Acroform	category
JBIG2Decode	category
RichMedia	category
Launch	category
EmbeddedFile	category
XFA	category
Colors	float32
Class	category

labels are used only during testing. This setup reflects realistic cybersecurity conditions in which malicious labels may not be available at training time. To simulate this, the autoencoder is trained exclusively on the benign subset, using an unsupervised learning paradigm. The model is then evaluated on both benign and malicious files by measuring reconstruction error and comparing predictions against the true labels. This hybrid semi-supervised evaluation strategy allows for assessing the model's ability to detect anomalies without having seen any malicious examples during training.

C. Security Measures

Given the potential risk of working with malicious documents, a security protocol was followed throughout the project. A dedicated and isolated virtual machine (VM) running Ubuntu was created for all data-related tasks. Internet access was enabled only when necessary to download the dataset and Python dependencies. Key VirtualBox features were disabled, including clipboard sharing, drag-and-drop, and shared folders. Snapshots were taken prior to any potentially risky operations.

Importantly, no raw PDF files were opened or executed at any point. The work was conducted exclusively on the structured metadata provided by the CIC-Evasive-PDFMal2022 dataset. This ensured that no direct interaction with malicious payloads occurred, thereby eliminating any real infection risk while preserving the value of the data for anomaly detection research.

IV. Feature Processing and Engineering

A. Feature Cleaning

Categorical columns were parsed to extract numeric values. Entries such as "X(Y)" were converted to "X", and any non-numeric anomalies were coerced to NaN.

Additionally, three columns were removed: `FileName`, `Text`, and `Header`. The `FileName` field contains hash-like identifiers that do not contribute meaningful information for modeling. The `Text` and `Header` columns were removed due to their low variability and limited semantic value in capturing anomalous patterns, as they consist of repeated or binary categorical values not well-suited for numerical encoding.

B. Encoding and Scaling

Remaining features were transformed into numeric types. Missing values (0.37% of rows) were dropped due to low impact. All values were scaled to [0, 1] using MinMaxScaler.

C. Class Removal During Training

The column `class` (Benign or Malicious) was excluded from training inputs but retained for post-evaluation.

D. Final Dataset Split

To support a semi-supervised learning approach, the dataset was split into training and testing sets with a careful design. From the set of benign samples, 80% were used to form the training set `X_benign_train`. The remaining 20% of benign samples (`X_benign_test`) were then combined with all available malicious samples (`X_malicious`) to construct the testing set `X_test`. A corresponding vector of true class labels (`y_test`) was also created to enable later evaluation of the model's performance.

This setup allows the autoencoder to learn only from benign examples during training, in line with an unsupervised anomaly detection paradigm. Evaluation on both benign and malicious data then reveals how well the model identifies deviations from learned normality.

The final data distribution after preprocessing and scaling was as follows:

- `X_train_scaled`: (3572, 29) – benign samples used for training
- `X_test_scaled`: (6414, 29) – test set including benign (unseen) and malicious samples
- `y_test`: (6414,) – true labels used for performance evaluation

V. AI Model Building

A. Model Architecture

The autoencoder was implemented in TensorFlow/Keras using the Model subclassing API. It consists of a symmetric feedforward architecture designed to learn compressed representations of benign PDFs based on extracted metadata. The structure is as follows:

- **Encoder:** Dense(64, relu) $\rightarrow$ Dropout(0.2) $\rightarrow$ Dense(32, relu) $\rightarrow$ Dense(7, relu)

- **Decoder:** Dense(32, relu) → Dense(64, relu) → Dense(29, sigmoid)

This bottleneck architecture compresses the 29-dimensional input into a 7-dimensional latent representation, enforcing the model to capture only the most relevant characteristics of benign behavior. The final sigmoid layer ensures that output values are scaled similarly to the input, which was normalized to the [0, 1] range.

B. Training Configuration

The model was trained using the Adam optimizer with a learning rate of 0.001. The main loss function was Mean Absolute Error (MAE), with Mean Squared Error (MSE) tracked as an additional performance metric. EarlyStopping was applied with a patience of 5 epochs and validation loss as the monitored metric. The training was conducted for a maximum of 50 epochs with a batch size of 64, using 10% of the training data for validation. The model converged after 11 epochs.

C. Model Observations

The training history, shown in Figures 1 and 2, reveals a rapid decline in both training and validation loss during the early epochs, stabilizing around epoch 10. Final values for MAE were approximately 0.009 for training and 0.0089 for validation. Similarly, MSE also decreased significantly and remained consistently low across both sets.

No signs of overfitting were observed throughout the training process. The parallel behavior of training and validation curves, along with their smooth convergence, indicate a well-regularized model. The use of dropout and a moderate bottleneck size likely contributed to this generalization capability. Overall, the training dynamics confirm that the model effectively learned to reconstruct benign samples with minimal error, setting a solid foundation for anomaly detection in the subsequent evaluation phase.

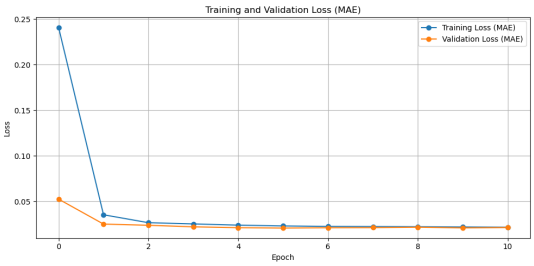

Fig. 1: Training and Validation Loss (MAE)

VI. Evaluating the Results

If a PDF is benign, its structure should be well reconstructed by the model, resulting in a low reconstruction error. In contrast, if the PDF is malicious, its internal structure and feature patterns will deviate from those learned during training, causing the model to fail in accurately reconstructing the input. This leads to a higher reconstruction error, which can be leveraged as a signal to flag the instance as anomalous.

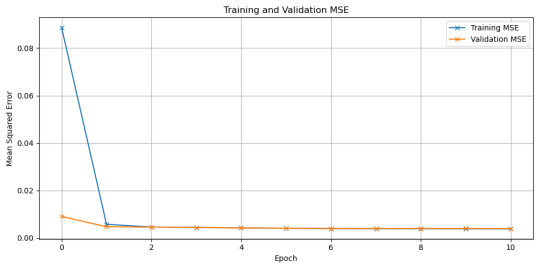

Fig. 2: Training and Validation MSE

A. Error Distribution

Reconstruction errors were computed on all test samples. Most benign files had low reconstruction errors (median: 0.012), while outliers with errors as high as 11.5 were observed, indicating anomalous structure. However, the overall distribution revealed a significant overlap between benign and malicious examples, suggesting limited discriminative power of the current model.

To better understand this phenomenon, Figures 3 and 4 present separate scatter plots of the reconstruction errors for benign and malicious samples. While some malicious PDFs exhibit clearly higher errors—validating the model's anomaly detection capability—many are reconstructed with low error values. This overlap suggests the autoencoder struggles to generalize structural deviations from metadata alone. As a result, a substantial number of malicious samples are likely to be misclassified as benign, leading to a high false negative rate.

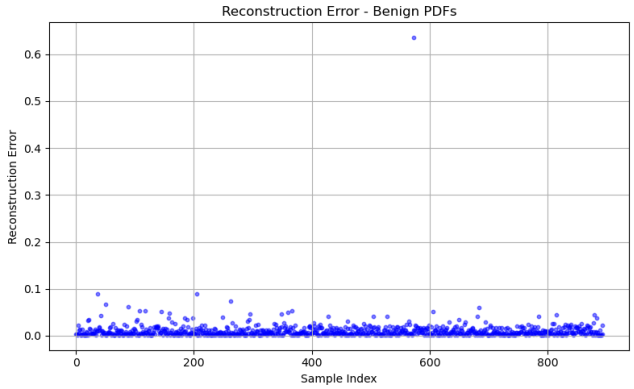

Fig. 3: Scatter Plots of Reconstruction Error by Class (Benign)

B. Threshold Optimization

To convert reconstruction error into class predictions, a decision threshold must be set. This threshold defines the cutoff point beyond which a sample is considered anomalous. As the reconstruction error distribution for benign and malicious PDFs shows overlap, choosing a proper threshold becomes essential to balance between false positives and false negatives.

We evaluated thresholds in the range [0.001, 0.2] and computed classification metrics for each. The optimal threshold

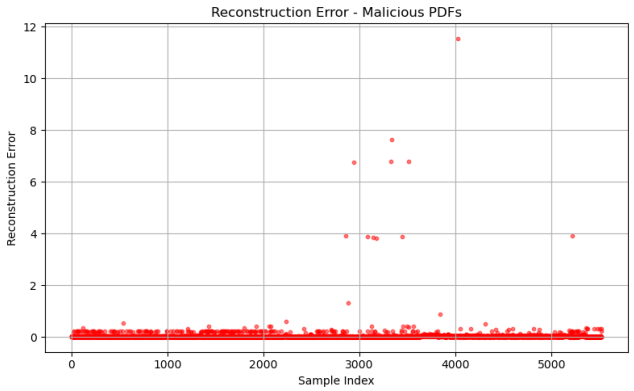

Fig. 4: Scatter Plots of Reconstruction Error by Class (Malicious)

was selected based on the maximum F1-score, which balances precision and recall. The best performance was obtained at a threshold of **0.0100**, which enabled a clearer separation between the two classes despite the earlier observed overlap in error distribution.

The precision-recall curve at Figure 5 confirms the expected trade-off between precision and recall when adjusting the reconstruction error threshold.

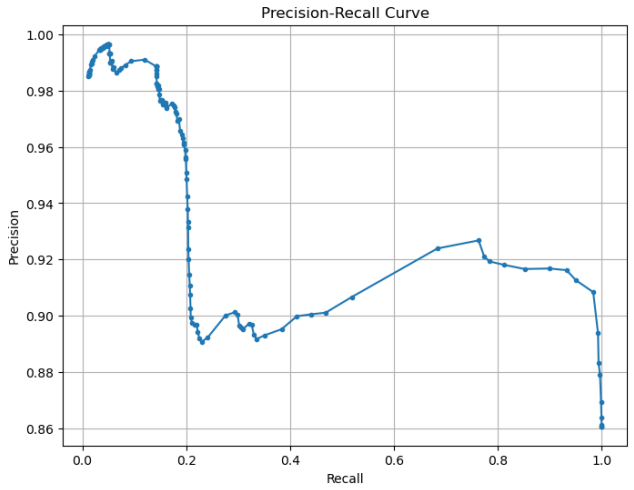

Fig. 5: Precision-Recall Curve Across Thresholds

This step demonstrates that although the autoencoder cannot perfectly separate benign and malicious examples, a careful threshold selection process can help recover meaningful distinctions and improve classification performance.

C. Confusion Matrix and Classification Report

At the chosen threshold of 0.0100, the classification metrics were as follows:

- **Precision:** 90.8%
- **Recall:** 98.4%
- **F1-score:** 94.4%

The updated confusion matrix indicates a trade-off favoring high recall: the model now classifies nearly all malicious samples correctly but at the cost of an increase in false positives. This behavior reflects the overlap in reconstruction errors and demonstrates the effect of shifting the threshold to favor minimizing false negatives.

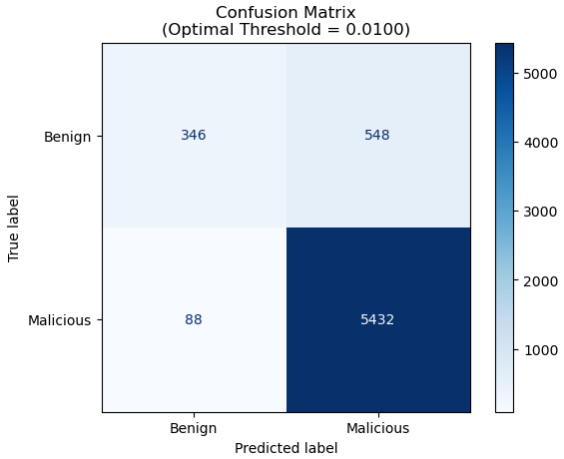

Fig. 6: Confusion Matrix (Threshold = 0.0100)

VII. Conclusion

This study investigated the use of deep autoencoders for anomaly detection in PDF metadata, with the goal of identifying malicious documents. The model was trained on benign samples using an unsupervised learning approach and evaluated on both benign and malicious samples.

Initial analysis of reconstruction error distributions revealed considerable overlap between the two classes, highlighting that the autoencoder alone does not inherently separate them well. However, through threshold optimization, we were able to extract useful distinctions and achieve meaningful classification performance, with a precision of 90.8%, recall of 98.4%, and F1-score of 94.4%.

Despite these improvements, the model often reconstructs malicious PDFs too well, leading to some false negatives when a threshold is not appropriately calibrated. These limitations suggest that either the current feature set lacks sufficient discriminatory power, or the model complexity is inadequate for capturing subtle patterns in the metadata. Nevertheless, the thresholding strategy proved effective at amplifying the distinction between benign and malicious documents.

A. Limitations and Challenges

Some sophisticated malicious PDFs may resemble benign ones closely, reducing the model's ability to distinguish them based solely on metadata. Additionally, inconsistencies or inaccuracies during metadata extraction may propagate noise into the learning process, adversely affecting performance. The overlap in reconstruction error distributions further confirms the challenge of relying on compressed latent representations to isolate anomalies in such structurally similar files. These

limitations emphasize the importance of complementary analysis techniques or hybrid approaches.

B. Applications and Implications

The autoencoder-based anomaly detection framework presents a lightweight and interpretable tool for enhancing malware detection workflows. Its unsupervised nature makes it suitable for real-world scenarios where labeled malicious data is scarce or unreliable. By flagging documents that deviate from learned benign behavior, the system can act as a first-pass filter in secure document handling environments such as email gateways, cloud storage, or digital forensics pipelines.

C. Future Work

Future research will focus on improving the discriminatory power of the input features. This could involve incorporating domain-specific transformations, richer contextual metadata, or applying hybrid models that combine deep learning with traditional static analysis. Additionally, exploring alternative autoencoder variants, such as variational or sparse architectures, and reducing the bottleneck dimension further may improve anomaly detection granularity. Other directions include leveraging attention-based encoding mechanisms, integrating time-aware features from document activity logs, and testing real-time deployment in operational PDF scanning systems. These directions aim to enhance the practicality and reliability of anomaly detection systems in real-world PDF-based attack scenarios.

REFERENCES

[1] H. Studiawan and F. Sohel, "Anomaly detection in a forensic timeline with deep autoencoders," *Journal of Information Security and Applications*, vol. 63, p. 103002, 2021. [Online]. Available: https://doi.org/10.1016/j.jisa.2021.103002

[2] A. Abbasi, A. R. Rehman Javed, A. Yasin, Z. Jalil, N. Kryvinska, and U. Tariq, "A large-scale benchmark dataset for anomaly detection and rare event classification for audio forensics," *IEEE Access*, vol. 10, pp. 38885–38894, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3166602

[3] N. Nissim, A. Cohen, C. Glezer, and Y. Elovici, "Detection of malicious PDF files and directions for enhancements: A state-of-the-art survey," *Computers & Security*, vol. 48, pp. 246–266, 2015. [Online]. Available: https://doi.org/10.1016/j.cose.2014.10.014

[4] N. Srndic and P. Laskov, "Detection of malicious PDF files based on hierarchical document structure," in *Proc. Network and Distributed System Security Symposium (NDSS)*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:1458246

[5] G. Elingiusti, G. D'Angelo, and S. Zanero, "PDF-malware detection: a survey and taxonomy of current techniques," *Journal of Computer Virology and Hacking Techniques*, vol. 14, pp. 1–24, 2018.

[6] A. Altaibek, I. Tokhtakhunov, M. Nurtas, D. Kozhamzharova, and M. Aitimov, "The efficacy of autoencoders in the utilization of tabular data for classification tasks," *Procedia Computer Science*, vol. 238, pp. 492–502, 2024. [Online]. Available: https://doi.org/10.1016/j.procs.2024.06.052

[7] Machine Learning Group (Université Libre de Bruxelles - ULB), "Autoencoders," in *Chapter 7: Deep Learning*, *Fraud Detection Handbook*, 2022. [Online]. Available: https://fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_7_DeepLearning/Autoencoders.html

[8] Canadian Institute for Cybersecurity, "CIC-Evasive-PDFMal2022 Dataset," University of New Brunswick, 2022. [Online]. Available: https://www.unb.ca/cic/datasets/pdfmal-2022.html