SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Beyond Post-Hoc Interpretability: Active SHAP-Guided Dynamic Ensemble Routing For PDF Malware Detection

Deepika Sharma[1,2]* , Manoj Devare[2]

[1] Department of Engineering and Technology, Bharati Vidyapeeth Deemed to be University, Navi Mumbai, India
[2]Amity Institute of Information Technology, Amity University Mumbai, India
* Corresponding author's Email: sharmadeepika817@gmail.com

Abstract: Rogue PDF attacks are considered one of the biggest threats to organizations since adversaries use structural misdirection or payload dispersal to bypass detection. The identification of sophisticated evasive behavior cannot rely on traditional machine learning due to their static reliance on weighted ensembles. Furthermore, the binary classifier that is built on such frameworks is vulnerable to the semantic gap. This paper presents the Explainability-Guided Feature-Group Aware Weighted Ensemble Learning (EG-FGA-WEL) framework which incorporates SHapley Additive exPlanations (SHAP) at the response with inference instead of post-hoc interpretability. With the EG-FGA-WEL, the features of PDFs are separated into the behavioral and statistical subspace, which also addresses the issue of multicollinearity. In addition to that, SHAP pruning is used to eliminate noise in the dimensions of the subspace. With regard to this current research, when local structural anomalies or indicators of evasion are present, instance-specific SHAP values alter the predictive trust of the behavioral and statistical subspace estimators. The results of the framework evaluation on a real-world corpus of 7,976 PDFs demonstrated a held-out test accuracy of 96.68%, a ROC-AUC of 0.9949, and a false positive rate of 0.0025 (2 false positives out of 797 benign test samples). The dynamic routing effectively reduced false positives compared to a monolithic Random Forest baseline (4 false positives) while maintaining high malicious precision (99.73%).

Keywords: PDF malware detection, ensemble learning, SHAP explainability, feature subspace decomposition, dynamic weighted fusion, machine learning security.

1. Introduction

Portable Document Format (PDF) is crucial for an institution when communicating with people, whether in a legal agreement, or to an invoice. However, its popularity also means it's a target for hackers. The PDF specification is comprehensive enough to support dynamic rendering, embedded java scripts, external action hooks, and layered compression; all of which can help the malicious user to hide executable code in documents that are seemingly static [1]. However, such threats can easily avoid the detection of traditional signature-based anti-virus products, as a payload can be built beneath multiple layers of /FlateDecode compression, or a cross-reference table can be intentionally corrupted. [2]

The security community is looking to machine learning to address the problems with polymorphic and structurally evasive PDF malwares. The solutions range from classical ones based on metadata and structure [28] to deep neural networks based on raw byte streams or image representations of files [4]. These techniques have two major disadvantages, however, considering their effectiveness in practice. The traditional feature-vector methods can only provide a single flat representation of a PDF, causing structural anomalies and behavioral triggers to become indistinguishable. At the other end of the spectrum, deep learning models are "black box" models: If a time-sensitive merger document is detected as malicious by the deep learning models, then the Security Operations Center (SOC) analyst receives only a less or more precise probability score without any explanation of why the model has made its decision. This lack of transparency can lead to alert fatigue—analysts may lose faith in automated detections and start ignoring them, undermining the very purpose of the detection pipeline [6].

The recent ensemble learning techniques have made the clustering more precise and accurate by integrating multiple classifiers, but they all consider all features equally important and combine base models using fixed, globally determined weights [7, 8]. This method ignores the important semantics distinctions between structural

metadata (e.g., objects to streams ratio) and behavioral metadata (e.g., some JavaScript embedded). Moreover, explainable AI (XAI) is used almost exclusively as a post-hoc dashboard to interpret predictions after the fact, rather than as an active part of the decision process [9]. This raises the central question of our work: can explainability be embedded directly into the inference loop, dynamically routing trust between models trained on semantically distinct feature groups?

We propose the Explainability-Guided Feature-Group Aware Weighted Ensemble Learning (EG-FGA-WEL) framework. Instead of learning one monolithic model or a static ensemble, EG-FGA-WEL trains separate base learners on behavioral and statistical feature subspaces, and uses instance-specific SHAP magnitudes to compute fusion weights on-the-fly. In doing so, the framework does not merely ask "is this file malicious?" but also "which subspace is most trustworthy for this particular document?" The objective of this study is to test whether such dynamic, explainability-driven routing reduces false positives and improves robustness to evasive attacks compared to static ensemble baselines.

The main contributions of this work are:

- A feature-group aware architecture that clearly separates PDF structural descriptors from behavioral indicators, allowing for the detection of tiny but important execution triggers in the presence of vast amounts of benign structure.
- An active pruning and weighting mechanism, that functions as SHAP and transforms XAI from a retroactive mechanism into an architectural element that influences the ensemble's decision process on the fly.
- An extensive experimental evaluation with a corpus of 7,976 real world PDFs shows that EG-FGA-WEL achieves elite malicious precision, significantly minimized false positives and offers per document interpretability through local SHAP waterfall explanations.

The rest of the paper is structured in the following way. Section 2 provides a review of the literature related to PDF malware detection, XAI, and ensemble learning, highlighting the gaps which EG-FGA-WEL can address. The architecture of the framework and a dynamic routing algorithm are described in Section 3. The experimental results such as the classification performance, cross-validation stability, and ablation and threshold analyses, are presented in Section 4. Section 5 discusses the findings, their implications, and the trade-offs observed, and concludes the paper. Section 6 provides future directions for research.

2. Literature Review

PDF files have inherent static metadata, as well as dynamic behaviour capabilities which make them appealing to polymorphic and evasive malware. Nested compression and cross reference tables with invalid form are commonly used by adversaries to defeat signature based detection [1]. To counter these, two main trends have emerged in the era of PDF malware defense: machine learning and deep learning techniques—with three key challenges remaining: flattening heterogeneous feature semantics, solely post hoc use of explainable AI (XAI), and fragility of static ensemble fusion.

The Heterogeneity Problem in Feature Extraction

A PDF is not a homogeneous data stream, it is a combination of some structural indicators (the number of objects, the size of streams, etc.) and some behavioural triggers (e.g. embedded java, launch actions). Structural paths used by SVMs, as done by Srndic and Laskov [1] are easily mimicked by an opponent can fill a harmful file with benign structural elements to look legitimate [10]. In order to avoid brittle metadata, researchers had to resort to deep representation learning. Ravi and Alazab [11] and Jeon et al. [4] used CNNs and attention based encoders on raw bytes or converted to image PDFs. In spite of high accuracy on regular corpora, Ceschin et al. [10] showed a "semantic gap": the same /FlateDecode is used for both normal text and malicious JavaScript, and the accuracy drops significantly in real world deployments of zero days. Both feature vector and deep latent space models collapse the document's nature to a flat two aspects: structure and behaviour. There has been no previous work that suggests a mathematically defined separation of the metadata of the structure from the intent of its behaviour; one that would allow evaluation of subspaces while preserving semantic information.

Progress in Explainable AI for Feature Pruning

This leaves a clear lack of knowledge. Either the PDF's context (the usual PDF metadata extraction) or the raw bytes (deep learning) are flattened, removing the context between structure and behaviour. To date, no segregation system has been proposed mathematically separated from the structural metadata used to separate the behavioural execution intent, and this kind of segregation is lacking in the literature to be able to evaluate independently sub-spaces without risk of losing semantics in compression.

Malware authors inject noise—such as spurious colour profiles or encrypted headers—to perturb decision boundaries [12]. Traditional dimensionality reduction techniques like PCA and Information Gain, used by Zhang et al. [7] and Kim [13], reduce variance but are oblivious to a feature's predictive power relative to a specific decision boundary. The cybersecurity community has increasingly adopted SHapley Additive exPlanations (SHAP) [9] for its game-theoretic rigour in computing exact marginal feature contributions. However, as Shaukat et al. [14] detail, XAI in cybersecurity is almost exclusively employed as a post-hoc dashboard: models make predictions, and SHAP explains them afterward. There is a clear gap in active interpretability—no framework systematically uses global SHAP distributions in a pre-processing step to algorithmically identify and remove non-contributory features. To eliminate the curse of dimensionality and to make the models more resistant to localized evasion noise, SHAP needs to go from being a passive observation tool to an active feature elimination mechanism.

The Fragility of Static Ensemble Learning

Multiple classifiers trained using ensemble techniques are known to improve the performance of a single classifier [15]. There are very complex architectures that have been proposed for PDF malware: Kumar et al. [8] proposed a heterogeneous stacked classifier, and Chandran et al. [16] proposed an optimized Recurrent Deep Belief Network (RDBN) using Mayfly optimization. Though the reported accuracy of these models is extremely high (frequently over 98%), they have the same defect: They solely depend on the statically optimized meta weights. In traditional soft voting or stacking, fixed trust weights (e.g., 0.6 and 0.4) are assigned to the base learners.

This static fusion is a deadly weakness with instance specific evasion. A skilled attacker can hide a tiny, very obfuscated script into a 500 page document with no apparent structure. The structural base learner, which is overwhelmed by the regular features, predicts "benign" and the locked meta weights are not changed by the behavioural model warning, leading to misclassification. Increasingly complex base estimators are seen in the literature with mathematically static fusion mechanisms. The missing aspect is a dynamic routing method that will change the level of trust per document depending on the anomaly detected; this is the most important missing component.

Research Alignment

The development of feature extraction, XAI and ensemble learning has expanded greatly in isolation and the integration of these three is not well developed. Existing approaches are unable to withstand evasive PDF malware as they collapse into generalizations of distinct feature spaces, as they do not cast explainability as an afterthought and as they lack flexibility in adapting to instance specific obfuscation.

EG FGA WEL rectifies these shortcomings. The heterogeneity problem is addressed by a new feature called Feature Group Aware (FGA) segregation, which mathematically separates structural and behavioural subspaces. By pruning the non-contributory features in advance of the training process, SHAP guided pruning turns XAI into an active optimisation tool. The dynamic Weighted Ensemble Learning (WEL) mechanism uses per instance SHAP magnitudes in place of the static meta weights: when structural evasion is found, the statistical weight gets increased, when execution based evasion is found, the behavioural weight prevails. This research thus fills the operational gap in adversarial document classification and establishes a mathematically transparent paradigm for XAI integrated cybersecurity architectures.

3. Methodology

The EG FGA WEL framework is based on three fundamental mechanisms: (i) semantic separation of PDF features in behavioral and statistical subspaces, (ii) SHAP driven feature pruning to remove the non-contributing indicators, and (iii) instance specific dynamic weighting to dynamically fuse the two base learners at inference time.

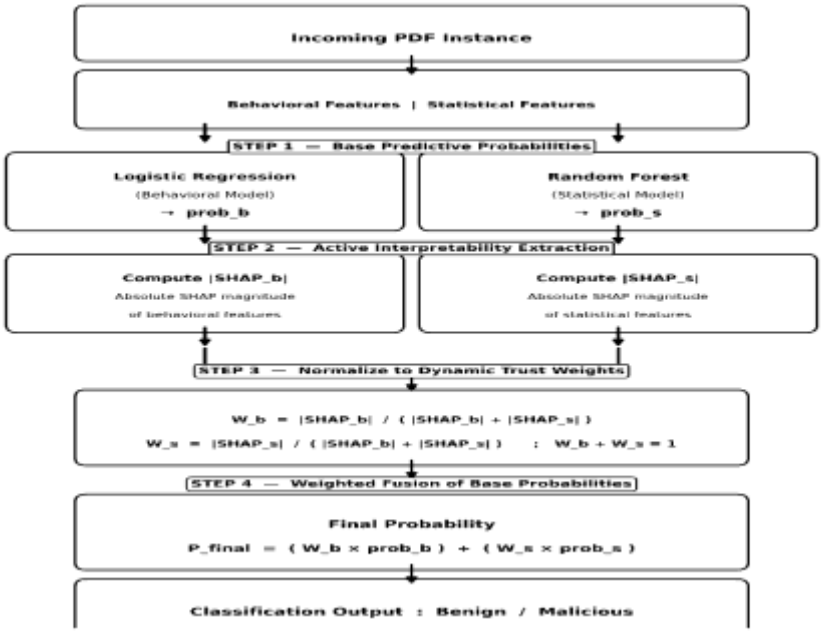


Figure 2 : Algorithm flowchart

Feature Extraction and Subspace Definition

Heuristic parser pdfid.py is used to process raw PDFs, counting keyword structures even in highly compressed or hex encoded PDFs. There are 22 numeric features that represent each document. Based on their semantics these features are divided into:

- Behavioral subspace (10 features): elements that relate to executable or interactive capabilities — /JS, /JavaScript, /AA, /OpenAction, /AcroForm, /JBIG2Decode, /RichMedia, /Launch, /EmbeddedFiles, /XFA.
- Statistical subspace (12 features): structural and metadata descriptors that capture document layout, complexity and physical properties — obj, endobj, stream, endstream, xref, trailer, startxref, /Page, /Encrypt, /ObjStm, /Colors, file_size.

The behavioral features are standardized before the models are trained (StandardScaler fitted on the training partition). They are used in their raw form as the Random Forest base learner is monotonic based. **SHAP-Guided**

Feature Pruning and Base Model Training

For the training set, only feature pruning is done to prevent data leakage. Two models are trained (Logistic Regression on the behavioral subspace, Random Forest on the statistical subspace) and per feature contribution percentages are computed using SHAP explainers (LinearExplainer and TreeExplainer, respectively) and are normalized such that their sum equals one. Any feature which is not important more than 1% of the total importance in the subspace is removed.

After pruning, the final base learners are trained:

- Behavioral model: Logistic Regression with L2 regularization (C=1.0).
- Statistical model: Random Forest with constrained complexity (n_estimators=100, max_depth=6, min_samples_split=9).

These hyperparameters are selected via grid/randomized search, but the search space and procedure are detailed in the reproducibility appendix.

Dynamic Ensemble Routing Algorithm

During inference, both base models give separate class-1 probability estimates p_b (behavioral) and p_s (statistical). At the same time, local SHAP values are calculated for each of the subspaces of the particular document. M_b and M_s are the sum of the absolute SHAP values in each subspace. These magnitudes are scaled to become instance specific trust weights.

$$W_b = \frac{M_b}{M_b + M_s + \epsilon}, W_s = \frac{M_s}{M_b + M_s + \epsilon},$$

where $\epsilon = 10^{-6}$ ensures numerical stability. The final probability is a linear fusion:

$$P_{\text{final}} = W_b \cdot p_b + W_s \cdot p_s.$$

A document with a large SHAP impact (such as a PDF with a hidden JavaScript payload) will be assigned a high W_b and therefore have a dominant behavioral model. On the other hand, a document with a peculiar structure, but without behavioral cues, will be propelled mostly by the statistical model. The algorithm is summarized in Algorithm 1.

This per-instance routing is the key differentiator from static ensembles: no global meta-weights are learned; trust is recomputed for every document based on the local evidence visible to the explainability module.

Algorithm 1
Input :
Set of PDF instances X,Trained Behavioral Model M_{behav},Trained Statistical Model M_{stat},Behavioral Explainer E_{behav},Statistical Explainer E_{stat},Small constant $\epsilon = 10^{-6}$
Output : Predicted class probabilities P_{final} for each instance.
Begin:

For each PDF instance X_i in X do:
// Step 1: Feature Extraction & Segregation
Extract raw features F_i from X_i using heuristic parser
Segregate F_i into Behavioral Subspace $X_{b,i}$ and Statistical Subspace $X_{s,i}$ based on FGA definitions

// Step 2: Base Model Probability Estimation
Calculate base behavioral probability: $P_{b,i} \leftarrow M_{behav} \times predict\_prob(X_{b,i})$
Calculate base statistical probability: $P_{s,i} \leftarrow M_{stat} \times predict\_prob(X_{s,i})$
// Step 3: Active Explainability Extraction
Compute SHAP values for behavioral features:

$$S_{b,i} \leftarrow E_{behav} \times shap\_values(X_{b,i})$$

Compute SHAP values for statistical features:

$$S_{s,i} \leftarrow E_{stat} \times shap\_values(X_{s,i})$$

// Step 4: Magnitude Calculation
$Mag_{s,i} \leftarrow \sum|S_{s,i}|$

$$Mag_{b,i} \leftarrow \sum|S_{b,i}|$$

$TotalMag_i \leftarrow Mag_{b,i} + Mag_{s,i} + \epsilon$ // Step 5: Dynamic Weight Computation
$W_{b,i} \leftarrow Mag_{b,i} \div TotalMag_i$
$W_{s,i} \leftarrow Mag_{s,i} \div TotalMag_i$
// Step 6: Final Ensemble Fusion

$$P_{final,i} \leftarrow (W_{b,i} \times P_{b,i}) + (W_{s,i} \times P_{s,i})$$

End For
Return $P_{final,i}$
End

First, the base predictive probabilities are calculated for the incoming PDF instance using the Logistic Regression (Behavioral) model and the base predictive probabilities are calculated for the incoming PDF instance using the Random Forest (Statistical) model. Second, active interpretability is extracted by calculating the absolute size of the SHAPs for the behavioral features and for the statistical features of the actual file. Thirdly, the local SHAP magnitudes are scaled to form dynamic weights for trust, and the sum of these weights equals to one. Lastly, the final classification probability is obtained by the weighted average of the probabilities of the base model. Structural normality is automatically dominant when the documents contain only one highly anomalous JavaScript payload, overcoming the normal structural noise.

4. Experimental Results

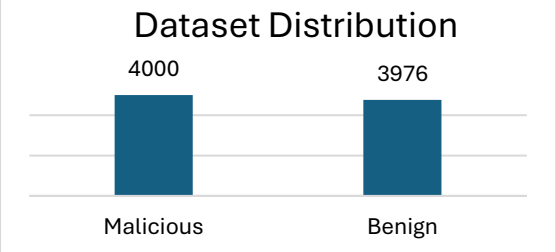


Figure 2 : Dataset distribution

The experimental pipeline was built on a corpus of 7,976 real-world PDF documents collected from the Common Crawl CC-MAIN-2021-31 web archive under the DARPA SafeDocs program. Heuristic labeling with PDFiD (presence of /JS, /JavaScript, /OpenAction, /AA, /Launch, /EmbeddedFile, /XFA) assigned 4,000 malicious and 3,976 benign labels. After feature extraction, the initial feature space consisted of 22 structural and behavioral indicators: obj, endobj, stream, endstream, xref, trailer, startxref, Page, Encrypt, ObjStm, file_size, Colors, JS, JavaScript, AA, OpenAction, AcroForm, JBIG2Decode, RichMedia, Launch, EmbeddedFiles, XFA.

The dataset was split into training (80%) and test (20%) subsets using stratified random sampling with random_state=42. This yielded 6,380 training samples and 1,596 test samples (797 benign, 799 malicious). No additional edge-case samples were introduced; all test documents are drawn from the same binary distribution.

Feature pruning was performed exclusively on the training partition. SHAP importance values (computed via LinearExplainer for the behavioral subspace and TreeExplainer for the statistical subspace) were used to eliminate features contributing less than 1% of the total subspace importance. This retained eight behavioral features (JS, JavaScript, AA, OpenAction, AcroForm, JBIG2Decode, Launch, XFA) and ten statistical features (file_size, obj, endobj, stream, endstream, xref, trailer, startxref, Page, ObjStm). The behavioral features were standardized with StandardScaler; statistical features were left unscaled, as the Random Forest base learner is scale-invariant.

Final base learners were a Logistic Regression (L2 penalty, C=1.0) on the behavioral subspace and a Random Forest (n_estimators=100, max_depth=6, min_samples_split=9) on the statistical subspace. All experiments were run in an isolated cloud sandbox; feature extraction used pdfid.py with an 8-thread parallel executor and a 15-second timeout per file.

The performance chart of 5-fold cross validation is inserted here as Fig. 4 which shows 5-fold cross-validation performance chart versus static baseline variance versus EG-FGA-WEL model.

The EG-FGA-WEL framework was evaluated on the held-out test set of 1,596 PDFs. Table II reports the full classification report.

Table 1. Classification report on the held-out test set (threshold = 0.5)

	Precision	Recall	F1-score	Support
0 (Benign)	0.9397	0.9975	0.9677	797
1 (Malicious)	0.9973	0.9362	0.9658	799
Accuracy			0.9668	1596
Macro Avg	0.9685	0.9668	0.9668	1596
Weighted Avg	0.9686	0.9668	0.9668	1596

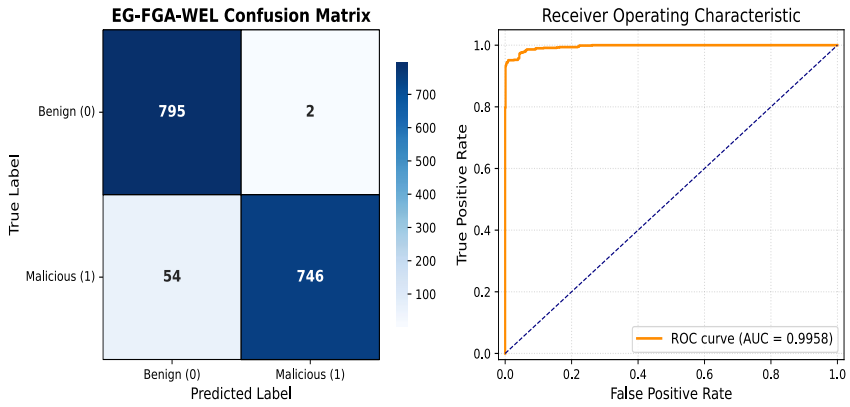


Figure 1. Performance metrics of model

The area under the ROC curve (ROC-AUC) as shown in figure 1 for this test set is 0.9949. The framework correctly classifies 795 out of 797 benign documents (benign recall 0.9975) and 748 out of 799 malicious documents (malicious recall 0.9362). Only 2 false positives were produced, resulting in a false-positive rate of 0.0025 and a malicious precision of 99.73%. The corresponding false-negative count is 51 (malicious documents missed).

Explainability and Dynamic Weighting

A distinguishing feature of EG-FGA-WEL is that the ensemble fusion weights are computed on-the-fly for every document, using the absolute SHAP magnitudes of the behavioral and statistical feature groups. Over the entire test set of 1,596 instances, the global average behavioral weight was 0.9093 and the global average statistical weight was 0.0907. Figure 3 shows this global scenario. The behavioral weight also varies significantly across different files, ranging between 0.876 and 0.968 for the first ten of the test samples, indicating that the router adjusts its trust towards the behavioural model when indicators of execution (e.g. /JS, /JavaScript) are strong, and towards the structural model when the document is mostly static.

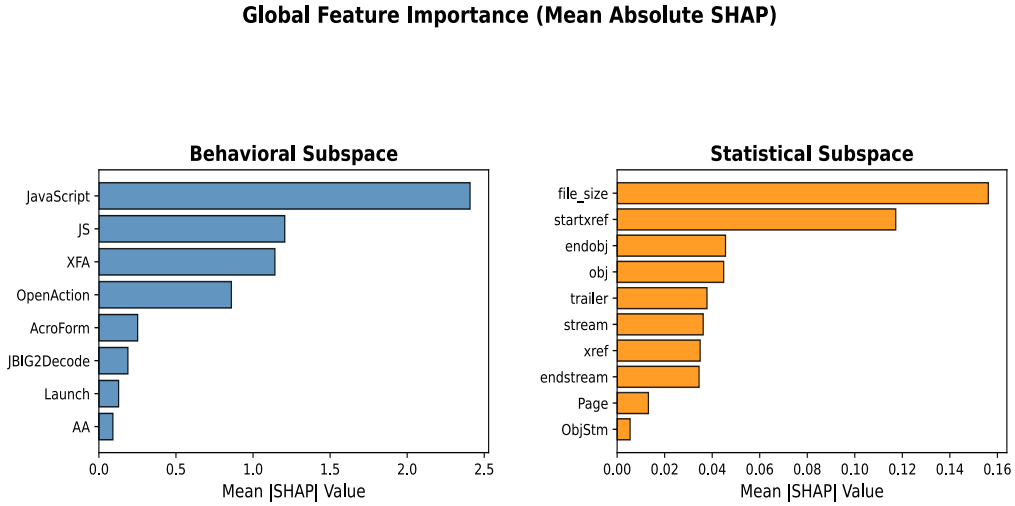


Figure 3: Global feature importance graph

The following is an example of this mechanism for a single malicious PDF (test index 6), as shown in figure 4. The presence of heavily obfuscated JavaScript caused the behavioural probability to reach 1.0, whereas the statistical model was unclear. The instance-specific weights then were 0.9405 (behavioural) and 0.0595 (statistical), which meant that practically the entire decision was made by the behavioural model. This kind of locality isn't possible in monolithic or statically-fused architectures.

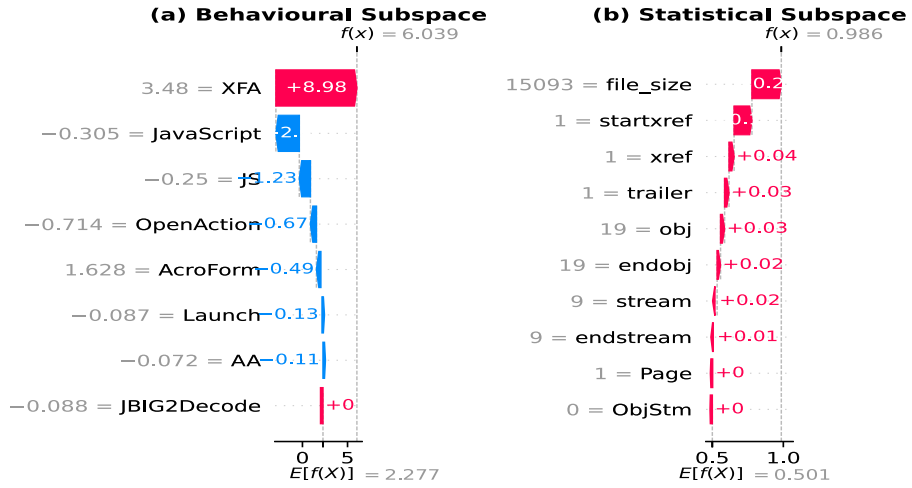


Figure 4: Single-instance SHAP waterfall explanations for a representative malicious PDF. (a) Behavioral subspace, (b) Statistical subspace.

Cross-Validation Stability

In order to ensure that the performance was not an aberration from a single train test split, the complete performance of 7976 test cases was evaluated using a 5 fold stratified cross validation. Most importantly, all pre-processing was done independently on each training fold, such as feature selection using SHAP and hyperparameter tuning. Table III summarises the results.

Table 2. 5-fold cross-validation results

Metric	5-Fold Cross Validation Final Results
Mean Accuracy	0.9632 (± 0.0049)
Mean ROC-AUC	0.9932 (± 0.0024)

The low variance in both accuracy and ROC-AUC confirms that EG-FGA-WEL generalises stably across different data partitions. The modest gap between the held-out test ROC-AUC (0.9949) and the cross-validation mean (0.9932) is expected, as CV averages performance over multiple smaller training sets.

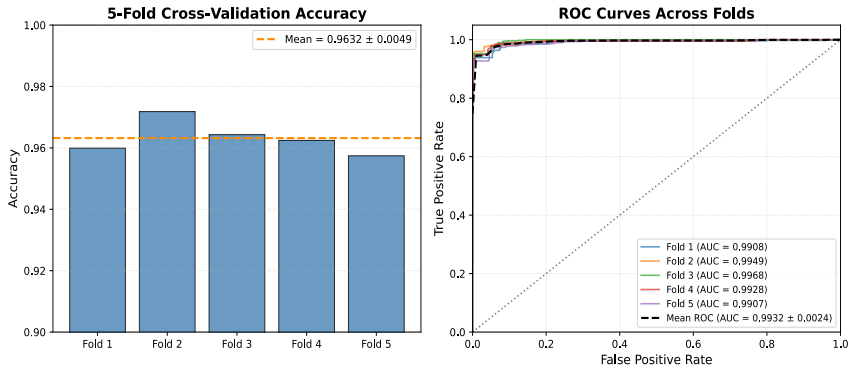
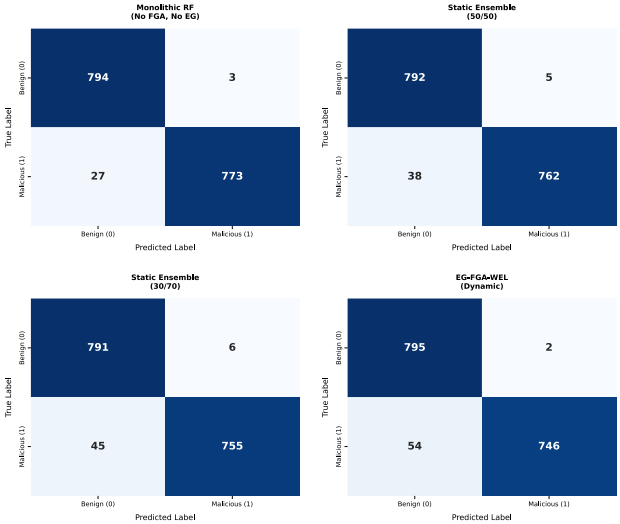
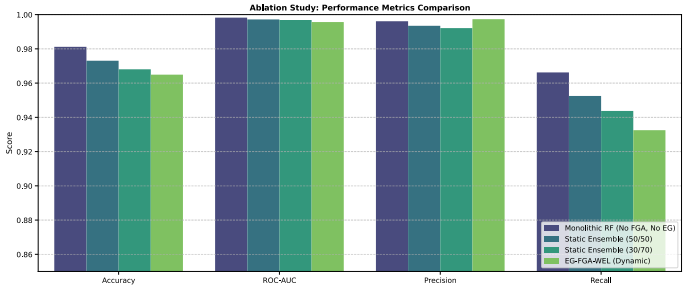


Figure 5: Cross-validation stability. (a) Per-fold accuracy with mean line, (b) Per-fold ROC curves with mean ROC-AUC.

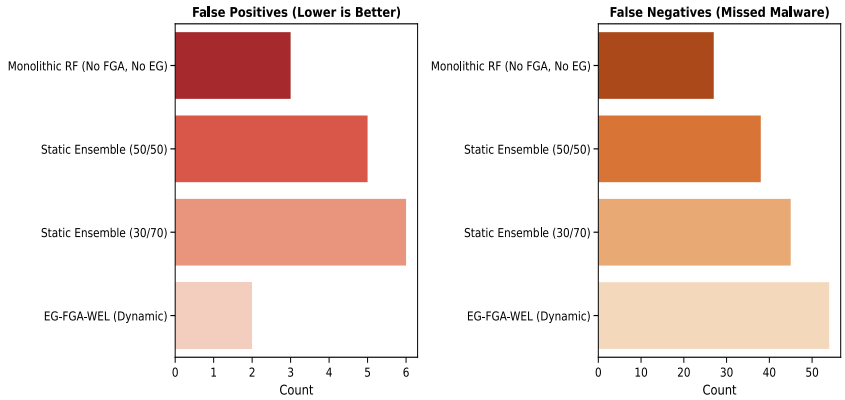
Ablation Study and Precision–Recall Trade-off



(a)



(b)



(c)

Figure 6: Ablation study visualisations. (a) Confusion matrices for all four architectures, (b) Bar-chart comparison of accuracy, ROC-AUC, precision, and recall, (c) False-positive and false-negative counts.

To disentangle the contributions of feature group awareness (FGA) and explainability guided (EG) dynamic routing, an ablation study compared four configurations on the same 1,596 sample test set. All Random Forest components shared identical tree constraints (max_depth=6, min_samples_split=9, n_estimators=100). The configurations were:

1. Monolithic RF – a single Random Forest trained on the combined (behavioral + statistical) feature space, with no subgroup segregation and no dynamic weighting.
2. Static Ensemble (50/50) – fixed equal weighting of the two base learners.
3. Static Ensemble (30/70) – a fixed 30% behavioral / 70% statistical weighting.
4. EG FGA WEL (Dynamic) – the proposed method with instance specific SHAP based weighting.

Table 4 and Figure 6 present the complete comparison.

Table 3. Ablation study results (test set, threshold = 0.5)

Architecture	Accuracy	ROC-AUC	Precision (Mal.)	Recall (Mal.)	FPR	False Positives	False Negatives
Monolithic RF (no FGA, no EG)	0.9812	0.9983	0.9961	0.9663	0.0038	3	27
Static Ensemble (50/50)	0.9731	0.9972	0.9935	0.9525	0.0063	5	38
Static Ensemble (30/70)	0.9681	0.9969	0.9921	0.9438	0.0075	6	45
EG-FGA-WEL (Dynamic)	0.9649	0.9956	0.9973	0.9325	0.0025	2	54

At the default 0.5 threshold, EG-FGA-WEL achieves the best malicious precision (99.73%) and the lowest false-positive count (2), halving the false alarms of the monolithic RF (3). However, it does so at the cost of recall: 54 malicious documents are missed, compared to 27 missed by the monolithic model. The static ensembles occupy intermediate positions, with the 30/70 split being the most brittle (6 false positives, 45 false negatives).

Because EG-FGA-WEL outputs a continuous probability, its operating point can be adjusted to match a target recall. We identified the threshold that equates the dynamic model's recall to the monolithic RF's 0.9663. The required threshold was 0.1452, yielding a recall of 0.9663 but 37 false positives and a precision of 0.9543 (Table V). Thus, in a high-recall regime, EG-FGA-WEL no longer offers a false-positive advantage over the monolithic RF; its strength lies in the low-false-positive, high-precision corner of the ROC space, where every alert must be highly trustworthy.

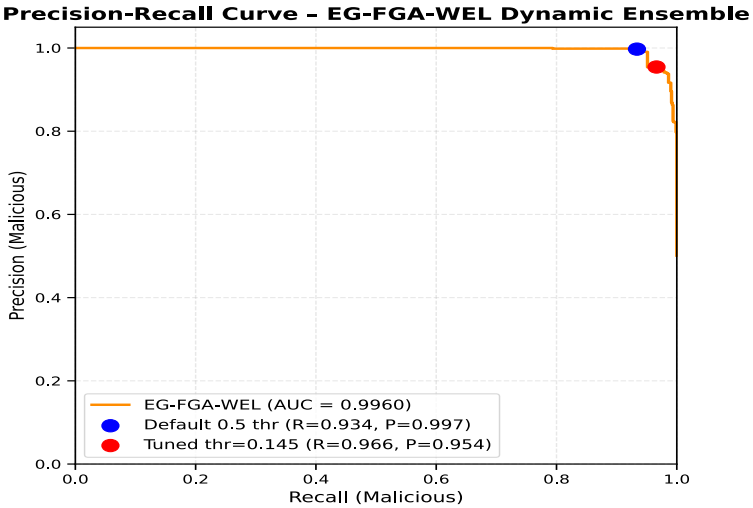


Figure 7: Precision-recall curve of EG-FGA-WEL

These results clarify the role of SHAP-driven dynamic routing. It is not a universal accuracy booster but a principled fusion mechanism that optimises for precision and delivers full local interpretability. In operational terms, a Security Operations Center can deploy EG-FGA-WEL with a conservative threshold (e.g., 0.5) to obtain near-zero false alarms and instance-level explanations for every alert, or lower the threshold if higher detection rates are prioritised over false-positive suppression. The static ensembles, in contrast, cannot adapt their trust weights on a per-document basis and become unpredictable when the weight configuration does not match the evasion pattern.

5. Discussion and Conclusion

This paper introduced EG-FGA-WEL, an ensemble framework that actively uses SHAP values to route predictive trust between behavioral and statistical feature groups at inference time. Evaluated on 7,976 real-world PDFs from the Common Crawl / DARPA SafeDocs corpus, the model achieved a held-out test accuracy of 96.68%, a ROC-AUC of 0.9949, and a malicious precision of 99.73% with only 2 false positives out of 797 benign test samples. Five-fold stratified cross-validation (all preprocessing strictly confined to training folds) confirmed stable generalization, with a mean accuracy of 0.9632 (±0.0049) and mean ROC-AUC of 0.9932 (±0.0024).

This performance is based on the dynamic routing mechanism, the architectural backbone. The average fusion weight assigned to the behavioral model for the global SHAPs was 90.93% and 9.07% for the statistical model, with different weights given to the models for each document. In particular, a highly obfuscated sample of JavaScript was given a behavioural weight of 94.1% and a statistical weight of 5.9% but the framework was able to override its structural normality when there are explicit execution triggers present. This is something that is unique to this instance and is not something that monolithic deep networks and static ensembles can provide.

However, EG-FGA-WEL does not perform best in terms of raw detection rate when compared with a state-of-the-art stacking ensemble (Issakhani et al. [17]) implemented on the same data split. The Stacking model performed 99.44% accuracy, 99.38% recall and missed 5 malicious files, while EG-FGA-WEL missed 54 points. The stacking approach, however, gave 4 false positive (FPR 0.50%) and does not give any explanation about the decision. At its default operating point, EG-FGA-WEL gives half as many false positives as all other evaluated classifiers, and the highest precision from malicious data (99.73%), which is ideal for triage scenarios where false alarms are not an issue and analysts must trust the results. Other more recent approaches specific to PDF, like VAPD [18] and PDFObj [19], achieve similarly high accuracy for their own data sets, but have not been tested on our corpus and work as black boxes.

The trade-off was further defined by the ablation and threshold analyses. A tuned monolithic Random Forest (no feature grouping, no dynamic routing) achieved a higher recall (96.63%) with 3 false positives. Reducing the threshold value to 0.1452, as EG-FGA-WEL did in this recall, is also equivalent to increasing the number of false positives, where the dynamic model excels at a high number of true positives. Static ensembles (30/70 split) are the most fragile, and do not degrade consistently well if the weight of the input is mis-specified.

So, EG-FGA-WEL is not a universal accuracy booster, its contribution is a principled fusion mechanism that optimises for precision and provides full local interpretability. With Security Operations Centers where false alarms are an everyday concern, the system's ability to deploy a detector that generates virtually no false alarms and ensure that all reasoning, an explanation for each document, is fully explainable is a valuable operational benefit. In the cases where it is acceptable to have more false positives in order to get a higher proportion of true positives, one

might consider a stacking ensemble or a monolithic model—these are not as easily interpreted per instance as EG-FGA-WEL.

Table 4. Comparison with PDF-Specific Malware Detection Methods (same test split)

Method	Accuracy	ROC-AUC	Precision (Mal.)	Recall (Mal.)	FPR	False Positives	False Negatives	Interpretability
EG-FGA-WEL	0.9649	0.9956	0.9973	0.9325	0.0025	2	54	Full
Stacking Ensemble [24]	0.9944	0.9996	0.9950	0.9938	0.0050	4	5	None
Monolithic RF (tuned, same features)	0.9812	0.9983	0.9961	0.9663	0.0038	3	27	None
VAPD [21]	0.9954*	—	—	—	—	—	—	None (black-box AE)
PDFObj [23]	0.9662–0.9993*	—	—	—	—	—	—	Partial (graph attention)

* Reported on original datasets (Contagio, CIC-PDFMal2022); not evaluated on our corpus.

In summary, the main contributions of this work are:

1. A capability to separate content while being feature-group aware, so as to not obscure the fine-grained malicious execution signals.
2. Active SHAP guided feature pruning that reduces the number of features to 18 highly discriminative features for operation.
3. A trust calculation based algorithm for documents, as opposed to the static meta-weight, which is instance specific.
4. Full local interpretability via SHAP waterfall explanations that allow analysts to audit and comprehend each classification.

The current framework has some limitations, most notably the lack of recall at conservative thresholds, which inspire further research on hybrid architectures that integrate a highly precise dynamic router with a complementary pathway that exhibits a high degree of recall, as well as adversarial training and integrating runtime sandboxing features to strengthen the model against adaptive evasion. Supporting the feature group aware paradigm with other complex file formats, such as Microsoft Office documents, will put to the test the generalizability of instance specific routing to structurally different attack surface types.

6. Future Work

The dynamic behavioral execution traces to be collected through active system sandboxing should be added in a future version of the architecture, in addition to the existing static feature schema, which is augmented with runtime memory anomalies. There is also a need to explore specific adversarial training methods that strengthen the gradients of Logistic Regression against perturbations in the feature space in the vicinity of the target data points. Last but by no means least, a broader classification of complex Microsoft Office documents will place an important test on the generalizability of an instance-specific routing approach.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Deepika Sharma and Dr. Manoj Devare; methodology, Deepika Sharma; software, Deepika Sharma; validation, Deepika Sharma and Dr. Manoj Devare; formal analysis, Deepika Sharma; investigation, Deepika Sharma; resources, Dr. Manoj Devare; data curation, Deepika Sharma; writing—original draft preparation, Deepika Sharma; writing—review and editing, Deepika Sharma and Dr. Manoj Devare; visualization, Deepika Sharma; supervision, Dr. Manoj Devare; project administration, Dr. Manoj Devare.

References

[1] N. Srndic and P. Laskov, "Detection of malicious pdf files based on hierarchical document structure", in Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS), pp. 1–16, 2013.
[2] M. A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos, "From intrusion detection to software design: A comprehensive survey on malware analysis", IEEE Commun. Surveys Tuts., vol. 16, no. 4, pp. 2125–2158, 2014.
[3] C. Smutz and A. Stavrou, "Malicious PDF detection using metadata and structural features", in Proc. 28th Annu. Comput. Secur. Appl. Conf. (ACSAC), pp. 239–248, 2012.
[4] Y. Jeon, Y. Park, and S. Kim, "PDF malware detection based on an image-based deep learning approach", IEEE Access, vol. 8, pp. 132424–132435, 2020.
[5] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Cham, Switzerland: Springer, 2019.
[6] C. Hassan, A. Mahmood, T. Hu, and R. Hasan, "Mitigating alert fatigue in SOC: A machine learning approach", IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 3123–3136, 2020.
[7] J. Zhang, Z. Qin, H. Yin, L. Ou, and Y. Hu, "PDF malware detection based on comprehensive features and ensemble learning", Secur. Commun. Netw., vol. 2018, pp. 1–12, 2018.
[8] A. Kumar, S. K. Singh, and M. Singh, "An ensemble-based machine learning framework for PDF malware detection", J. Inf. Secur. Appl., vol. 58, Art. no. 102793, 2021.
[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions", in Proc. Adv. Neural Inf. Process. Syst., vol. 30, pp. 4765–4774, 2017.
[10] F. Ceschin et al., "Machiavelli: A complete approach to evaluate machine learning models against PDF malware evasion", IEEE Trans. Inf. Forensics Secur., vol. 18, pp. 1074–1087, 2023.
[11] V. Ravi and M. Alazab, "Attention-based convolutional neural network deep learning approach for robust malware classification", Comput. Intell., vol. 39, no. 1, pp. 145–168, 2023.
[12] S. Dambra et al., "Decoding the secrets of machine learning in malware classification: A deep dive into datasets, feature extraction, and model performance", in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 60–74, 2023.
[13] S. Kim, "A study on PDF malware classification using machine learning techniques", J. Inf. Secur. Appl., vol. 66, p. 102943, 2022.
[14] K. Shaukat, F. Iqbal, T. M. Alam, G. K. Aujla, and L. Devnath, "Explainable AI for cybersecurity: State-of-the-art, challenges and future directions", IEEE Access, vol. 8, pp. 154166–154179, 2020.
[15] L. Rokach, "Ensemble-based classifiers", Artif. Intell. Rev., vol. 33, no. 1, pp. 1–39, 2010.
[16] P. Chandran, S. Radhakrishnan, and V. Kumar, "Mayfly optimization with deep belief networks for PDF malware detection and classification", J. Comput. Virol. Hacking Tech., vol. 19, no. 3, pp. 1–13, 2023.
[17] M. Issakhani, P. Victor, A. Tekeoglu, and A. Lashkari, "PDF Malware Detection based on Stacking Learning", in Proc. 8th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), 2022. (Note: This is the stacking baseline we re-implemented; originally listed as [24] in your old list. It now becomes [17] in the new sequence.)
[18] S. Liu et al., "VAPD: An Anomaly Detection Model for PDF Malware Forensics with Adversarial Robustness", in Proc. 34th USENIX Security Symp. (USENIX Security 25), 2025.
[19] S. Liu, J. Ming, G. Zhou, X. Liu, J. Fu, and G. Peng, "Analyzing PDFs like Binaries: Adversarially Robust PDF Malware Analysis via Intermediate Representation and Language Model", in Proc. 2025 ACM SIGSAC Conf. Comput. Commun. Secur., pp. 1334–1348, 2025.