Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RANSOM-XAI: AN EXPLAINABLE STATIC-BEHAVIORAL FUSION FRAMEWORK FOR FAMILY-DISJOINT RANSOMWARE DETECTION

HANI AL-BALASMEH

College of Engineering, University of Technology Bahrain (UTB), Bahrain
E-mail: h.albalasmeh@utb.edu.bh

ABSTRACT

Ransomware detection remains difficult when previously unseen malware families resemble known threats and a classifier provides insufficient evidence for security decisions. This study proposes RANSOM-XAI, a static–behavioral fusion framework that separates Portable Executable attributes from sandbox behavior, calibrates each LightGBM branch, and combines the class probabilities using a logistic stacker. A family-disjoint research design evaluates transfer to four withheld ransomware families and four withheld other-malware families. On 21,752 samples, the family-disjoint test achieved 97.44% accuracy, 0.9716 macro-F1, 0.9606 Matthews correlation coefficient, and 0.9647 ransomware F1. Early feature fusion achieved a slightly higher macro-F1 (0.9735), whereas the proposed method retained interpretable branch evidence. SHAP masking gave a 5.54-fold greater ransomware-probability reduction than random masking; top-10 explanation stability was 0.9636. Requiring branch agreement and confidence of at least 0.85 allowed 66.24% of decisions to be automated at 98.82% accuracy. However, behavioral-branch failure reduced macro-F1 to 0.1539, and slight behavioral-feature noise caused substantial deterioration. The contribution is thus an auditable, family-disjoint decision framework combining calibrated fusion, quantitative explanation checks, and selective review rather than a claim of universal detection or the highest raw predictive score. Reliable runtime telemetry and external validation remain prerequisites for deployment.

Keywords: *Ransomware Detection, Explainable Machine Learning, Family-Disjoint Validation, Static–Behavioral Fusion, Selective Classification*.

1. INTRODUCTION

Ransomware converts endpoint compromise into an availability and extortion crisis by denying access to files or systems and pressuring victims to pay for restoration. Contemporary defensive guidance therefore treats detection, containment, recovery preparation, and evidence preservation as complementary controls rather than relying on a single signature mechanism [1]. Machine-learning research has broadened the available evidence from byte signatures to Portable Executable (PE) structure, runtime system events, file activity, registry operations, network communication, and process behavior [2]-[6].

Research motivation and importance. Encryption-oriented malware creates a decision problem rather than simply a labeling problem: an incorrect quarantine may disrupt essential services, while delayed escalation may permit data loss. The relevant unanswered question is whether an executable classifier can provide both defensible evidence and a reliable route to human review when encountering families not represented in fitting. Unlike a random split, a family-disjoint test challenges transfer beyond memorized family artifacts, and a confidence- and agreement-based policy explicitly limits autonomous intervention. This why-and-how framing defines the practical problem addressed here.

Two methodological weaknesses nevertheless limit the practical meaning of many published scores. First, random splitting can place samples from the same malware family, campaign, packer, or collection process in both training and test sets. A model may then learn family-specific artifacts instead of transferable ransomware behavior. Security-machine-learning research has repeatedly warned that duplicated or semantically related observations can create optimistic estimates that deteriorate under temporal, environmental, or family shift [7], [8]. Second, a detector that emits a class label without auditable evidence does not answer the operational question: why should an analyst quarantine this process, block this executable, or delay an automated response? Explainability is particularly important when false

positives can interrupt legitimate business activity and false negatives can permit destructive encryption [9]-[12].

Static and dynamic analysis offer different advantages. Static PE evidence is available before execution and is inexpensive to obtain, but obfuscation, packing, and compiler variation can weaken family transfer. Dynamic evidence captures what a program actually does, but it requires monitored execution and can be degraded by sandbox evasion, incomplete observation, or telemetry loss. Hybrid fusion is therefore attractive, yet a useful hybrid detector must reveal which branch supports a decision, report uncertainty, and remain explicit about the failure caused by missing behavioral evidence [5], [6], [13].

This work addresses these requirements through RANSOM-XAI, a calibrated late-stacking framework that keeps static and behavioral models separate until their probability outputs are fused. The separation preserves branch-specific explanations and enables a confidence-aware policy: a high-confidence decision is automated only when the two branches agree; otherwise, the sample is referred for analyst review. The approach is evaluated as a three-class task - benign, ransomware, and other malware - because distinguishing ransomware from general malware is more operationally meaningful than an easier benign-versus-ransomware binary problem.

Problem statement and proposed solution. Existing static, dynamic, and hybrid systems can report excellent benchmark scores, but family overlap between training and test samples, dependence on particular telemetry sources, and untested explanation reliability can make these scores operationally misleading [4]–[10]. RANSOM-XAI tests the hypothesis that branch-preserving, calibrated late fusion can deliver high discrimination on withheld families while offering explanations that survive perturbation and a conservative abstention rule. The method does not assume that explanatory visibility alone improves classification or that automation is intrinsically safe.

1.1 Research Questions and Hypotheses

The study asks three research questions. RQ1 examines whether static-behavioral fusion can maintain high ransomware discrimination when entire families are unseen during training. RQ2 examines whether branch-specific explanations are faithful, compact, and stable rather than merely visually plausible. RQ3 examines whether model agreement and calibrated confidence can support reliable selective automation while preserving an analyst-review path.

The corresponding hypotheses are: H1, family-disjoint macro-F1 and ransomware F1 both exceed 0.90; H2, masking the explanation-ranked features produces at least three times the probability reduction of random masking and top-10 bootstrap Jaccard stability exceeds 0.80; and H3, automated decisions exceed 0.98 accuracy while uncertain or disagreeing cases are withheld for review.

1.2 Contributions and Scope

The contributions are fourfold. First, a leakage-resistant family-disjoint protocol is defined for hybrid ransomware evaluation. Second, a calibrated stacking architecture preserves complementary PE and behavior evidence without collapsing explanation provenance. Third, explanation quality is audited through perturbation faithfulness, compactness, and bootstrap stability, rather than reporting SHAP plots alone. Fourth, an agreement- and confidence-aware decision policy converts probabilistic predictions into transparent automation and review outcomes.

The scope is deliberately bounded. The experiments use pre-extracted Windows PE and sandbox-behavior features; no live ransomware binaries are executed in this study. The work evaluates family transfer within the supplied collection and does not claim universal zero-day detection, resistance to adaptive attackers, or production safety of automated containment. These boundaries are retained throughout the analysis.

Conceptual foundation and research design. The study adopts comparative, quantitative, observational machine-learning research on an existing labeled feature table. The static branch supplies pre-execution structure; the dynamic branch supplies observed activity; late stacking integrates their calibrated probability evidence; and selective classification withholds automated action where branch agreement or confidence is insufficient. This combination links complementary evidence, out-of-family generalization, post-hoc explanation auditing, and risk-aware human oversight. It extends methodological concerns in security ML [7], empirical static and dynamic ransomware analyses [3]–[6], and reliability-oriented interpretability research [9]–[12], while testing them on the specified dataset rather than asserting cross-industry transfer.

2. RELATED WORK

2.1 Static Ransomware Detection

Static analysis extracts evidence without running the sample, including DOS and PE header values, import or section structure, entropy, byte sequences, strings, and image representations. Moreira et al. transformed PE-header evidence into images and applied an Xception network, demonstrating that header structure can discriminate ransomware at low runtime cost [3]. Other work has used PE-header learning for early or zero-day-oriented evaluation [14] and general static malware corpora such as EMBER [15]. Static models are attractive for pre-execution screening, but packers, metadata manipulation, toolchain similarity, and family-specific artifacts can impair transfer to unseen lineages.

2.2 Dynamic and Behavioral Detection

Dynamic analysis monitors execution-related evidence such as file-system operations, API calls, registry modifications, network requests, process creation, entropy changes, and system calls. XRan combines dynamic evidence with deep learning and post-hoc explanations [4]. XRGuard uses Event Tracing for Windows and model-agnostic explanation to expose behavioral indicators [5]. Recent work has also emphasized lightweight API-call ensembles [16] and image-based dynamic features [17]. Behavioral evidence is typically more semantically aligned with ransomware actions, but it is expensive, environment-sensitive, and vulnerable to observation gaps or deliberate dormancy.

2.3 Hybrid and Explainable Detection

Hybrid systems seek to combine pre-execution and runtime evidence. RansomTrack integrates Radare2-derived static features with Frida-based dynamic instrumentation and reports SHAP-based interpretation [6]. Broader malware studies also show that explanation quality must be evaluated carefully: correlated features can redistribute attribution, local explanations can be unstable, and visually appealing rankings do not establish causal relevance [9], [10], [18]. SHAP provides a consistent additive attribution framework [11], while calibration methods separate confidence quality from class accuracy [19], [20].

The gap addressed here is not another claim that hybrid learning always produces the highest accuracy. Instead, the study combines family-disjoint evaluation, calibrated branch-preserving fusion, quantitative explanation auditing, telemetry stress testing, and selective automation in one reproducible pipeline. Table 1 positions the contribution against representative work.

Table 1. Positioning against representative ransomware-detection research

Study	Evidence / method	XAI / evaluation	Limitation relative to this work
Moreira et al. [3]	PE-header images; Xception	Static performance	No behavioral branch or selective triage
XRan [4]	Dynamic analysis; CNN	LIME and SHAP	No family-disjoint static-behavior fusion
XRGuard [5]	ETW behavioral logs	Model-agnostic XAI	Dynamic-only evidence
RansomTrack [6]	Static + dynamic behavior	SHAP; hybrid data	Different runtime instrumentation and scope
Galli et al. [9]	Behavioral malware models	XAI analysis	General malware focus
RANSOM-XAI	Calibrated branch stacking	Family-disjoint, XAI audit, triage	Dataset-bound; behavior telemetry required

Synthesis of the research gap and explicit questions. Static PE-based detection is computationally convenient but can overfit toolchains and packer artifacts [3], [14], [15]; dynamic detection is closer to malicious behavior but depends on sandbox completeness [4], [5], [13], [16]; recent hybrid approaches combine the two but do not necessarily expose calibrated branch disagreement or measure explanation reliability under complete family exclusion [6]. The unresolved research questions are therefore: (RQ1) how much ransomware performance transfers to withheld families; (RQ2) whether reported explanations are influential and stable; and (RQ3) whether confidence plus branch agreement can limit automated errors. The prespecified H1–H3 in Section 1.1 operationalize these questions.

3. PROPOSED RANSOM-XAI FRAMEWORK

RANSOM-XAI processes a sample through two complementary branches. The static branch represents PE structure and security metadata available before execution. The behavioral branch

represents monitored registry, network, process, file, DLL, and API activity. Each branch produces calibrated class probabilities for benign, other malware, and ransomware. A logistic meta-model learns how to combine these outputs, while the branch outputs remain available for explanation and agreement testing. Figure 1 summarizes the complete training, inference, explanation, and triage path.

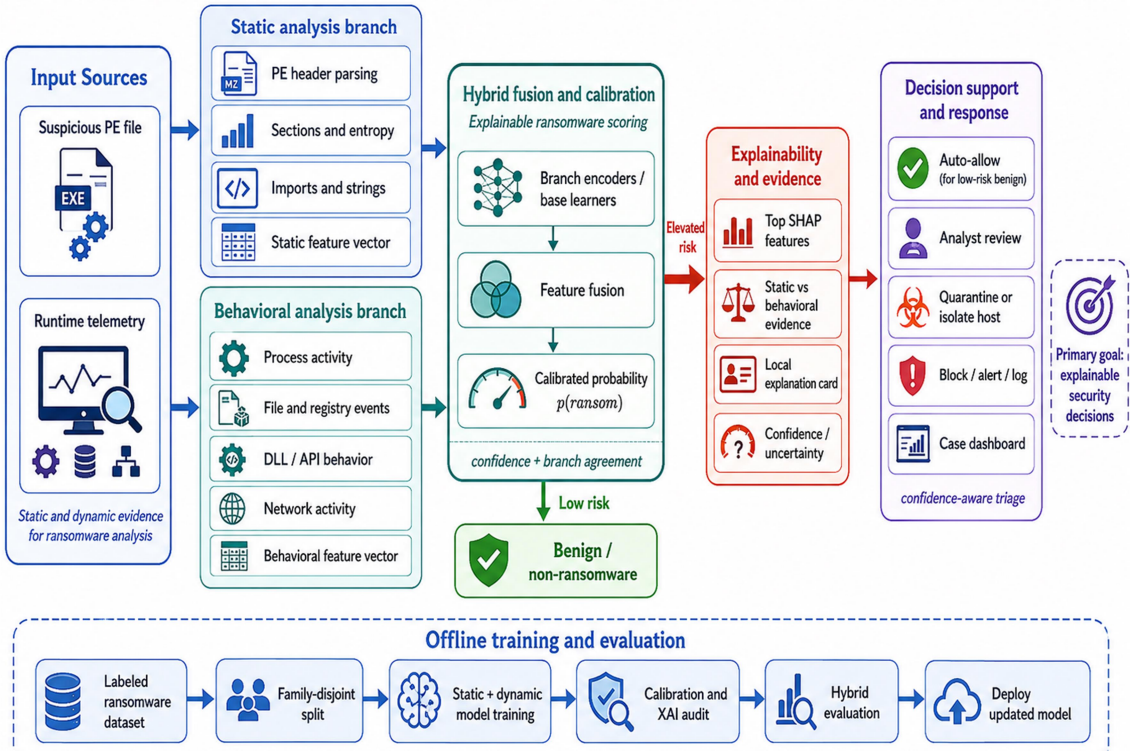


Figure 1. Architecture of the proposed RANSOM-XAI framework

3.1 Threat Model and Trust Assumptions

The defender can inspect an executable in a controlled analysis environment and can extract PE metadata and summarized runtime events. The adversary may obfuscate or pack the executable, alter nonfunctional header fields, delay activity, or suppress selected behaviors in a sandbox. The training labels, feature-extraction pipeline, stored models, and analysis host are trusted. File hashes and family labels are used only to remove leakage and define evaluation groups; they are never model inputs. The framework does not defend its own model files against tampering and does not replace endpoint detection, backups, application control, or incident-response procedures.

3.2 Static and Behavioral Branches

Let x_s denote the engineered static vector and x_b the behavioral vector. Categorical PE attributes are one-hot encoded, continuous values are median-imputed, and security-characteristic strings are converted into explicit flags and counts. The behavioral branch uses 18 numeric telemetry variables covering registry reads, writes, and deletions; DNS, HTTP, and connection activity; malicious, suspicious, and monitored processes; file findings; DLL calls; and API activity. The raw cryptographic identifiers md5 and sha1 are removed, as are the class, category, family, and file-extension metadata fields.

Two LightGBM classifiers learn class-score vectors z_s=f_s(x_s) and z_b=f_b(x_b). LightGBM is selected because histogram-based tree ensembles handle nonlinear interactions, mixed scales, sparse indicators, and missingness efficiently [21]. The branch probabilities are calibrated on a validation set with isotonic regression:

$$q_s = C_s(f_s(x_s)), \quad q_b = C_b(f_b(x_b)). \quad (1)$$

Calibration is performed after model fitting and only on the validation partition. This separation prevents the test set from influencing confidence estimates.

3.3 Explainable Stacking Fusion

The meta-classifier receives the concatenated calibrated probabilities rather than the original feature vectors. For three classes, the stacked input

contains six values. A multinomial logistic model produces the fused probability vector:

q = softmax(W [q_s || q_b] + c), y_hat = arg max_k q_k. (2)

Late stacking offers two transparency benefits. First, the final decision can be decomposed into static and behavioral support. Second, TreeSHAP can explain each branch in its own feature space before the low-dimensional meta-model combines them. This avoids presenting one blended ranking that obscures whether a decision was driven by executable structure or runtime behavior.

3.4 Confidence-Aware Security Decisions

The operational policy uses both branch agreement and fused confidence. A decision is eligible for automation when the static and behavioral branch labels agree and max(q) is at least 0.85. Otherwise, the outcome is marked for analyst review. The threshold is a transparent policy parameter rather than a universal safety guarantee. The review package includes the fused class probabilities, both branch predictions, confidence, and the leading positive and negative SHAP contributions.

action(x) = automate, if y_s = y_b and max(q) >= 0.85; review, otherwise. (3)

Algorithm 1. Family-disjoint training and explanation audit

Input: Labeled static and behavioral feature table D with family identifiers used only for splitting.
Output: Calibrated branch models, stacking model, explanation audit, and deployment package.
1: Remove hashes, labels, family metadata, and other identity-bearing fields from the model features.
2: Hold complete ransomware and other-malware families out for testing; stratify the remaining samples into training and validation partitions.
3: Engineer PE security flags, impute missing numeric values, and encode categorical PE attributes.
4: Fit static and behavioral LightGBM branches on the training partition.
5: Calibrate each branch on validation data and train a multinomial logistic stacker from the six calibrated probabilities.
6: Evaluate family-disjoint and stratified-random protocols using accuracy, macro-F1, class F1, MCC, log loss, Brier score, ECE, and ROC-AUC.
7: Compute branch-specific TreeSHAP explanations; audit compactness, masking faithfulness, and bootstrap stability.
8: Run branch-outage, feature-dropout, telemetry-noise, latency, and model-size tests; save artifacts and hashes.

Algorithm 2. Explainable inference and selective response

Input: Static PE vector x_s, behavioral vector x_b, calibrated models, stacker, and policy threshold 0.85.
Output: Predicted class, probability vector, explanation package, and automate-or-review action.
1: Apply the fitted static and behavioral preprocessors without using file hashes or family labels.
2: Generate and calibrate the branch probability vectors q_s and q_b.
3: Fuse q_s and q_b with the trained logistic stacker to obtain q and y_hat.
4: Generate branch-specific SHAP evidence and identify the smallest ranked feature set explaining 80% of attribution mass.
5: If the branch labels agree and max(q) >= 0.85, mark the decision eligible for automated policy handling.
6: Otherwise route the sample, probabilities, disagreement status, and evidence to analyst review.
7: Log the decision, confidence, model version, feature availability, and explanation summary for audit and drift monitoring.

4. EXPERIMENTAL METHODOLOGY

4.1 Dataset and Label Construction

The experiment uses Ransomware Dataset 2024 [22], supplied as a deduplicated feature table. It contains 21,752 observations: 10,876 benign samples, 4,762 ransomware samples, and 6,114 other-malware samples. Other malware comprises remote-access trojans, stealers, and trojans. The table contains 77 raw columns. After deterministic feature engineering, the model space contains 90 variables: 72 static variables and 18 dynamic variables. The study uses only the extracted feature table; it does not distribute or execute malware binaries.

Class is reconstructed from the dataset category: Benign remains benign, Ransomware remains ransomware, and RAT, Stealer, and Trojan categories are grouped as Other Malware. This formulation tests whether the detector distinguishes ransomware-specific evidence from generic malicious behavior. The dataset SHA-256 digest is dd8347389b98dc4ad9552353262079efab3c600148bcfa1d0c1f94186b27cb8e.

Table 2. Dataset composition and family-disjoint partition

Partition	n	Benign	Other malware	Ransom.
Full dataset	21,752	10,876	6,114	4,762
Training	12,185	-	-	-
Validation	3,438	-	-	-
Family-disjoint test	6,129	2,719	1,839	1,571

4.2 Family-Disjoint and Random Protocols

The primary protocol excludes four ransomware families - DarkSide, Maze, Ryuk, and WannaCry - and four other-malware families - Emotet, Formbook, NanoCore, and RedLine - from both training and calibration. A deterministic 25% benign sample is added to the test partition. The remaining observations are split into training and validation subsets with class stratification. This design asks whether the learned evidence transfers to named families absent from fitting, rather than measuring recognition of shuffled family variants.

A conventional stratified-random split is also evaluated as a diagnostic. Its purpose is not to replace the primary test but to quantify the optimism produced when related family observations can appear across partitions. All preprocessing objects are fitted on training data only, branch calibration and stacking use validation

data only, and every reported final metric is calculated once on the untouched test partition.

4.3 Models, Baselines, and Hyperparameters

Design comparability. The stratified-random protocol provides a within-collection reference, not an external validation cohort. The family-disjoint protocol is the primary analysis; all eight excluded malware families are unseen by model training and probability calibration. This design is stricter in the particular dimension of family identity, but does not prove independence across capture environments, tools, or feature-extraction artifacts. Literature comparisons are therefore qualitative unless methods and held-out families are aligned.

The static, behavioral, and early-fusion models use 160-tree multiclass LightGBM classifiers with learning rate 0.06, 31 leaves, minimum child size 24, feature subsampling 0.85, row subsampling 0.90, and L2 regularization 0.8. Random forest and Extra Trees use 100 trees with square-root feature sampling and class balancing. Multinomial logistic regression uses balanced class weights and a maximum of 1,800 iterations. The proposed meta-model is balanced multinomial logistic regression trained on validation-set calibrated probabilities. A fixed random seed of 20260806 is used throughout.

4.4 Evaluation Metrics

Accuracy is reported for comparability, but macro-F1 and MCC are primary because the three classes differ in prevalence and operational cost. Ransomware precision, recall, and F1 expose the target-class trade-off. Log loss, multiclass Brier score, and 15-bin expected calibration error (ECE) assess probability quality; macro one-versus-rest ROC-AUC assesses ranking. Confusion matrices and per-family recall reveal where aggregate scores conceal transfer failures [23], [24].

4.5 Explanation, Robustness, and Efficiency Audits

TreeSHAP explanations are calculated for 100 correctly and incorrectly represented ransomware test observations from the static and behavioral models. Compactness is the median number of ranked features required to capture 80% of absolute SHAP mass. Faithfulness masks the top five static and top three behavioral features to their median transformed values and compares the ransomware-probability reduction with equally sized random masks. Stability is the mean pairwise Jaccard similarity among top-10 global rankings from ten bootstrap resamples.

Robustness tests independently replace 5%, 10%, and 20% of transformed feature cells with training medians; remove one complete branch; and inject Gaussian noise into behavioral features at 1%, 5%, and 10% of feature standard deviation. Efficiency is measured in the recorded Python environment using repeated batched inference. The timing is not presented as end-to-end incident-response latency because it excludes sandbox execution and feature collection.

5. RESULTS AND ANALYSIS

5.1 Model Comparison and Split Optimism

Under family-disjoint evaluation, RANSOM-XAI reaches 0.9744 accuracy, 0.9716 macro-F1, 0.9606 MCC, and 0.9647 ransomware F1. These values satisfy H1. Early-fusion LightGBM is marginally stronger on raw macro-F1 (0.9735) and ransomware F1 (0.9724). The proposed framework is therefore not claimed as the numerical winner. Its contribution is the calibrated, branch-preserving evidence and selective decision layer that early fusion does not provide. Dynamic-only LightGBM remains strong (0.9685 macro-F1), whereas static-only performance falls to 0.5941 macro-F1 and 0.3621 ransomware F1, demonstrating that PE evidence learned from the training families transfers poorly by itself.

The stratified-random protocol raises proposed macro-F1 from 0.9716 to 0.9919 and ransomware F1 from 0.9647 to 0.9904. The 2.03 percentage-point macro-F1 difference is substantial for a high-performing classifier and supports the use of family-disjoint testing. Figure 2 compares the main models under the harder protocol.

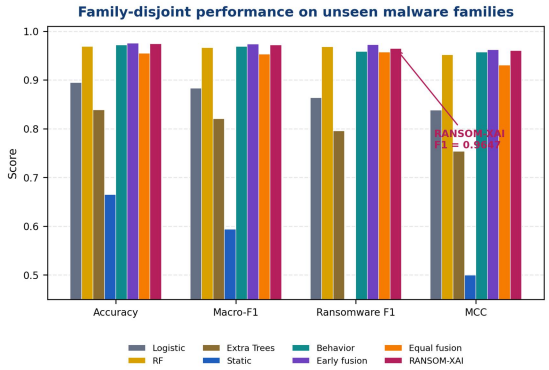


Figure 2. Family-Disjoint Comparison Of Ransomware-Detection Models

Table 3. Family-Disjoint Predictive And Calibration Performance

Model	Acc.	Macro-F1	MCC	Ransom F1	ECE
Logistic	0.8949	0.8832	0.8380	0.8637	0.0207
Random forest	0.9687	0.9664	0.9517	0.9681	0.1225
Extra Trees	0.8388	0.8206	0.7534	0.7958	0.0554
Static LGBM	0.6647	0.5941	0.4996	0.3621	0.1840
Behavior LGBM	0.9721	0.9685	0.9570	0.9584	0.0116
Early fusion	0.9755	0.9735	0.9622	0.9724	0.0091
Equal late fusion	0.9546	0.9528	0.9304	0.9571	0.1361
RANSOM-XAI	0.9744	0.9716	0.9606	0.9647	0.0178

Comparative interpretation. Relative to early fusion, the proposed late stacker trades 0.0019 macro-F1 (0.19 percentage points) and 0.0077 ransomware F1 (0.77 percentage points) for separately inspectable and calibratable evidence branches. The random-versus-family-disjoint macro-F1 difference is 0.0203 (2.03 percentage points). These differences are arithmetic contrasts from the reported Table 3 and protocol scores; no inferential confidence interval or significance test is asserted. Therefore, H1 assesses adequate transfer, not superiority over every comparator.

5.2 Class-Level Error Structure

The proposed confusion matrix contains 95 ransomware errors: 82 ransomware samples are classified as other malware, and 13 are classified as benign. Conversely, only 11 other-malware and two benign samples are incorrectly labeled ransomware. Ransomware precision is therefore high (0.9913), while recall is lower (0.9395). This asymmetry is appropriate for interpreting the system: it rarely issues a ransomware label without support, but some unseen ransomware behavior resembles other malware and would require policy review or complementary controls.

Benign F1 is 0.9883, other-malware F1 is 0.9619, and ransomware F1 is 0.9647. Figure 3 shows the confusion matrix and the precision-recall trade-off. The results are not evidence that all ransomware stages are detected before damage; the table contains aggregate static and sandbox features rather than a time-to-encryption sequence.

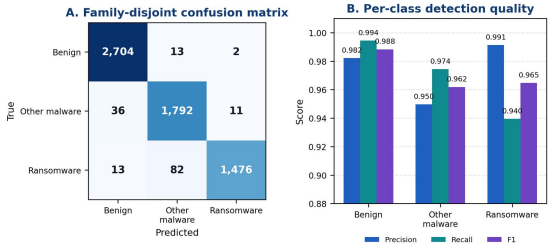


Figure 3. Family-disjoint confusion matrix and per-class metrics

Operational error analysis. On the 1,571 held-out ransomware samples, 95 were assigned another class (82 other malware and 13 benign), giving a false-negative proportion of 95/1,571 = 6.05%. Conversely, only 13 non-ransomware observations were incorrectly classified as ransomware (11 other malware and two benign). These errors are not equivalent in an incident-response setting. The first category potentially delays ransomware triage, while the second can trigger unnecessary investigation or containment. This class-level asymmetry is more informative to practitioners than overall accuracy alone.

5.3 Transfer to Unseen Ransomware Families

Family-level analysis reveals meaningful variation. Maze recall is 1.0000 and WannaCry recall is 0.9879, while DarkSide and Ryuk reach 0.8896 and 0.8842. Mean ransomware probability follows a similar pattern, ranging from 0.8371 for Ryuk to 0.9003 for Maze. The result indicates that a strong aggregate score does not eliminate family-specific blind spots. DarkSide and Ryuk deserve targeted error analysis and additional behavioral coverage before deployment.

The review policy is intentionally conservative for unseen ransomware. It routes 63.88% of DarkSide, 94.83% of Maze, 52.56% of Ryuk, and 98.79% of WannaCry cases to analyst review because branch agreement is low even when the fused model is correct. This behavior exposes the static branch weakness rather than allowing a high dynamic score to hide it. Figure 4 and Table 4 summarize the transfer results.

Table 4. Held-out ransomware-family performance

Family	n	Recall	Mean p(ransom)	Review rate
DarkSide	335	0.8896	0.8624	0.6388
Maze	290	1.0000	0.9003	0.9483
Ryuk	449	0.8842	0.8371	0.5256

Family	n	Recall	Mean p(ransom)	Review rate
WannaCry	497	0.9879	0.8952	0.9879

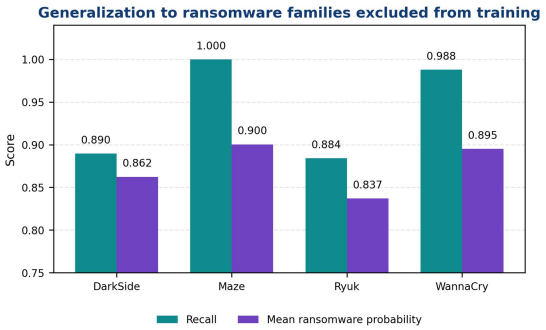


Figure 4. Generalization To Ransomware Families Excluded From Training

Unseen-family interpretation. The weakest held-out ransomware recalls, Ryuk 0.8842 and DarkSide 0.8896, lag the best observed family (Maze 1.0000) by 11.58 and 11.04 percentage points, respectively. A robust operational claim must therefore be conditional on family composition. The elevated review rates for Maze and WannaCry despite high recalls further demonstrate that classifier correctness and automation eligibility are distinct outcomes. Targeted per-family review and revalidation would be needed before deploying an automated containment policy.

5.4 Explanation Faithfulness and Stability

The static branch is dominated by OperatingSystemVersion, followed by SizeOfImage, .rdata offsets, DLL-characteristic counts, SectionAlignment, and section sizes. The behavioral branch is led by suspicious files, malicious processes, text-file activity, DLL calls, unknown files, and process counts. The difference is operationally useful: static explanations describe executable construction, whereas behavioral explanations describe observed activity. The latter aligns more directly with the branch that generalizes to unseen families.

The explanation audit supports H2. The median explanation needs 7.5 features to cover 80% of absolute attribution mass. Replacing the highest-ranked evidence reduces mean ransomware probability by 0.6037, compared with 0.1090 for random masks, a 5.54-fold ratio. Bootstrap top-10 Jaccard stability is 0.9636. These tests do not establish causal features, but they show that the reported rankings are more influential and stable than arbitrary alternatives.

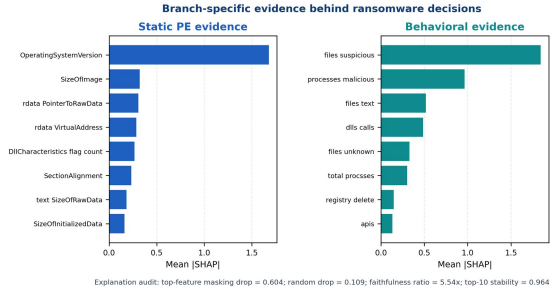


Figure 5. Branch-Specific SHAP Evidence And Explanation-Audit Results

Explanation limits. The 5.54-fold faithfulness ratio evaluates score sensitivity to masking the ranked variables, not whether those variables causally generated malicious behavior. Likewise, 0.9636 top-10 Jaccard stability measures consistency of a global ranking under the implemented resampling procedure; it does not establish stability under new families, new sandbox platforms, or adaptive manipulation. The central contribution relative to merely showing SHAP plots is the inclusion of measurable checks and their explicit limitations.

5.5 Robustness and Branch Dependence

Random feature dropout produces a gradual decline: macro-F1 decreases from 0.9716 to 0.9425 at 5%, 0.9273 at 10%, and 0.8759 at 20%. The system is more fragile to behavior corruption. Gaussian perturbation equal to only 1% of each dynamic feature standard deviation reduces macro-F1 to 0.8014, with 10% noise reducing it to 0.7015. These results show that the model is sensitive not merely to missing cells but to distorted behavioral magnitudes.

Branch-outage tests clarify the asymmetry. When static evidence is replaced by its median baseline, macro-F1 remains 0.9676. When behavioral evidence is unavailable, macro-F1 collapses to 0.1539 and the model predicts Other Malware for the test set. The term hybrid must therefore not be interpreted as equal dependence. In this dataset, behavior is the transferable detection channel and static evidence mainly contributes complementary context, agreement testing, and explanation. Reliable instrumentation, schema validation, range checks, and missing-branch alarms are deployment requirements.

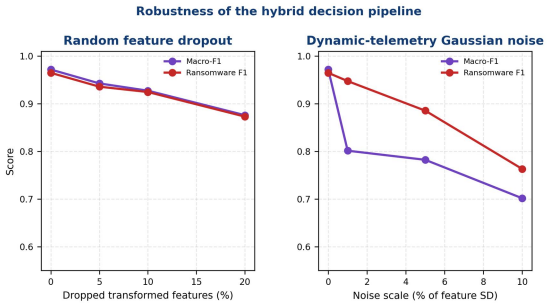


Figure 6. Robustness Under Feature Dropout And Dynamic-Telemetry Corruption

Table 5. Hybrid Robustness And Branch-Ablation Results

Condition	Macro-F1	Ransom F1	MCC	ECE
Clean	0.9716	0.9647	0.9606	0.0178
Feature dropout 5%	0.9425	0.9357	0.9186	0.0198
Feature dropout 10%	0.9273	0.9245	0.8958	0.0268
Feature dropout 20%	0.8759	0.8730	0.8232	0.0647
Dynamic branch unavailable	0.1539	0.0000	0.0000	0.6516
Static branch unavailable	0.9676	0.9576	0.9556	0.0288
Dynamic Gaussian noise 1% SD	0.8014	0.9474	0.7193	0.1442
Dynamic Gaussian noise 5% SD	0.7822	0.8854	0.6745	0.1438
Dynamic Gaussian noise 10% SD	0.7015	0.7630	0.5490	0.2064

Behavior dependence and failure mechanism. The branch-outage contrast is unusually sharp: static-branch removal retains macro-F1 0.9676, only 0.0040 below clean performance, whereas behavioral-branch removal lowers macro-F1 by 0.8177 to 0.1539. The 1%-SD behavioral noise experiment lowers macro-F1 by 0.1702. Thus, fusion provides decision-level transparency but not redundancy against failure of the dynamic channel. This open issue requires missingness alarms, data quality checks, and a fail-to-review policy instead of interpreting the second branch as an adequate standby detector.

5.6 Confidence-Aware Triage

The agreement-confidence rule automates 66.24% of test decisions and sends 33.76% to review. Accuracy among automated decisions is 0.9882, satisfying H3. Branch agreement is 0.6628, and mean fused confidence is 0.9666. Only 22.60% of ransomware cases are auto-covered because the weak static branch often disagrees with the correct behavioral prediction. This is a safety-performance trade-off, not a hidden success metric.

The policy can be tuned by changing the 0.85 confidence threshold or allowing behavior-dominant automation under additional controls. The present rule intentionally requires both evidence sources to agree, making its decisions easier to audit at the cost of high ransomware review volume. Figure 7 separates overall automation coverage, automated accuracy, agreement, and ransomware auto-coverage.

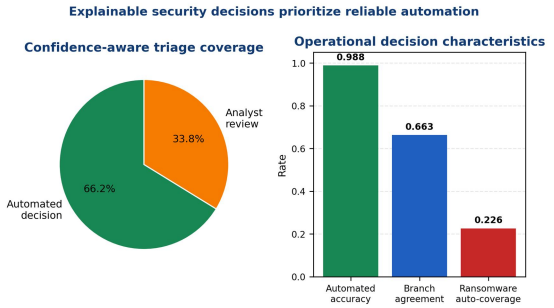


Figure 7. Coverage And Reliability Of Confidence-Aware Security Decisions

Coverage-risk trade-off. Of all held-out samples, 33.76% are withheld for review; the automated subset achieves 98.82% accuracy but its distribution differs from the complete test set. Ransomware auto-coverage is only 22.60%, exposing potential analyst workload and delayed action as practical costs. H3 is met under the specified retrospective criterion, but the paper does not evaluate staffing capacity, response timeliness, or error costs under operational ransomware prevalence.

5.7 Calibration and Efficiency

RANSOM-XAI has family-disjoint log loss 0.1078, multiclass Brier score 0.0461, and ECE 0.0178. Early fusion has slightly lower log loss, Brier score, and ECE, while equal-probability fusion is notably overconfident (ECE 0.1361). The proposed calibration is therefore adequate for a transparent policy but not uniformly superior to every baseline. Reliability monitoring should be repeated when family mix, sandbox configuration, or telemetry prevalence changes.

The serialized preprocessors, models, calibrators, and stacker occupy 6.33 MiB. Model-only batched inference has median cost 0.01318 ms per sample and p95 0.01668 ms for batches of 1024. Including feature preprocessing, a batch of 512 has median 0.03278 ms per sample and median batch time 16.78 ms. These figures demonstrate computational

compactness after features exist; they exclude dynamic execution and collection, which dominate practical analysis time.

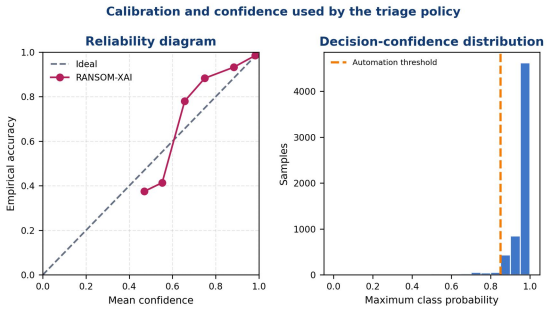


Figure 8. Probability calibration and decision-confidence distribution

Calibration and efficiency qualification. The model ECE of 0.0178 is worse than the early-fusion ECE of 0.0091 in this test, and the reported 6.33-MiB artifact footprint excludes capture and sandbox instrumentation. Therefore, the operational contribution is not the globally best calibrated detector nor an end-to-end latency improvement. Reliable interpretation requires disaggregating model inference, data acquisition, dynamic execution time, confidence reliability, and policy approval.

5.8 Evidence against Research Questions

RQ1 and H1 are supported: family-disjoint macro-F1 is 0.9716 and ransomware F1 is 0.9647, both above the prespecified 0.90 target. RQ2 and H2 are supported by a 5.54 explanation-faithfulness ratio and 0.9636 top-10 bootstrap Jaccard stability. RQ3 and H3 are supported because automated decisions reach 0.9882 accuracy at 66.24% coverage.

The robustness evidence is mixed: random feature dropout degrades gradually, but dynamic-telemetry perturbation and complete behavioral-branch loss are severe. A raw-score dominance claim is intentionally rejected because early fusion is slightly stronger in family-disjoint F1. The proposed method is justified by auditable branch evidence, calibrated confidence, and selective response rather than numerical supremacy.

Research-question synthesis. RQ1/H1 is supported on the specified unseen-family partition by macro-F1 0.9716 and ransomware F1 0.9647; the weaker DarkSide and Ryuk recalls are important exceptions. RQ2/H2 is supported within the defined perturbation audit by a 5.54 masking ratio and 0.9636 stability, without causal attribution. RQ3/H3 is supported retrospectively by 66.24% automation coverage at 98.82% automated accuracy, but only 22.60% of ransomware flows satisfy the rule. Together, the results identify the conditions under which the framework is useful and the failure modes requiring future evaluation.

6. DISCUSSION

6.1 What the Hybrid Results Mean

The results demonstrate complementarity but not symmetry. Runtime behavior is the primary generalization channel, while static PE evidence is highly sensitive to the families represented during fitting. This finding is consistent with the conceptual advantage of behavior: encryption-related file activity, suspicious process evidence, DLL interaction, and registry changes can recur across implementations even when executable construction changes. Static metadata remains useful for pre-execution context, fast screening, and cross-checking, but should not be treated as a sufficient fallback for unseen-family detection in this dataset.

Early fusion slightly exceeds the proposed stacker in raw predictive metrics because a single tree ensemble can learn cross-feature interactions directly. RANSOM-XAI accepts a small score trade-off to obtain calibrated branch probabilities, separate explanation channels, explicit disagreement detection, and selective automation. The appropriate model therefore depends on the deployment objective. A batch triage system seeking only ranking quality might choose early fusion; an operational control requiring traceable evidence and abstention may prefer the proposed design.

Positioning relative to the state of the art. The existing literature covers PE-header models [3], explainable behavior-only models [4], [5], and a hybrid runtime framework [6]. These studies differ in samples, execution environments, and evaluation protocols, preventing a fair numerical leaderboard. The distinctive contribution here is an integrated experiment that holds out entire named families, preserves source-specific explanations, audits their importance and stability, examines complete branch loss, and reports the coverage-versus-accuracy of an explicit selective decision rule. The results also show that direct early fusion remains a legitimate choice when maximum predictive F1 is the principal objective.

6.2 Explainable Security Decisions

An explanation is useful only when it changes or supports a decision process. RANSOM-XAI links branch-specific feature evidence to a concrete action: automation is allowed only under confidence and agreement; other cases are reviewed. The explanation audit adds three quality checks. Compactness controls cognitive burden, perturbation tests whether the ranked evidence matters to the score, and bootstrap stability checks whether the global ranking is reproducible across resampled explanations. These tests remain correlational, and adaptive adversaries could manipulate prominent features, but they are stronger evidence than an unattributed bar chart.

A production implementation should present local values alongside SHAP direction, compare them with benign and malware baselines, record feature availability, and attach a model version. Analysts should be able to distinguish "ransomware because suspicious file and process behavior is present" from "ransomware because the executable resembles training-family PE structure." The latter warrants more caution under family shift.

6.3 Deployment and Governance Implications

The stress tests make telemetry governance central to deployment. Dynamic values require schema validation, unit consistency, range monitoring, and explicit missingness indicators. A branch-outage monitor should stop automated actions rather than silently supplying default values. Calibration and review volume should be tracked by family, time window, endpoint type, and analysis environment. NIST AI risk-management guidance recommends documented validity, transparency, monitoring, and human oversight for consequential AI-supported decisions [25]. These principles align with the proposed selective policy.

Containment actions also require organizational policy. The classifier can recommend alert, isolate, or block outcomes, but downstream execution should consider asset criticality, business continuity, backup status, and incident-response authority. High-confidence model agreement is not proof that quarantine is harmless. The article therefore evaluates decision reliability, not the effectiveness of containment or recovery.

6.4 Limitations and Threats to Validity

The first limitation is dataset representativeness. The feature table aggregates a particular collection and sandbox process; family names may inherit vendor-label uncertainty, and class balance differs from operational prevalence. Second, family-disjoint evaluation is stricter than random splitting but is not a genuine future zero-day test. Held-out families may still share tools, packers, or behavior with training families. Third, the dynamic variables are summary counts rather than event sequences, so the model cannot establish detection time relative to encryption.

Fourth, the perturbation tests approximate missing or corrupted telemetry but do not simulate an adaptive ransomware author. Fifth, SHAP explanations describe model dependence, not causal mechanisms. Correlated PE and behavioral features can share or shift attribution. Sixth, the confidence policy is evaluated retrospectively and its threshold is not optimized for a specific organization's risk cost. Seventh, latency measurements begin after feature extraction and should not be compared directly with systems that include sandbox runtime.

External validation should therefore use independently collected datasets, temporal holdouts, alternative sandboxes, packed and evasive samples, and endpoint traces with time-to-encryption labels. Prospective analyst studies are also needed to determine whether the explanation package improves decision quality, review time, and trust calibration.

Outstanding issues and bounded claims. Still unresolved are dataset provenance across families, family-label uncertainty, external and time-separated replication, adaptive packer and sandbox evasion, telemetry availability in production, incident-response delays, false-positive costs, and analyst workload. The present retrospective, fixed-dataset study cannot establish incident containment effectiveness, prospective response safety, or genuine zero-day generalization. Future research should use independent collection sites, alternate sandbox instrumentation, family-blocked uncertainty estimates, time-to-encryption outcomes, and a prospective analyst-review trial.

7. CONCLUSION AND FUTURE WORK

This paper introduced RANSOM-XAI, an explainable static-behavioral fusion framework for ransomware detection under family-disjoint evaluation. The calibrated stacker achieved 97.44% accuracy, 0.9716 macro-F1, and 0.9647 ransomware F1 on test families excluded from training. The explanation audit showed a 5.54-fold faithfulness advantage over random masks and 0.9636 top-10 stability. A transparent agreement-confidence rule automated 66.24% of decisions at

98.82% accuracy while referring uncertainty to analysts.

The strongest practical conclusion is not that hybrid fusion automatically solves ransomware detection. Behavioral telemetry drives generalization, static transfer is weak, and corrupted runtime evidence can sharply degrade performance. Future work will evaluate independent corpora, event-sequence models, time-to-encryption detection, calibrated behavior-dominant fallback rules, adversarial feature manipulation, and analyst-centered decision studies. The released code, fixed split, hashes, metrics, figures, and trained artifacts support replication of the present claims.

Scientific contribution in context. Whereas comparable methods primarily emphasize a static representation, a behavioral detector, or a hybrid accuracy score [3]–[6], RANSOM-XAI combines (i) family-disjoint evaluation of three operationally different software classes, (ii) calibrated branch-preserving fusion with separate explanatory provenance, (iii) quantitative explanation faithfulness and stability audits, and (iv) an evaluated agreement-confidence abstention policy. Its measured strengths and weaknesses matter equally: 0.9716 macro-F1 on held-out families and 98.82% accuracy on the selectively automated subset coexist with weak static-only transfer and severe behavior-channel dependence. This is a reproducible contribution to accountable decision design, not a claim that hybrid fusion guarantees the best classifier or safe automatic containment.

DATA AND CODE AVAILABILITY

The public dataset is cited in [22]. The submission supplement contains the exact experimental script, split file, result tables, trained model artifacts, figures, environment requirements, and SHA-256 manifest. The supplement excludes malware binaries and contains only the public feature table and derived artifacts. A persistent repository DOI should be added by the author before final publication.

ETHICS AND SAFETY STATEMENT

The study analyzes an existing feature table and does not execute, reconstruct, or distribute ransomware. The code performs defensive classification, calibration, explanation, and robustness evaluation. Automated response is discussed as a policy recommendation; no destructive or offensive functionality is included.

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] Cybersecurity and Infrastructure Security Agency, 'StopRansomware Guide,' CISA, 2023-2026.

[2] J. Ferdous, R. Islam, A. Mahboubi, and M. Z. Islam, 'AI-Based Ransomware Detection: A Comprehensive Review,' IEEE Access, Vol. 12, 2024, pp. 136666-136695, doi:10.1109/ACCESS.2024.3461965.

[3] C. C. Moreira, D. C. Moreira, and C. de S. de Sales Jr., 'Improving Ransomware Detection Based on Portable Executable Header Using Xception Convolutional Neural Network,' Computers & Security, Vol. 130, 2023, 103265, doi:10.1016/j.cose.2023.103265.

[4] S. Gulmez, A. G. Kakisim, and I. Sogukpinar, 'XRan: Explainable Deep Learning-Based Ransomware Detection Using Dynamic Analysis,' Computers & Security, Vol. 139, 2024, 103703, doi:10.1016/j.cose.2024.103703.

[5] M. A. Alvi and Z. Jalil, 'XRGuard: A Model-Agnostic Approach to Ransomware Detection Using Dynamic Analysis and Explainable AI,' IEEE Access, Vol. 13, 2025, pp. 53159-53170, doi:10.1109/ACCESS.2025.3553562.

[6] B. Caliskan, I. Gulatas, H. H. Kilinc, and A. H. Zaim, 'RansomTrack: A Hybrid Behavioral Analysis Framework for Ransomware Detection,' arXiv:2604.08739, 2026.

[7] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, 'Dos and Don'ts of Machine Learning in Computer Security,' 31st USENIX Security Symposium, 2022, pp. 3971-3988.

[8] Y. Kaya et al., 'ML-Based Behavioral Malware Detection Is Far From a Solved Problem,' arXiv:2405.06124, revised 2025.

[9] A. Galli, V. La Gatta, V. Moscato, M. Postiglione, and G. Sperli, 'Explainability in AI-Based Behavioral Malware Detection Systems,' Computers & Security, Vol. 141, 2024, 103842, doi:10.1016/j.cose.2024.103842.

[10] H. Manthena et al., 'Explainable Artificial Intelligence for Malware Analysis: A Survey of Techniques, Applications, and Open Challenges,' arXiv:2409.13723, revised 2025.

[11] S. M. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions,' Advances in Neural Information Processing Systems 30, 2017, pp. 4765-4774.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, 'Why Should I Trust You? Explaining the Predictions of Any Classifier,' 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135-1144.

[13] U. Urooj, B. A. Al-rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, 'Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions,' Applied Sciences, Vol. 12, No. 1, 2022, 172, doi:10.3390/app12010172.

[14] A. Venckauskas et al., 'Zero-Day Ransomware Attack Detection Using Static Analysis and Machine Learning Algorithms,' Applied Sciences, Vol. 15, No. 19, 2025, 10576.

[15] H. S. Anderson and P. Roth, 'EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models,' arXiv:1804.04637, 2018.

[16] A. Hashmi, M. Sarwar, and I. U. Haq, 'Lightweight and Explainable Early Ransomware Detection Using Dynamic API-Call Features and Ensemble Machine Learning,' Engineering Reports, Vol. 8, No. 4, 2026, e70734, doi:10.1002/eng2.70734.

[17] J. Ferdous et al., 'A Novel Technique for Ransomware Detection Using Image-Based Dynamic Features and Transfer Learning to Address Dataset Limitations,' Scientific Reports, Vol. 15, 2025, 32342, doi:10.1038/s41598-025-17647-1.

[18] C. Molnar, Interpretable Machine Learning, 2nd ed., 2022.

[19] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, 'On Calibration of Modern Neural Networks,' 34th International Conference on Machine Learning, 2017, pp. 1321-1330.

[20] A. Niculescu-Mizil and R. Caruana, 'Predicting Good Probabilities with Supervised Learning,' 22nd International Conference on Machine Learning, 2005, pp. 625-632.

[21] G. Ke et al., 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree,' Advances in Neural Information Processing Systems 30, 2017, pp. 3146-3154.

[22] A. Hussain, 'Ransomware Dataset 2024,' Zenodo, Version 1.0, 2024, doi:10.5281/zenodo.13890887.

[23] B. W. Matthews, 'Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme,' Biochimica et Biophysica Acta, Vol. 405, No. 2, 1975, pp. 442-451.

[24] T. Saito and M. Rehmsmeier, 'The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets,' PLoS ONE, Vol. 10, No. 3, 2015, e0118432.

[25] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, 2023.

[26] L. Breiman, 'Random Forests,' Machine Learning, Vol. 45, 2001, pp. 5-32.

[27] P. Geurts, D. Ernst, and L. Wehenkel, 'Extremely Randomized Trees,' Machine Learning, Vol. 63, 2006, pp. 3-42.

[28] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System,' 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

APPENDIX A.

REPRODUCIBILITY CHECKPOINTS

Replication begins with the supplied feature table whose SHA-256 digest is dd8347389b98dc4ad9552353262079efab3c600148 bcfa1d0c1f94186b27cb8e. The fixed random seed is 20260806. The family-disjoint test must contain DarkSide, Maze, Ryuk, and WannaCry as ransomware families and Emotet, Formbook, NanoCore, and RedLine as other-malware families. The split file records the original row index, class, category, family, and assigned partition. A valid rerun should report 12,185 training, 3,438 validation, and 6,129 test observations.

The feature-engineering stage must exclude md5, sha1, file_extension, Class, Category, and Family. PE-characteristic strings are converted to explicit flags and flag counts; categorical PE fields are one-hot encoded with unknown-value handling; continuous variables are median-imputed using training statistics. Static and behavioral preprocessors are fitted independently. Branch classifiers are trained only on the training partition, isotonic calibrators and the logistic stacker only on validation data, and all reported metrics only on the untouched test partition.

A result is considered structurally reproducible when the class order is Benign, Other Malware, and Ransomware; the family-disjoint confusion matrix equals [[2704,13,2],[36,1792,11],[13,82,1476]]; and the proposed metrics round to accuracy 0.9744, macro-F1 0.9716, MCC 0.9606, and ransomware F1 0.9647. Small timing differences are expected across hardware and software versions, but model predictions and split membership should remain fixed under the stated environment.

APPENDIX B.

ANALYST DECISION-CARD SPECIFICATION

Each reviewed sample should be accompanied by a compact decision card containing the final class and three calibrated probabilities; the static and behavioral branch labels; branch agreement; the automation threshold; feature-availability status; and the model, preprocessor, and explanation versions. The evidence panel should list the leading positive and negative static SHAP values separately from the leading behavioral SHAP values, together with the observed feature values and reference ranges.

The card should distinguish model evidence from response authority. A high ransomware probability supported by suspicious file and process activity can justify urgent containment review, but a recommendation should still identify asset criticality, current backup status, business-owner notification, and the human or policy authority approving isolation. When the branches disagree or a behavioral feature channel is missing, the card should state that automation was withheld and should never silently replace unavailable evidence with a normal-state assumption.

For post-deployment monitoring, decision cards should be aggregated by time, endpoint group, predicted class, analyst disposition, and family when ground truth becomes available. Minimum monitoring indicators include review rate, automated coverage, automated error rate, confidence distribution, ECE, feature-missingness, branch disagreement, and shifts in the leading explanations. These records support calibration updates, threshold governance, incident reconstruction, and detection of telemetry or model drift.