JIRBDAI

Journal of Institutional Research, Big Data Analytics and Innovation

A Systematic Review of Hybrid Deep Learning and Explainable Artificial Intelligence Approaches for Polymorphic Malware Detection

***Amina Fadila Shehu, Enoch Haruna, Peter Ogedebe, Saidu Ibrahim**

Department of Computer Science, Baze University, Abuja, Nigeria

Email: fadila.shehu@bazeuniversity.edu.ng

ABSTRACT

Polymorphic malware remains difficult to detect because it alters its observable code structure while preserving malicious functionality, thereby weakening signature-based, static-only and single-modality learning systems. Objective: This paper systematically reviews recent literature on hybrid deep learning, multimodal feature representation, malware datasets and explainable artificial intelligence for polymorphic malware detection. Methods: A structured review protocol was applied to peer-reviewed and high-quality scholarly sources on malware detection, deep learning, explainability, datasets, obfuscation, polymorphism and metamorphism, with emphasis on works from 2020 to 2026 and foundational studies where necessary. Studies were grouped thematically according to detection approach, feature modality, dataset support, robustness and interpretability. Findings: The review shows that convolutional models are effective for byte and image representations, recurrent and gated models support API and opcode sequence learning, and multimodal fusion improves coverage against representation-changing malware. However, most studies remain limited by single-modality designs, weak polymorphic dataset validation, insufficient cross-dataset testing and limited explanation of feature contribution. Conclusion: The literature supports a research direction that combines hybrid CNN-LSTM-GRU modelling, polymorphism-aware dataset construction, ablation analysis and multi-method XAI. The review provides a structured research agenda for developing robust and interpretable malware detection systems suitable for evolving polymorphic threats.

KEYWORDS: Polymorphic Malware; Hybrid Deep Learning; Malware Detection; Explainable Artificial Intelligence; CNN; LSTM; GRU; Systematic Literature Review.

I. INTRODUCTION

Malware detection remains a major cyber security challenge because malicious software continually evolves to bypass defensive controls. Cohen (1987) established the theoretical foundations of self-replicating and modifying computer programs. Leder et al. (2009) later demonstrated how code obfuscation could be used to alter malware structure while retaining its underlying functionality. Metamorphic techniques further weaken fixed-signature detection by changing the observable form of malicious code (You & Yim, 2010). More recent reviews confirm that these transformations remain central to polymorphic malware detection research (Alrzini & Pennington, 2020; Brezinski & Ferens, 2023). Earlier malware-detection systems relied heavily on signatures, file hashes and fixed byte patterns, but these assumptions become unreliable when malware changes its representation without changing its malicious purpose. Machine learning and deep learning have increasingly been adopted to address the limitations of conventional malware detection (Bensaoud et al., 2024). CNN-based models can learn structural and spatial patterns from malware binaries. Nataraj et al. (2011) demonstrated the

Volume 2, Issue 3, 2026

https://doi.org/10.5281/zenodo.21777609

potential of binary-image representation, while Raff et al. (2018) examined learning directly from raw byte sequences. Vasan et al. (2020) subsequently showed the effectiveness of fine-tuned CNNs for colour malware images.

Sequence-based models provide a complementary view. Kolosnjaji et al. (2016) applied recurrent learning to ordered malware behavior, while Maniriho et al. (2023) used a CNN–BiGRU architecture to process API-call sequences. Hybrid architectures have therefore emerged to combine different malware representations, as demonstrated by Gibert et al. (2020) and Vuran Sarı and Acı (2025). However, high classification accuracy alone does not establish robustness against polymorphic transformation. Models may still rely on dataset artefacts or shortcut features that fail when attackers alter the underlying feature space (Arp et al., 2022; Pendlebury et al., 2019). Explainable artificial intelligence has become increasingly important in malware detection because analysts require defensible reasons for model predictions (Galli et al., 2024). Recent malware-specific systems illustrate this development. Alani and Awad (2022) used explainability in a lightweight Android malware detector, while Alani et al. (2023) applied SHAP to obfuscated-malware detection. Vuran Sarı and Acı (2025) combined SHAP and LIME with a hybrid CNN–GRU model. Different explanation methods serve different purposes. SHAP provides global and local feature attribution (Lundberg & Lee, 2017), while LIME explains individual predictions through interpretable local approximations (Ribeiro et al., 2016). Integrated gradients attributes neural-network predictions to input features (Sundararajan et al., 2017), whereas Grad-CAM highlights influential image regions in convolutional models (Selvaraju et al., 2017). These approaches can improve transparency, although their outputs must still be interpreted cautiously because post-hoc explanations may be unstable or vulnerable to manipulation (Slack et al., 2020).

This review responds to the need for a consolidated synthesis of polymorphic malware detection literature. It examines traditional detection, machine learning, deep learning, feature modalities, datasets, explainability, ablation and recent polymorphism-focused approaches. The review is organised around the following research questions:

(a). RQ1: What detection approaches have been used for polymorphic, metamorphic and obfuscation-resilient malware detection?

(b). RQ2: Which feature modalities are most frequently used, and what are their strengths and limitations for polymorphic malware?

(c). RQ3: How do existing datasets support or limit rigorous evaluation of polymorphic malware detection?

(d). RQ4: How are explainability and ablation used to interpret malware detection models?

(e). RQ5: What research gaps remain for robust and interpretable hybrid deep learning detection of polymorphic malware?

II. METHODOLOGY

This paper follows a structured systematic review approach suitable for computing and cybersecurity research. The review was designed using PRISMA 2020 reporting principles to make the identification, screening, eligibility assessment and inclusion of studies transparent and reproducible (Page et al., 2021). A multi-database search log, screening spreadsheet, reviewer-agreement sheet, PRISMA flow diagram and quality-appraisal checklist were prepared to document the review process and support the credibility of the final synthesis.

A. Search Strategy

A structured multi-database search strategy was used to identify relevant studies on polymorphic malware detection, hybrid deep learning, multimodal malware representation, dataset benchmarking and explainable artificial intelligence. The search was conducted on 4 March 2026 across Google Scholar,

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink and Scopus/Web of Science. These databases were selected because they index major cyber security, computer science, and artificial intelligence and software engineering publications. The search covered studies published between 2020 and 2026, while older foundational works were retained where they introduced core concepts, datasets, algorithms or explainability techniques that remain relevant to the review. The search terms were organized into five concept groups: polymorphic and metamorphic malware, deep learning detection, malware feature modalities, datasets and benchmarks, and explainable artificial intelligence.

The core search expression was adapted for each database using equivalent Boolean operators and database-specific syntax. The main search expression combined the following terms:

("polymorphic malware" OR "metamorphic malware" OR "obfuscated malware" OR "malware evasion") AND ("deep learning" OR CNN OR LSTM OR GRU OR Transformer OR "hybrid model") AND ("malware detection" OR "malware classification") AND ("explainable AI" OR XAI OR SHAP OR LIME OR "integrated gradients" OR Grad-CAM OR ablation)

Additional targeted searches were conducted for major malware datasets and benchmarks, including EMBER, SOREL-20M, BODMAS, Malimg, Microsoft BIG 2015 and Android malware datasets. This was necessary because several dataset papers are not always retrieved through polymorphic-malware search strings but are central to evaluating the robustness and limitations of malware detection research. The initial search retrieved 476 records before year filtering. After applying the 2020–2026 publication-year filter, 283 records remained. Older foundational studies were added manually where they were necessary for explaining core concepts such as computer-virus theory, recurrent neural networks, convolutional learning, malware image representation and explainable AI methods.

Table 1: Database-Level Search Log for the Systematic Review

Database	Search Date	Search Focus	Records Retrieved before Year Filter	Records Retained after 2020–2026 Filter
Google Scholar	4 March 2026	Broad search across polymorphic malware, deep learning, hybrid detection and XAI	210	126
IEEE Xplore	4 March 2026	Engineering and cyber security studies on malware detection, CNN, LSTM, GRU and hybrid models	78	46
ACM Digital Library	4 March 2026	Computer science studies on malware analysis, machine learning, datasets and explainability	42	24
ScienceDirect	4 March 2026	Journal articles on malware detection, deep learning, explainable AI and cyber security analytics	58	35
SpringerLink	4 March 2026	Cyber security, artificial intelligence and malware-detection studies	47	28
Scopus/Web of Science	4 March 2026	Indexed multidisciplinary literature and citation-tracked cyber security studies	41	24
Total			**476**	**283**

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

The database-level search log shows that 476 records were initially retrieved across six scholarly sources. After applying the 2020–2026 publication-year filter, 283 records remained for deduplication and preliminary relevance filtering. Google Scholar produced the largest number of records because of its broad indexing coverage, while IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink and Scopus/Web of Science provided more targeted peer-reviewed sources in cyber security, artificial intelligence and computer science. The use of multiple databases was intended to reduce the risk of source bias and improve the reproducibility of the review process.

B. Inclusion and Exclusion Criteria

The inclusion and exclusion criteria used in the review is presented in Table 2:

Table 2: Inclusion and Exclusion Criteria for Selecting Studies

Criterion type	Included studies	Excluded studies
Topic relevance	Studies addressing malware detection, malware classification, polymorphism, obfuscation, metamorphism, adversarial evasion, feature representation, datasets or XAI in malware analysis.	Studies unrelated to malware detection or cyber security machine learning.
Methodological relevance	Studies presenting detection models, datasets, systematic reviews, surveys, feature analyses, XAI methods or evaluation frameworks.	Opinion pieces without technical or methodological contribution.
Publication period	Mainly 2020–2026, with older foundational works retained when necessary.	Older works with no foundational relevance and no continuing influence.
Evidence quality	Peer-reviewed journals, conferences and reputable preprints where the method and data are clear.	Unverifiable sources, promotional material or duplicated publications.
Language	English-language scholarly publications.	Non-English sources not accompanied by reliable English metadata.

C. Screening and Data Extraction

The screening process followed a transparent multi-stage procedure. First, all records retrieved from the database search were exported into a consolidated screening spreadsheet. The initial search produced 476 records across Google Scholar, IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink and Scopus/Web of Science. After applying the 2020–2026 publication-year filter, 283 records remained. The 283 records were then reviewed for duplication and preliminary relevance. Eighteen duplicate records were removed. A further 131 records were removed during preliminary filtering because they were outside the scope of the review, were not scholarly research papers, were not focused on malware detection, or did not address deep learning, polymorphism, obfuscation, datasets, explainability or cyber security machine learning. This left 134 records for title and abstract screening. During title and abstract screening, 70 records were excluded because they did not sufficiently address the review focus. The most common reasons for exclusion at this stage were: general cyber security relevance without malware detection, malware studies without deep learning or machine learning methods, papers unrelated to polymorphism or obfuscation-resilient detection, and papers that did not provide relevant evidence for the research questions. The remaining 64 papers were assessed at full-text eligibility stage.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

At the full-text stage, 25 papers were excluded. Twelve were excluded because they did not present a deep learning or hybrid learning method relevant to the review. Eight were excluded because they did not report a malware dataset or provided insufficient dataset information for meaningful appraisal. Five were excluded because they did not contribute evidence on explainability, polymorphic malware, obfuscation, multimodal representation or the stated review questions. After this process, 39 studies were retained for qualitative synthesis. For each included study, data were extracted using a structured extraction form. The extracted fields included author, year, study aim, malware platform, malware type, dataset, detection method, feature modality, model architecture, evaluation metrics, explainability method, ablation evidence, robustness testing and stated limitations. These extracted fields were used to organize the thematic synthesis, comparative tables and research-gap analysis.

PRISMA 2020 Flow Reporting

The PRISMA 2020 flow was used to report how studies moved through the identification, screening, eligibility and inclusion stages. The review initially identified 476 records. After applying the 2020–2026 date filter, 283 records remained. From these, 18 duplicates were removed and 131 records were removed during preliminary source, document-type and topic filtering. This left 134 records for title and abstract screening. Of the 134 records screened, 70 were excluded for lack of relevance to the review focus. The remaining 64 records were assessed in full text. At the full-text eligibility stage, 25 papers were excluded for documented reasons: 12 did not present a relevant deep learning or hybrid learning method, 8 did not provide sufficient malware dataset information, and 5 did not contribute directly to explainability, polymorphism, obfuscation or the stated review questions. Therefore, 39 studies were included in the final qualitative synthesis. This flow clarifies the transition from the initial search results to the final included studies and supports the reproducibility of the review process. A supplementary study-selection spreadsheet should be retained with the manuscript to document the title, source database, screening decision and exclusion reason for each record assessed during the review.

Table 3: PRISMA 2020-Style Screening Summary

Review stage	Number of Records	Explanation
Records identified	476	Records retrieved across Google Scholar, IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink and Scopus/Web of Science.
Records retained after 2020–2026 filter	283	Records retained after applying the publication-year filter. Older foundational studies were retained manually where necessary.
Duplicates removed	18	Duplicate records identified across databases and removed.
Preliminary exclusions	131	Records removed because they were outside scope, non-scholarly, not focused on malware detection, or not related to deep learning, polymorphism, obfuscation, datasets, XAI or cyber security machine learning.
Records screened by title and abstract	134	Records remaining after deduplication and preliminary filtering.
Records excluded after title and abstract screening	70	Records excluded because they did not sufficiently address the review focus or research questions.
Full-text papers assessed	64	Papers assessed in full text for eligibility.
Full-text papers excluded	25	Excluded because of no relevant deep learning/hybrid method (n = 12), insufficient malware dataset reporting (n = 8), or limited relevance to XAI, polymorphism, obfuscation or the review focus (n = 5).
Studies included in qualitative synthesis	39	Final studies retained for thematic synthesis, comparative analysis and research-gap identification.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

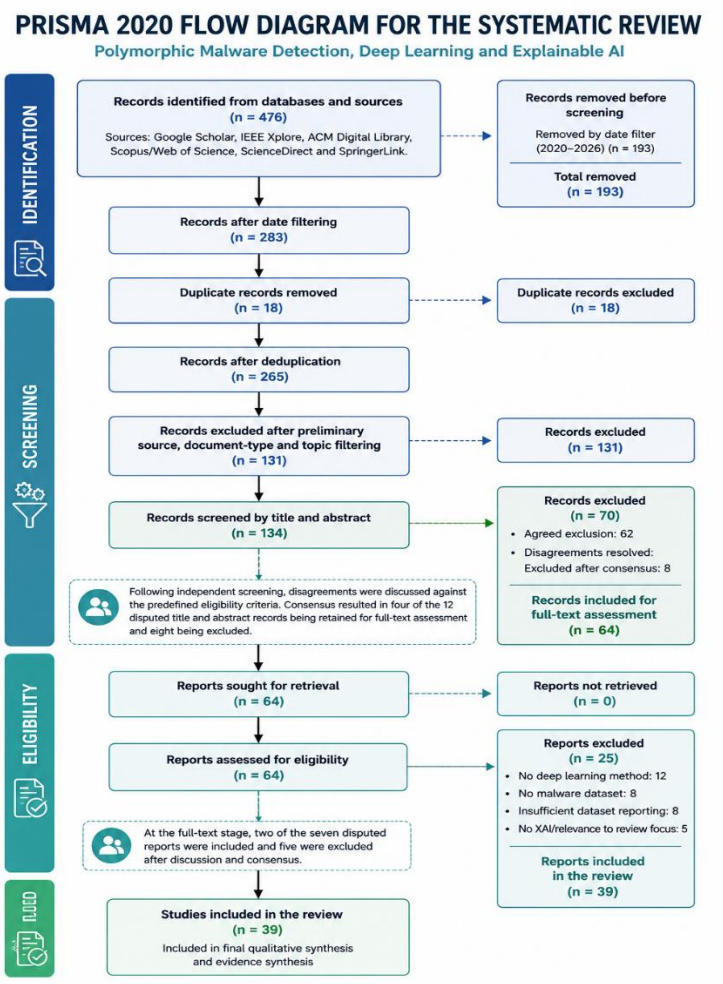

Figure 1: PRISMA 2020-style Flow Diagram Based on the Consolidated Multi-Database Search Log

Reviewer Agreement and Screening Reliability

To improve the reliability of the screening process, two co-authors independently verified the title/abstract screening decisions and the full-text eligibility decisions using the predefined inclusion and exclusion criteria. Screening decisions were coded as include, exclude or uncertain. Disagreements were discussed against the eligibility criteria and resolved by consensus; where consensus could not be reached, a third author was available for adjudication. Inter-rater agreement was assessed using Cohen's Kappa coefficient. At the title and abstract screening stage, the reviewers assessed 134 records, agreed on 122 records and disagreed on 12 records, producing a Cohen's Kappa value of 0.82. At the full-text eligibility stage, the reviewers assessed 64 papers, agreed on 57 papers and disagreed on 7 papers, producing a Cohen's Kappa value of 0.77. These values indicate strong agreement at title/abstract screening and substantial agreement at full-text eligibility assessment. Following independent verification, consensus resulted in four of the 12 disputed title and abstract records being retained for full-text assessment and eight being excluded. At the full-text stage, two of the seven disputed reports were included and five were excluded. The final consensus decisions were used to construct the PRISMA flow diagram and determine the studies included in the review.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

Table 4: Reviewer Agreement during Study Selection

Screening Stage	Records Assessed	Agreements	Disagreements	Percentage Agreement	Cohen's κ	95% CI	Interpretation	Resolution Procedure
Title/abstract screening	134	122	12	91.0%	0.82	0.72–0.92	Strong agreement	Disagreements resolved through discussion and consensus
Full-text eligibility	64	57	7	89.1%	0.77	0.60–0.93	Substantial agreement	Disagreements resolved through discussion and consensus

D. Quality Appraisal

Quality appraisal is essential because malware detection papers often report high accuracy without sufficient evidence of robust generalisation. Each study was assessed against six criteria: clarity of problem definition, dataset transparency, feature representation, evaluation design, robustness testing and interpretability. A simple three-level scoring scheme can be used: 2 = clearly addressed, 1 = partially addressed, and 0 = not addressed. The appraisal showed that dataset transparency and evaluation design were generally better reported than robustness testing and interpretability. Studies with weak scores may still be discussed if they are foundational, but their limitations have been made explicit.

Table 5: Study Quality Appraisal Checklist

Quality criterion	What was assessed
Problem clarity	Whether the study clearly distinguishes general malware detection from polymorphic, metamorphic or obfuscation-resilient detection.
Dataset transparency	Whether the dataset source, label definition, class balance, temporal coverage and preprocessing are described.
Feature representation	Whether the feature modality is appropriate and clearly linked to the malware behavior being detected.
Evaluation design	Whether metrics, baselines, validation splits and leakage controls are reported.
Robustness	Whether cross-dataset testing, temporal testing, adversarial testing or obfuscation robustness is evaluated.
Interpretability	Whether explanations, ablation or feature-contribution analysis are included and interpreted.

Source: Author's quality appraisal framework adapted for malware-detection studies.

III. RESULTS AND THEMATIC SYNTHESIS

A. Traditional Malware Detection and Its Limits

Traditional malware detection includes signature matching, heuristic analysis, static analysis, dynamic analysis and hybrid analysis (Aboaoja et al., 2022; Ucci et al., 2019; Ye et al., 2017). Signature-based methods are fast and reliable for known malware but perform poorly when variants mutate (Brezinski & Ferens, 2023; You & Yim, 2010). Static analysis is safe and scalable, but packing, encryption, junk-code insertion and instruction substitution can distort static features (Pan et al., 2020). Dynamic analysis captures runtime behaviour such as API calls, file access, registry changes and process injection, but it is computationally expensive and may be evaded through sandbox detection or delayed execution

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

(Kolosnjaji et al., 2016; Maniriho et al., 2023). Hybrid analysis combines static and dynamic evidence, yet many hybrid studies still use shallow fusion or limited feature groups (Gibert et al., 2020).

For polymorphic malware, these limitations are especially important. A detector based only on byte signatures may fail when the decryptor changes. A detector based only on static metadata may miss runtime intent. A detector based only on API behaviour may be affected by injected noise or incomplete execution traces. The literature therefore supports the use of multimodal learning, but it also shows that multimodal design must be validated rather than assumed to be effective (Bensaoud et al., 2024; Brezinski & Ferens, 2023; Maniriho et al., 2023; Mauri & Damiani, 2025).

B. Machine Learning-Based Malware Detection

Classical machine learning models such as Random Forest, Support Vector Machine, Logistic Regression, Naive Bayes, XGBoost, LightGBM and AdaBoost have been widely used for malware detection (Chen & Ren, 2023; Kolter & Maloof, 2006; Ucci et al., 2019). Their strengths include efficiency, compatibility with structured features and relative interpretability. Common inputs include PE headers, imports, sections, entropy, strings, byte histograms, opcode frequencies and API frequencies (Anderson & Roth, 2018; Chen & Ren, 2023). These models remain relevant because structured metadata can reveal meaningful properties such as packing, abnormal entropy, unusual sections and suspicious imports. However, machine learning approaches often depend on handcrafted features. Polymorphic malware can deliberately manipulate those features, creating brittleness. The literature also warns that high performance may be inflated by leakage, weak baselines, temporal bias, duplicate samples or unrealistic evaluation. Therefore, classical machine learning provides useful baselines and structured-feature branches, but it does not remove the need for deep multimodal representation and careful validation (Brezinski & Ferens, 2023; Demetrio et al., 2021; Arp et al., 2022).

C. Deep Learning and Hybrid Architectures

Deep learning is increasingly used in malware detection because it can learn complex representations from high-dimensional data (Bensaoud et al., 2024). CNNs are particularly suitable for byte streams and binary images. Nataraj et al. (2011) demonstrated image-based malware classification, Raff et al. (2018) examined end-to-end learning from executable bytes, and Vasan et al. (2020) extended image-based analysis through fine-tuned convolutional networks. Recurrent architectures are better suited to ordered behavioral and instruction sequences. Kolosnjaji et al. (2016) used recurrent learning for malware-system-call analysis, while Maniriho et al. (2023) combined CNN and BiGRU components for API-call detection. Transformer models can capture longer-range dependencies but generally require larger datasets and greater computational resources (Zhang et al., 2023). Hybrid architectures combine these strengths and are therefore attractive for malware that changes across several representations (Gibert et al., 2020; Owoh et al., 2024). The reviewed literature suggests that hybrid CNN-LSTM, CNN-GRU, CNN-BiGRU, CNN-Transformer and multimodal dense-fusion models often outperform unimodal baselines. Nevertheless, many studies remain Android-focused, API-only, image-only or family-classification oriented. Few studies explicitly model polymorphic/non-polymorphic labels across structural, behavioural, opcode and spatial modalities. This creates a gap for a hybrid framework that combines CNN image analysis, LSTM API modelling, GRU opcode modelling and dense structured-feature learning.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

D. Feature Modalities for Polymorphic Malware Detection

Feature modality determines the type of malware evidence available to a detection model. Structural features include entropy, file size, PE headers, imports, sections, strings and packing indicators. EMBER illustrates how these static characteristics can support Windows PE malware detection (Anderson & Roth, 2018). API-call sequences represent interactions with the operating system and provide evidence of runtime behavior. Their value has been demonstrated in sequence-based frameworks such as API-MalDetect and GRU–GAN (Maniriho et al., 2023; Owoh et al., 2024). Opcode sequences provide lower-level information about instruction patterns, control flow and code transformation. Binary-image approaches instead convert byte values into spatial representations that CNNs can process, as demonstrated by Nataraj et al. (2011) and Vasan et al. (2020). No single modality is sufficient for polymorphic malware detection. Structural characteristics can be modified through packing and obfuscation, while API traces may be affected by noise insertion, incomplete execution or sandbox evasion. Opcode sequences can be changed through instruction substitution and dead-code insertion, and binary images may capture structure without representing runtime intent. Multimodal fusion addresses some of these limitations by reducing dependence on one fragile representation (Gibert et al., 2020). Nevertheless, the design must be evaluated carefully because combining modalities does not automatically guarantee transformation-aware detection (Brezinski & Ferens, 2023).

Table 6. Feature modalities used in polymorphic malware detection.

Modality	Typical features	Strength for polymorphic detection	Main limitation
Structural/tabular	Entropy, file size, PE headers, imports, sections, packing indicators	Fast, interpretable, useful for detecting packing and abnormal structure.	Can be manipulated and may produce shortcuts if over-dominant.
API behavior	Ordered API calls, system calls, file/registry/process/network actions	Captures runtime intent that may survive static mutation.	Affected by sandbox evasion, API noise and incomplete traces.
Opcode sequence	Instruction tokens, opcode n-grams, execution-flow patterns	Useful for low-level code transformation and obfuscation analysis.	Sensitive to disassembly errors, packing and instruction substitution.
Binary image	Byte-to-pixel grayscale images, image textures, CNN feature maps	Captures spatial structure and packed/obfuscated texture patterns.	May miss behavioral intent and can be altered by binary rewriting.
Multimodal fusion	Combination of structured, API, opcode and image features	Reduces reliance on one fragile representation.	Requires careful dataset alignment, ablation and interpretability.

Source: Author's synthesis from Anderson and Roth (2018), Catak et al. (2021), Maniriho et al (2023), Nataraj et al (2011), and Vasan et al (2020).

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

E. Datasets and Benchmarking

Dataset quality remains one of the most important weaknesses in malware-detection research. EMBER provides a widely used benchmark for static Windows PE malware detection (Anderson & Roth, 2018), while SOREL-20M supports large-scale malware learning using extracted PE features and metadata (Harang & Rudd, 2020). BODMAS was developed to support temporal analysis and concept-drift research (Yang et al., 2021). Other datasets serve different experimental purposes. Malimg supports image-based malware-family classification (Nataraj et al., 2011), while Microsoft BIG 2015 provides byte and assembly files for Windows malware-family analysis (Ronen et al., 2018). Android datasets support mobile-malware research but are not directly transferable to Windows PE polymorphism (Mahdavifar et al., 2020). Although these resources have enabled substantial progress, they rarely provide controlled parent–variant lineage, explicit polymorphic labels, transformation metadata and aligned structural, behavioral, opcode and image features. A rigorous polymorphic-malware dataset should contain explicit class labels, base-sample-to-variant lineage, transformation metadata, multiple feature modalities and leakage-aware validation. Family-aware or temporal splits are particularly important because random splitting can produce overly optimistic performance estimates (Pendlebury et al., 2019). Dataset design should also ensure that individual features do not make the classification problem trivial. Arp et al. (2022) warn that machine-learning systems in security may exploit unintended shortcuts, while Jia et al. (2023) emphasize the importance of evaluating robustness under meaningful malware transformations.

Table 7: Dataset and Benchmarking Resources Relevant to Malware Detection

Dataset/resource	Main contribution	Limitation for polymorphic malware detection
EMBER	Static PE feature benchmark with byte histograms, entropy histograms, strings, headers, sections and imports.	Static malicious/benign task; no controlled polymorphic lineage or ordered behavioral traces.
SOREL-20M	Large-scale PE malware benchmark with extracted features and metadata.	Designed for large-scale malicious PE detection rather than explicit polymorphic labels.
BODMAS	PE malware dataset useful for temporal analysis and drift-aware studies.	Temporal evolution is not the same as controlled polymorphic transformation.
Malimg	Binary-to-image malware classification dataset.	Image/family focus; lacks API, opcode and structured multimodal data.
Microsoft BIG 2015	Byte and assembly files for malware family classification.	Family classification target; no polymorphic/non-polymorphic ground truth.
AndroZoo/CIC Android datasets	Large-scale Android malware resources with static/dynamic features.	Platform-specific and not directly suited to Windows PE polymorphism.
Obfuscation robustness datasets	Useful for evaluating detector response to obfuscation.	May not include full multimodal polymorphic representation and lineage.

Source: Author's synthesis from Anderson and Roth (2018), Harang and Rudd (2020), Nataraj et al (2011), Ronen et al (2018), and Yang et al (2021).

F. Explainable Artificial Intelligence and Ablation

XAI is necessary in malware detection because analysts require defensible reasons for model decisions (Galli et al., 2024). Alani and Awad (2022) demonstrated its use in lightweight Android malware detection, while Alani et al. (2023) applied SHAP to obfuscated-malware classification. Vuran Sarı and Acı (2025) combined SHAP and LIME with a hybrid CNN–GRU model. Different explanation methods serve different purposes. SHAP supports global and local feature attribution (Lundberg & Lee, 2017), while LIME explains individual predictions through interpretable local approximations (Ribeiro et al., 2016). Integrated gradients attributes neural-network predictions to input features (Sundararajan et al., 2017), whereas Grad-CAM highlights influential regions in CNN-based image models (Selvaraju et al.,

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

2017). Ablation is equally important. If removing the image branch, API branch, opcode branch or structured branch does not affect performance, the proposed hybrid design may not be justified. Conversely, if each branch contributes complementary performance or stability, the architecture is more defensible. Ablation also helps detect dataset shortcuts: if one feature group alone nearly matches the full model, the dataset may be too easy or leakage-prone. For this reason, ablation and XAI should be treated as part of the evaluation design, not as optional post-processing (Arp et al., 2022; Gibert et al., 2020).

Table 8: Explainability and Ablation Methods for Malware Detection Models.

Explanation method	Best suited input	Use in malware detection	Caution
Permutation importance	Structured features	Global sensitivity of entropy, imports, section counts and indicators.	Correlated features can distort ranking.
Integrated gradients	Differentiable neural models with tabular or embedded inputs	Local attribution for structured features or embeddings.	Baseline choice affects results.
Saliency/Grad-CAM	Binary-image CNN branch	Highlights image regions contributing to prediction.	Visual relevance does not prove causal malware behavior.
Token occlusion	API and opcode sequences	Tests importance of specific calls or tokens by removal/masking.	Perturbations must remain meaningful.
Branch ablation	Multimodal architectures	Quantifies modality-level contribution.	Must use the same splits and training settings for fair comparison.

Source: Author's evaluation metrics, and explainability method, and ablation evidence synthesis from Galli et al. (2024), Lundberg and Lee (2017), Ribeiro et al. (2016), Selvaraju et al. (2017), and Sundararajan et al. (2017).

G. Comparative Review of Representative Studies

The table below summarises representative literature by methodological category. The purpose is not to list every available malware detection study but to identify how existing work supports or limits the development of robust polymorphic malware detection (Alsharif, 2025; Bensaoud et al., 2024; Mauri & Damiani, 2025; Zhang et al., 2023).

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

Table 9: Comparative Review of Representative Studies

Category	Representative studies	Strengths	Remaining limitation
Traditional and survey literature	Aboaoja et al.; Ucci et al.; Ye et al.; Pan et al.	Provide taxonomies of static, dynamic, hybrid, ML and DL malware detection.	Survey-level or general malware focus; limited controlled polymorphic modelling.
Classical ML	Kolter and Maloof; Chen and Ren; Ucci et al.	Demonstrate usefulness of PE metadata, feature fusion and ensemble classifiers.	Depend heavily on handcrafted static or tabular features.
API sequence DL	Kolosnjaji et al.; Maniriho et al.; Owoh et al.	Show effectiveness of API-call sequence learning with CNN, BiGRU or GRU variants.	Often behavioral/API-centered; limited structural/image/opcode fusion.
Image and byte DL	Nataraj et al.; Raff et al.; Vasan et al.; Krcal et al.; Catak et al.	Support binary image and raw-byte representation for CNN-style learning.	May miss runtime behavior and polymorphic intent.
Dataset studies	Anderson and Roth; Harang and Rudd; Yang et al.; Ronen et al.; Mahdavifar et al.	Provide important benchmarks for PE, image, temporal and family classification.	Not designed as full polymorphic multimodal benchmarks.
XAI studies	Ribeiro et al.; Lundberg and Lee; Sundararajan et al.; Selvaraju et al.; Galli et al.	Provide tools and frameworks for local/global interpretability.	Often single-modality or vulnerable to explanation instability.
Polymorphism and obfuscation-focused work	Alrzini and Pennington; Nawaz et al.; Jia et al.; Alsharif; Mauri and Damiani.	Directly address polymorphic, metamorphic or obfuscation-resilient detection.	Fragmented across behavior, ransomware, network traffic, memory or ensemble learning.

Source: Author's synthesis of representative studies cited in the review.

H. Quantitative Comparative Synthesis of Representative Studies

To address the quantitative dimension of the review, representative empirical studies were compared by dataset, feature modality, model architecture, reported performance and interpretability support. A formal meta-analysis was not conducted because the studies used heterogeneous datasets, platforms, feature spaces, class definitions and evaluation splits. Therefore, the values in Table 10 should be interpreted as author-reported descriptive results rather than directly pooled effect sizes.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

Table 10: Quantitative Comparison of Representative Malware Detection Studies

Study	Dataset/platform	Feature modality	Model/algorithm	Reported quantitative result
Chen and Ren (2023)	Microsoft BIG 2015 Windows malware-family dataset	Static binary and assembly features with multi-feature fusion	XGBoost, LightGBM, CatBoost and other ML baselines	XGBoost achieved 99.87% accuracy and 0.007 log loss; LightGBM and CatBoost exceeded 99.7% accuracy.
Vasan et al (2020)	Malimg dataset; IoT-Android mobile dataset	Colour malware images generated from raw binaries	Fine-tuned CNN architecture (IMCFN)	98.82% accuracy on Malimg and more than 97.35% accuracy on the IoT-Android dataset.
Vuran Sari and Aci (2025)	Tailored static malware dataset using API-call and DLL information	API calls and DLL static-analysis features	CNN-GRU-3, CNN, LSTM, GRU, XGB, ETC, KNN and RF	XGB achieved 99.81% accuracy; CNN-GRU-3 achieved 99.37% accuracy among deep-learning models.
Mauri and Damiani (2025)	Real malware system-call logs with thread-level polymorphic behavior	Behavioral system-call sequences assigned to execution threads	Polymorphism-aware ensemble with minority-report voting	Baseline model accuracy could fall to 50% under thread randomization; ensemble reached 99.7% best-case detection capability.
Owoh et al (2024)	Multiple Windows PE API-call sequence datasets	Dynamic API-call sequences	GRU-GAN compared with BiLSTM and BiGRU	Hybrid GRU-GAN achieved 98.9% accuracy on the most challenging dataset and improved resource utilization.
Maniriho et al (2023)	Windows API-call sequence benchmark datasets	Raw and long ordered API-call sequences	NLP encoder with CNN-BiGRU feature extractor	Reported performance across accuracy, precision, recall, F1-score and AUC-ROC, with API-MalDetect outperforming benchmark techniques across datasets.
Catak et al. (2021)	Malware image dataset with augmented samples	Three-channel malware images with additive noise augmentation	CNN-based malware family classifier	Conventional CNN class-level F1 ranged from 0.00 to 0.80 before augmentation; augmentation improved malware-family classification performance.
Gibert et al (2020)	Portable Executable malware classification datasets	Windows API calls, assembly mnemonics and byte sequences	HYDRA multimodal deep-learning framework	Evaluated using 10-fold cross-validation; multimodal fusion improved classification relative to narrower feature views.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

Source: Author's synthesis of representative studies included in the review. Reported values are taken from the original studies and are not directly pooled because the datasets, platforms, feature spaces and evaluation protocols differ.

Table 11. Interpretability Support and Critical Limitations of Representative Studies

Study group	XAI or ablation support	Critical limitation for polymorphic malware detection
Static feature-fusion ML studies	Usually limited; some models are interpretable through feature importance, but explanations are not always domain-validated.	Very high accuracy may reflect family-specific or dataset-specific artefacts rather than transformation-aware detection.
Image-based CNN studies	Saliency or image-region explanation is often absent; when present, visual relevance is difficult to connect to malware behavior.	Image texture can reveal packing or structure but may miss runtime intent and can be affected by binary rewriting.
API-sequence deep-learning studies	Some studies report important API calls or local explanations, but explanation stability is rarely tested.	Behavioral traces can be incomplete, sandbox-dependent or vulnerable to delayed execution and API-noise injection.
Hybrid multimodal studies	Ablation is essential but not consistently reported across all modalities.	Fusion can improve coverage, but without branch-level contribution analysis the model may still depend on one dominant shortcut modality.
Polymorphism-focused behavioral studies	Often stronger on robustness testing than explanation; ensemble logic may not expose fine-grained evidence.	Some studies model specific polymorphic behaviors such as thread randomization but do not provide full multimodal polymorphic lineage.
XAI-focused malware studies	Use SHAP, LIME, integrated gradients, Grad-CAM, saliency maps or token occlusion.	Post-hoc explanations can be unstable or manipulable; they should be interpreted with ablation, robustness testing and malware-domain knowledge.

The comparison confirms that reported performance in malware detection is often high, but the evidence is not directly comparable across studies. The strongest reported scores usually come from controlled datasets or family-classification settings, while fewer studies test whether the model remains reliable under explicit polymorphic transformation. This distinction is important because a model that performs well on malware-family classification may not necessarily detect polymorphic behavior. The critical weakness across the literature is therefore not the absence of high accuracy, but the limited combination of polymorphic ground truth, multimodal feature coverage, leakage-aware validation, branch-wise ablation and stable explainability.

I. Research Gaps

The synthesis identifies five major research gaps. First, many studies continue to address general malware classification rather than explicit polymorphic/non-polymorphic detection. A model that detects malicious software does not necessarily identify the transformations that make malware polymorphic (Mauri & Damiani, 2025). Nawaz et al. (2022) similarly show that polymorphism-focused detection requires more specific evaluation than conventional malware classification. Second, existing datasets are rarely designed around controlled polymorphic behavior. Widely used benchmarks provide valuable malware samples and features but generally lack parent–variant lineage and transformation metadata (Anderson & Roth, 2018). Jia et al. (2023) further demonstrate the importance of testing models under

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

deliberate obfuscation or functionality-preserving modification. Third, hybrid deep-learning studies often combine only a limited number of feature modalities. HYDRA demonstrates the benefits of multimodal learning (Gibert et al., 2020), while API-MalDetect shows the effectiveness of deep sequence modelling (Maniriho et al., 2023). However, few studies integrate structural, behavioral, opcode and spatial evidence within one transformation-aware framework. Fourth, evaluation practices remain inconsistent. Accuracy, precision, recall and F1-score are commonly reported, but cross-dataset validation, temporal splitting, family-aware evaluation and leakage control remain less frequent (Arp et al., 2022; Pendlebury et al., 2019). Fifth, interpretability remains underdeveloped. Although XAI is increasingly used in malware research, explanations are often applied after training without stability analysis or domain validation (Galli et al., 2024). Post-hoc explanations may also be manipulated, which limits their reliability when used without complementary robustness testing (Slack et al., 2020).

Table 12. Consolidated Research Gaps and Recommended Responses

Gap	Why it matters	Recommended response
Insufficient polymorphic ground truth	General malware labels do not prove transformation-aware detection.	Construct datasets with explicit polymorphic/non-polymorphic labels and transformation metadata.
Weak multimodal coverage	Polymorphism affects structure, behavior, opcodes and binary layout.	Fuse API, opcode, structured and image features in a single architecture.
Limited ablation	Hybrid models may rely on one dominant shortcut feature.	Report branch-wise ablation and feature-level contribution.
Limited robustness testing	High benchmark accuracy may not generalize to new families or time periods.	Use temporal, family-aware, cross-dataset and obfuscation-resilience tests.
Unstable or shallow XAI	Explanations can be misleading or manipulated.	Use multiple explanation methods and interpret them with malware-domain knowledge.

Source: Author's synthesis of the review findings.

J. Synthesis of Findings According to Research Questions

To make the review findings more explicit and directly aligned with the stated research questions, Table 13 maps each research question to the evidence synthesised in the review, the direct answer emerging from the literature and the implication for future polymorphic malware detection research.

Table 13: Mapping of Research Questions to Synthesized Findings and Implications

Research question	Evidence synthesized	Direct answer from the review	Implication
RQ1: What detection approaches have been used for polymorphic, metamorphic and obfuscation-resilient malware detection?	Traditional signature, static, dynamic and hybrid analysis; classical ML; CNN, LSTM, GRU, Transformer and hybrid deep learning; ensemble and obfuscation-focused approaches.	The literature has moved from signature-dependent and handcrafted-feature approaches toward deep and hybrid learning. However, many studies still address general malware classification rather than explicit polymorphic transformation detection.	Future work should combine complementary detection views and validate whether the model detects transformation-aware evidence rather than dataset shortcuts.
RQ2: Which feature modalities are most frequently used,	Structured PE metadata, entropy, imports and sections; API/system-call sequences; opcode	Each modality captures a different malware view. Structural features are fast and interpretable, API and opcode	A robust detector should fuse structured, behavioral, opcode and image features while

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

and what are their strengths and limitations?	sequences; byte streams; binary-image representations; multimodal fusion.	sequences capture behavioral and instruction-level evidence, and binary images capture spatial texture. No single modality is sufficient against polymorphic change.	using ablation to test whether each modality contributes useful evidence.
RQ3: How do existing datasets support or limit rigorous evaluation of polymorphic malware detection?	EMBER, SOREL-20M, BODMAS, Malimg, Microsoft BIG 2015, Android datasets and obfuscation robustness datasets.	Existing datasets are valuable for static PE detection, temporal analysis, image/family classification and Android malware research, but they rarely provide explicit polymorphic/non-polymorphic labels, transformation metadata or base-sample-to-variant lineage across multiple modalities.	Dataset construction should include lineage, transformation history, leakage checks, family-aware or temporal splits and multimodal alignment.
RQ4: How are explainability and ablation used to interpret malware detection models?	SHAP, LIME, permutation importance, integrated gradients, Grad-CAM, saliency mapping, token occlusion and branch-level ablation.	XAI is increasingly used, but often as post-hoc interpretation rather than as part of the evaluation design. Ablation is also underreported, making it difficult to know whether hybrid models truly benefit from all branches.	Interpretability should combine global, local, token-level, image-region and branch-ablation evidence, and explanations should be checked for stability and malware-domain meaning.
RQ5: What research gaps remain for robust and interpretable hybrid deep learning detection?	Evidence from dataset comparison, feature-modality analysis, representative model comparison, XAI studies and robustness literature.	The major gaps are insufficient polymorphic ground truth, limited multimodal coverage, weak ablation, limited robustness testing and shallow or unstable XAI.	The recommended agenda is a polymorphism-aware, multimodal CNN-LSTM-GRU-dense framework supported by rigorous dataset validation, robustness testing, ablation and multi-method XAI.

The mapping shows that the central contribution of the review is not merely the identification of hybrid deep learning as a promising direction. Rather, the evidence indicates that future claims about polymorphic malware detection must be supported by explicit polymorphic ground truth, multimodal feature alignment, robust evaluation and interpretable evidence of feature contribution.

K. Proposed Research Agenda

Based on the findings of this review, a future framework for polymorphic malware detection should combine four elements. The first is a hybrid multimodal model in which each branch is aligned with a specific feature view: CNN for binary images, LSTM for API behavior, GRU for opcode sequences and dense layers for structured metadata. Multimodal malware research provides evidence that combining complementary representations can improve detection coverage (Bensaoud et al., 2024). The second element is a dataset-construction protocol that explicitly records polymorphic transformations, variant lineage and family relationships. Such information is necessary to distinguish transformation-aware

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

learning from general malware classification (Jia et al., 2023). The third element is rigorous evaluation through multiple performance metrics, branch-wise ablation, and family-aware testing and cross-dataset validation. The fourth element is interpretability through feature importance, token attribution, image saliency and modality-level ablation. These components are integrated in the conceptual framework presented in Figure 2. The proposed framework begins with polymorphism-aware dataset construction and then aligns each feature modality with a specialized model branch. Structured PE metadata are processed through dense layers, API-call sequences through an LSTM branch, opcode sequences through a GRU branch, and byte-level binary images through a CNN branch. The learned representations are fused for classification, while explainability and ablation components are used to validate whether the decision is supported by meaningful malware evidence rather than by dataset shortcuts.

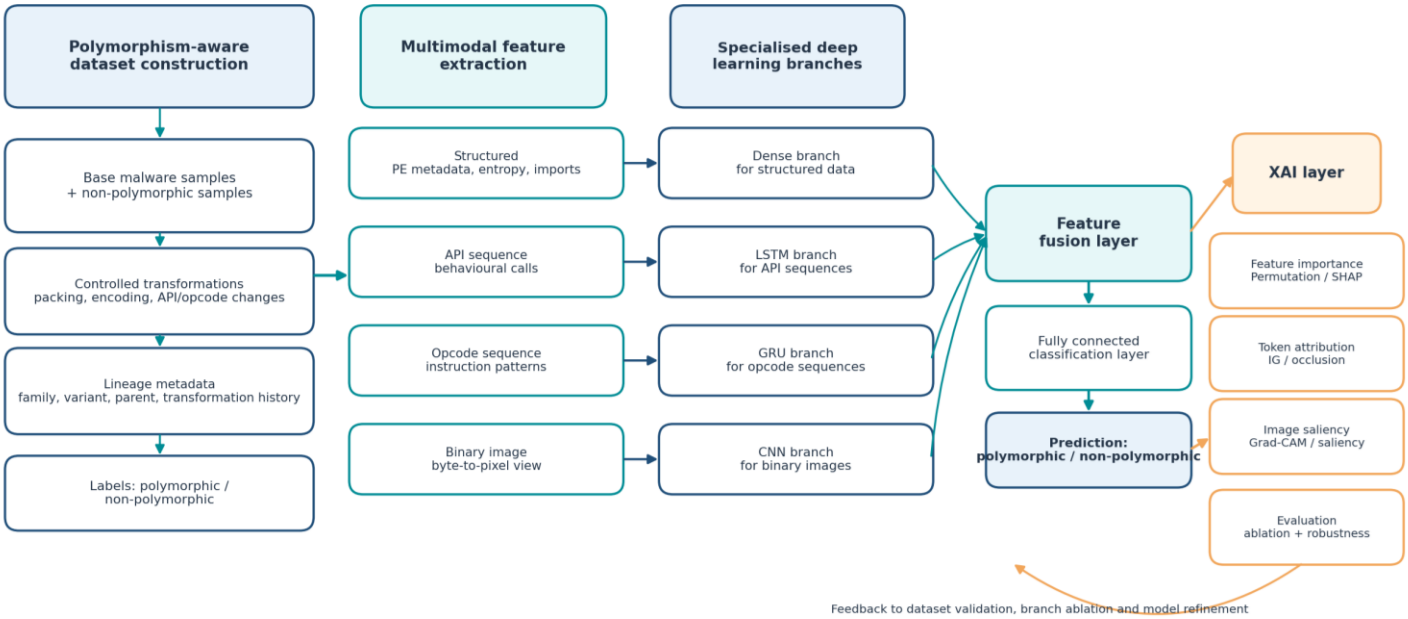

Figure 2: Conceptual Framework for the Proposed Hybrid CNN-LSTM-GRU-XAI Polymorphic Malware Detection Architecture.

L. Implications for Research and Practice

For researchers, the findings indicate that future polymorphic-malware studies should prioritise dataset validity, multimodal fusion and interpretable evaluation. Claims of transformation-aware detection should not be based solely on general malware datasets or random train–test splits. Leakage controls, family-aware validation and transparent evaluation are essential for credible security machine learning (Arp et al., 2022; Pendlebury et al., 2019). Researchers should also avoid relying on a single metric or explanation method. For practitioners, operational detection systems should not depend on one representation. Static indicators are efficient but can be modified, while behavioral traces may be incomplete or influenced by execution environments. Multimodal systems can provide broader evidence, but deployment must account for inference cost, memory consumption, false-positive burden, retraining and analyst feedback (Maniriho et al., 2023). Explanations should also be treated as decision-support evidence rather than unquestionable proof (Galli et al., 2024).

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21777609

M. Limitations of the Review

This review is limited by the availability and transparency of source studies. Malware datasets are often restricted because of safety, legal and licensing concerns, and some studies do not release code or samples. Although a multi-database search strategy and reviewer-agreement process were used, the review remains dependent on the quality of information reported in the included studies. In several cases, studies reported high classification performance without sufficient detail on sample deduplication, temporal splitting, family-aware evaluation, feature leakage control, cross-dataset validation or explanation stability. The quantitative comparison should therefore be interpreted as a structured comparative synthesis rather than a meta-analysis, because the included studies used different datasets, platforms, feature representations, evaluation settings and performance metrics.

IV. Conclusion

The systematic synthesis shows that polymorphic malware detection requires more than high benchmark accuracy. Polymorphic malware alters its observable representation, making signature-dependent and single-modality approaches fragile. Although deep-learning research has advanced through CNN, LSTM, GRU and hybrid architectures, many studies still lack explicit polymorphic ground truth, multimodal coverage, branch-level ablation and robust explainability. Existing benchmarks remain valuable, but they do not fully support controlled transformations across structural, behavioral, opcode and spatial modalities. The review therefore supports a research direction centered on hybrid multimodal learning, polymorphism-aware dataset construction, rigorous evaluation and multi-method explainability. A credible detector should integrate structured metadata, API-call sequences, opcode sequences and binary-image features while demonstrating, through ablation and XAI, which evidence drives its predictions. Such work would strengthen both the scientific validity of polymorphic-malware research and the operational trustworthiness of malware-detection systems.

References

[1] Aboaoja, F. A., Zainal, A., Ghaleb, F. A., Al-Rimy, B. A. S., Eisa, T. A. E., & Elnour, A. A. H. (2022). Malware detection issues, challenges, and future directions: A survey. Applied Sciences, 12(17), Article 8482. https://doi.org/10.3390/app12178482

[2] Alani, M. M., & Awad, A. I. (2022). PAIRED: An explainable lightweight Android malware detection system. Journal of Information Security and Applications, 67, Article 103188. https://doi.org/10.1016/j.jisa.2022.103188

[3] Alani, M. M., Mashatan, A., & Miri, A. (2023). XMal: A lightweight memory-based explainable obfuscated-malware detector. Computers & Security, 133, Article 103409. https://doi.org/10.1016/j.cose.2023.103409

[4] Alrzini, J. R. S., & Pennington, D. (2020). A review of polymorphic malware detection techniques. International Journal of Advanced Research in Engineering and Technology, 11(12), 1238-1247. https://doi.org/10.34218/IJARET.11.12.2020.119

[5] Alsharif, N. (2025). A hybrid ensemble deep learning approach for detecting polymorphic malware. In 2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC). IEEE. https://doi.org/10.1109/ICAISC64594.2025.10959351

[6] Anderson, H. S., & Roth, P. (2018). EMBER: An open dataset for training static PE malware machine learning models. arXiv. https://doi.org/10.48550/arXiv.1804.04637

[7] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In Proceedings of the 31st USENIX Security Symposium (pp. 3971-3988). USENIX Association.

[8] Bensaoud, A., Kalita, J., & Bensaoud, M. (2024). A survey of malware detection using deep learning. Machine Learning with Applications, 16, Article 100546. https://doi.org/10.1016/j.mlwa.2024.100546
[9] Brezinski, K., & Ferens, K. (2023). Metamorphic malware and obfuscation: A survey of techniques, variants, and generation kits. Security and Communication Networks, 2023, Article 8227751. https://doi.org/10.1155/2023/8227751
[10] Catak, F. O., Ahmed, J., Sahinbas, K., & Khand, Z. H. (2021). Data augmentation based malware detection using convolutional neural networks. PeerJ Computer Science, 7, Article e346. https://doi.org/10.7717/peerj-cs.346
[11] Chen, Z., & Ren, F. (2023). An efficient boosting-based Windows malware family classification system using multi-features fusion. Applied Sciences, 13(6), Article 4060. https://doi.org/10.3390/app13064060
[12] Cohen, F. (1987). Computer viruses: Theory and experiments. Computers & Security, 6(1), 22-35. https://doi.org/10.1016/0167-4048 (87)90122-2
[13] Demetrio, L., Biggio, B., Lagorio, G., Roli, F., & Armando, A. (2021). Functionality-preserving evasion attacks against Windows PE malware detectors. Computers & Security, 110, Article 102402. https://doi.org/10.1016/j.cose.2021.102402
[14] Galli, A., La Gatta, V., Moscato, V., Postiglione, M., & Sperli, G. (2024). Explainability in AI-based behavioral malware detection systems. Computers & Security, 141, Article 103842. https://doi.org/10.1016/j.cose.2024.103842
[15] Gibert, D., Mateu, C., & Planes, J. (2020). HYDRA: A multimodal deep learning framework for malware classification. Computers & Security, 95, Article 101873. https://doi.org/10.1016/j.cose.2020.101873
[16] Harang, R., & Rudd, E. M. (2020). SOREL-20M: A large scale benchmark dataset for malicious PE detection. arXiv. https://doi.org/10.48550/arXiv.2012.07634
[17] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
[18] Jia, L., Yang, Y., Tang, B., & Jiang, Z. (2023). ERMDS: An obfuscation dataset for evaluating robustness of learning-based malware detection system. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 3(1), Article 100106. https://doi.org/10.1016/j.tbench.2023.100106
[19] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). Deep learning for classification of malware system call sequences. In Australasian Joint Conference on Artificial Intelligence (pp. 137-149). Springer. https://doi.org/10.1007/978-3-319-50127-7_11
[20] Kolter, J. Z., & Maloof, M. A. (2006). Learning to detect and classify malicious executables in the wild. Journal of Machine Learning Research, 7, 2721-2744.
[21] Leder, F., Steinbock, B., & Martini, P. (2009). Classification and detection of metamorphic malware using value set analysis. In Proceedings of the 4th International Conference on Malicious and Unwanted Software (pp. 39-46). IEEE. https://doi.org/10.1109/MALWARE.2009.5403027
[22] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, 30, 4765-4774.
[23] Mahdavifar, S., Kadir, A. F. A., Fatemi, R., Alhadidi, D., & Ghorbani, A. A. (2020). Dynamic Android malware category classification using semi-supervised deep learning. In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (pp. 515-522). IEEE.
[24] Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2023). API-MalDetect: Automated malware detection framework for Windows based on API calls and deep learning techniques. Journal

of Network and Computer Applications, 218, Article 103704. https://doi.org/10.1016/j.jnca.2023.103704

[25] Mauri, L., & Damiani, E. (2025). Hardening behavioral classifiers against polymorphic malware: An ensemble approach based on minority report. Information Sciences, 689, Article 121499. https://doi.org/10.1016/j.ins.2024.121499

[26] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware images: Visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security (pp. 1-7). ACM. https://doi.org/10.1145/2016904.2016908

[27] Nawaz, S., Fournier-Viger, P., Nawaz, M. Z., Chen, G., Wu, Y., Wei, W., & Yu, P. S. (2022). MalSPM: Metamorphic malware behavior analysis using sequential pattern mining. IEEE Access, 10, 69952-69966.

[28] Owoh, N., Adejoh, J., Hosseinzadeh, S., Ashawa, M., Osamor, J., & Qureshi, A. (2024). Malware detection based on API call sequence analysis: A gated recurrent unit-generative adversarial network model approach. Future Internet, 16(10), Article 369. https://doi.org/10.3390/fi16100369

[29] Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hrobjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., & Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ, 372, Article n71. https://doi.org/10.1136/bmj.n71

[30] Pan, Y., Ge, X., Fang, C., & Fan, Y. (2020). A systematic literature review of Android malware detection using static analysis. IEEE Access, 8, 116363-116379. https://doi.org/10.1109/ACCESS.2020.3002842

[31] Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., & Cavallaro, L. (2019). TESSERACT: Eliminating experimental bias in malware classification across space and time. In Proceedings of the 28th USENIX Security Symposium (pp. 729-746). USENIX Association.

[32] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2018). Malware detection by eating a whole EXE. In Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security. https://doi.org/10.1609/aaai.v32i1.12102

[33] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144). ACM. https://doi.org/10.1145/2939672.2939778

[34] Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., & Ahmadi, M. (2018). Microsoft malware classification challenge. arXiv. https://doi.org/10.48550/arXiv.1802.10135

[35] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 618-626). https://doi.org/10.1109/ICCV.2017.74

[36] Slack, D., Hilgard, A., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling LIME and SHAP: Adversarial attacks on post-hoc explanation methods. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (pp. 180-186). ACM. https://doi.org/10.1145/3375627.3375830

[37] Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning (pp. 3319-3328). PMLR.

[38] Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, 81, 123-147. https://doi.org/10.1016/j.cose.2018.11.001

[39] Vasan, D., Alazab, M., Wassan, S., Naeem, H., Safaei, B., & Zheng, Q. (2020). IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. Computer Networks, 171, Article 107138. https://doi.org/10.1016/j.comnet.2020.107138

[40] Vuran Sari, N., & Aci, M. (2025). A hybrid CNN-GRU model with XAI-driven interpretability using LIME and SHAP for static analysis in malware detection. PeerJ Computer Science, 11, Article e3258. https://doi.org/10.7717/peerj-cs.3258

[41] Yang, L., Ciptadi, A., Laziuk, I., Ahmadzadeh, A., & Wang, G. (2021). BODMAS: An open dataset for learning based temporal analysis of PE malware. In 2021 IEEE Security and Privacy Workshops (SPW) (pp. 78-84). IEEE. https://doi.org/10.1109/SPW53761.2021.00019

[42] Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys, 50(3), Article 41. https://doi.org/10.1145/3073559

[43] You, I., & Yim, K. (2010). Malware obfuscation techniques: A brief survey. In 2010 International Conference on Broadband, Wireless Computing, Communication and Applications (pp. 297-300). IEEE. https://doi.org/10.1109/BWCCA.2010.85

[44] Zhang, J., Guo, Z., Luo, X., & Zhu, H. (2023). Deep learning for malware detection: A survey and research directions. ACM Computing Surveys, 55(9), Article 180. https://doi.org/10.1145/3569112