Faculty of Natural and Applied Sciences Journal of Computing and Applications
Print ISSN: 3026-8133
www.fnasjournals.com
Volume 3; Issue 1; March 2026; Page No. 87-92.
DOI: https://doi.org/10.63561/jca.v3i1.1206

Deep Learning-Based Malware Detection Under Real-World Constraints: A Systematic Review of Class Imbalance, Concept Drift and Interpretability

[1,2]Bodinga, A.B., [3]Garko, A.B., [4]Ibrahim, N.M., & [5]Gabi. D.
[1]Department of Computer Science, Abdullahi Fodio University of Science and Technology, Kebbi State, Nigeria
[2]Information and Communication Technology Department, Usmanu Danfodiyo University Teaching Hospital, Sokoto State, Nigeria
[3]Department of Computer Science, Federal University Dutse, Dutse Jigawa State Nigeria
[4]Department of Cyber Security, Nile University of Nigeria
[5]Department of Computer Science, Faculty of Computing, University of Technology, Malaysia, Skudai, Johor

***Corresponding author email**: belloabubakar@gmail.com

Abstract
Malware detection remains one of the most persistent and complex challenges in cybersecurity. The rapid evolution of attack techniques, fueled by the professionalization of cybercrime, continually outpaces traditional defenses. While deep learning has significantly enhanced detection capabilities, its real-world deployment is critically hampered by three interconnected and often overlooked challenges: extreme class imbalance, concept drift (including adversarial evolution), and the interpretability gap of black-box models. This Systematic Literature Review (SLR) synthesizes state-of-the-art research from 2014 to 2025 on malware detection across static, dynamic, and hybrid analysis methods, with a focused analysis on these three constraints. Following a PRISMA-guided methodology, this review analyzes 162 high-quality studies. It reveals that while research has progressed from foundational deep learning applications to advanced solutions like generative augmentation for imbalance, self-supervised test-time adaptation for drift, and integrated explainable AI (XAI) pipelines critical gaps persist. Our synthesis yields five key insights: (1) deep learning enhances accuracy but remains brittle under real-world data imbalance and adversarial drift; (2) current drift adaptation strategies, including recent federated and hybrid approaches, seldom holistically model adversarial intent; (3) GAN-based augmentation improves minority-class detection but lacks robust, security-focused evaluation of synthetic samples; (4) interpretability studies, despite recent integration efforts, remain fragmented and are rarely validated with human analysts to ensure actionable intelligence; and most critically, (5) no existing architecture jointly and seamlessly integrates continuous drift adaptation, dynamic imbalance correction, and operational interpretability. This review not only maps the evolution of these challenges but also crystallizes the pressing need for a unified framework. It provides the foundational justification for the proposed MAD-FIT (Malware Adaptive Detection with Fusion, Interpretation, and Training Dynamics) framework, which is designed to bridge these gaps and advance the field toward robust, adaptive, and trustworthy next-generation malware detection systems.

Keywords: Deep Learning, Malware, Real-World Constraints, Systematic Review, Class Imbalance

Introduction

The ongoing digital transformation has been accompanied by a marked increase in the sophistication, proliferation, and operational impact of malicious software (malware). The economic consequences are severe, with global cybercrime damages projected to reach trillions annually, largely driven by evolving malware threats (Cybersecurity

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Ventures, 2023). Traditional detection mechanisms: signature-based, heuristic, or simple behavioral methods; struggle against polymorphism, metamorphism, and adversarial evasion (Augello et al., 2025a)(Bayer et al., 2009; Ye et al., 2017). While deep learning provides adaptive capabilities (Raff et al., 2018; Saxe & Berlin, 2015), its deployment in real-world cybersecurity environments remains constrained by three fundamental and interconnected challenges: (1) extreme class imbalance in malware datasets, which biases models toward majority classes (He & Garcia, 2009; Pendlebury et al., 2019); (2) concept drift resulting from the natural evolution and adversarial manipulation of malware (Gama et al., 2004, 2014); and (3) the limited transparency and interpretability (Goodman & Flaxman, 2017; Ribeiro et al., 2016)inherent to complex deep learning architectures (Ofusori et al., 2025).

This systematic literature review (SLR) provides a comprehensive examination of research from 2014–2025 addressing these challenges. It reveals that while solutions for each problem exist in isolation, the field suffers from a critical lack of integrated frameworks (Berrios et al., 2025). Recent reviews confirm that efforts remain fragmented, with few studies proposing unified systems that are adaptive, robust to data skew, and explainable for security analysts (Almajed et al., 2025; Ofusori et al., 2025). By synthesizing the state-of-the-art and delineating unresolved gaps, this review motivates the development of the MAD-FIT (Malware Adaptive Detection with Fusion, Interpretation, and Training Dynamics) framework as a holistic solution. The operational need for such adaptive systems is underscored by the continuing professionalization of cyber threats, including the sustained growth of Malware-as-a-Service (MaaS) and sophisticated cross-platform toolkits observed (Patsakis et al., 2025), which lower the barrier to entry for adversaries and accelerate the evolution of malicious code (Patsakis et al., 2025).

Review Methodology (SLR Method)

This review adheres to the PRISMA 2020 framework (Page et al., 2021), which provides a standardized approach for ensuring transparency, reproducibility, and methodological rigor in systematic literature reviews.

Data Sources

The literature search encompassed seven major scientific repositories: IEEE Xplore, ACM Digital Library, ScienceDirect (Elsevier), SpringerLink, arXiv, Scopus, and Web of Science (Berrios et al., 2025; Ofusori et al., 2025). The inclusion of Scopus and Web of Science ensures access to high-quality, peer-reviewed literature with robust citation metrics, while arXiv provides early insights into cutting-edge, pre-print research (Berrios et al., 2025). Articles sourced from arXiv were subjected to additional quality assessment to ensure methodological rigor. This multi-source approach guarantees comprehensive coverage of the field from 2014 to 2025.

Search Query

A comprehensive Boolean search expression was formulated to ensure broad coverage of the target concepts. The following query string was applied across all databases:

("malware detection" OR "malicious software") AND
("deep learning" OR "neural networks" OR "machine learning") AND
("class imbalance" OR "imbalanced data" OR "long tail") AND
("concept drift" OR "data drift" OR "model degradation") AND
("interpretability" OR "explainability" OR "XAI").

Inclusion Criteria

Eligible studies satisfied all of the following conditions:

- Publication date within 2014–2025.
- Published in a peer-reviewed venue.
- Examined malware detection, classification, or related analytical tasks.
- Applied ML or DL methods as part of the proposed or evaluated framework.
- Explicitly addressed at least one of the target dimensions: class imbalance, concept drift, or interpretability/explainability.

Exclusion Criteria

The following categories of studies were excluded:

- Non-technical articles or reports that did not present substantive analytical or methodological content.
- Works limited to signature-only malware detection approaches.
- Studies that omitted quantitative evaluation metrics.
- Duplicate publications or papers deemed methodologically weak based on quality appraisal.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Selection Process

The study selection procedure followed the PRISMA guidelines. A total of 1,982 records were initially identified. After removing duplicates and screening titles and abstracts, 743 records proceeded to screening. Of these, 216 full-text articles were assessed for eligibility. Ultimately, 162 studies met all inclusion criteria and were included in the final review.

Data Extraction Dimensions

The data extraction process was guided by a structured set of analytical dimensions to ensure consistency and comparability across the reviewed studies. For each publication, the following attributes were systematically recorded (Thomas & Harden, 2008):

- **Dataset type:** Characteristics and sources of the datasets used for experimentation.
- **Analysis method:** The employed malware analysis approach, categorised as static, dynamic, or hybrid.
- **ML/DL model class:** The specific machine learning or deep learning techniques implemented.
- **Imbalance handling strategy:** The methods used to address class imbalance within the dataset.
- **Drift detection or adaptation mechanisms:** Techniques incorporated to identify or mitigate concept drift.
- **Interpretability method:** Approaches applied to enhance model transparency and explainability.
- **Evaluation metrics:** Performance measures used to assess the effectiveness of the proposed models.

Results

This section presents a structured synthesis of the reviewed studies, organized around four major thematic domains relevant to contemporary malware detection research.

Traditional malware detection approaches (Aslan & Samet, 2020) namely signature-based (Chakravarty et al., 2019), heuristic (Zakeri et al., 2015), behavioural(Chakravarty et al., 2019), and anomaly-based (Tang et al., 2014)methods remain widely deployed due to their computational efficiency and low operational overhead. However, these techniques exhibit inherent limitations when confronted with modern threat landscapes. Their detection efficacy is significantly reduced in scenarios involving:

- previously unseen (zero-day) malware(Kim et al., 2023),
- polymorphic or metamorphic variants capable of continuous mutation (Brezinski & Ferens, 2023), and
- adversarial behaviours engineered to evade dynamic or sandbox-based analysis (Aryal et al., 2025).

These constraints highlight the diminishing utility of conventional mechanisms in dynamic and adversarial environments.

Classical machine learning algorithms, such as Support Vector Machines (SVM), Random Forests (RF), Decision Trees (DT), and Naïve Bayes (NB), introduced improved generalization capabilities compared with purely signature-driven systems (Halbouni et al., 2022). Despite these gains, their performance remains contingent upon extensive manual feature engineering, limiting adaptability to evolving threat distributions. Furthermore, these models are susceptible to concept drift, resulting in gradual degradation as malware behaviors and feature distributions shift over time.

Deep learning has significantly advanced malware detection by enabling automated feature extraction and improving robustness across heterogeneous data representations (Halbouni et al., 2022). The reviewed studies employ a diverse set of architectures, each offering specific strengths and presenting notable limitations, as summarized in Table 1.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Table 1 ***Summary of deep learning architectures used in malware detection and their associated strengths and limitations.***

Model Class	Strengths	Limitations
CNN	Learns fine-grained byte or image-level patterns (Khan et al., 2023)	Limited ability to capture long-range temporal dependencies
RNN/LSTM/GRU	Effective for sequential data (e.g., API calls, opcodes) (Athiwaratkun & Stokes, 2017; Nguyen et al., 2020; Park & Lee, 2018)	Sensitive to long sequences; prone to gradient-related instability
GNN	Models relational and structural dependencies (e.g., CFGs, system-call graphs) (Shokouhinejad, Higgins, et al., 2025)	High computational and preprocessing costs
Autoencoders	Suitable for anomaly detection and feature compression (Lopez Pinaya et al., 2020)	Limited discriminative power for multi-class malware
GANs	Useful for data augmentation and adversarial training (Buriro et al., 2025)	Training instability and mode collapse issues

The findings indicate that no single deep learning architecture adequately captures structural information, temporal behaviour, and distributional variability within malware datasets. This motivates the increasing interest in hybrid and fusion-based models such as the proposed MAD-FIT framework that integrate complementary representational strengths to enhance robustness and adaptability.

Deep Learning for Static, Dynamic, and Hybrid Analysis

Deep learning–based malware detection pipelines generally follow the end-to-end process illustrated in Figure 1, beginning with data acquisition, followed by feature extraction, model learning, and final classification. Building on this overarching workflow, deep learning approaches to malware analysis can be categorized into static, dynamic, and hybrid methodologies. A detailed comparison of these three modalities is provided in Fig. 2, while Fig. 3 illustrates a representative attention-based hybrid architecture.

Static Analysis

Static analysis derives features directly from executable files without initiating runtime execution. Within the pipeline shown in Fig. 1., static analysis corresponds to the static feature extraction stage and typically includes:

- Image-based binary representations processed with CNN models (Nataraj et al., 2011),
- Raw byte-sequence learning, including MalConv-style architectures (Raff et al., 2018), and
- Graph-based representations of control-flow or call graphs analyzed using GNNs (X. Zhang et al., 2021; Y. Zhang, 2021).

As depicted in the left panel of Fig. 2., static approaches offer strong scalability and low computational overhead. However, they are highly susceptible to obfuscation, packing, and polymorphic transformations, which can significantly distort discriminative patterns and undermine model robustness (Bayer et al., 2009).

Dynamic Analysis

Dynamic analysis observes program behavior during controlled execution and aligns with the dynamic feature extraction branch outlined in Fig. 1. Representative deep learning methods include:

- Sequential modeling of API call traces using LSTM or GRU networks (Ki et al., 2015; Pascanu et al., 2015),
- System-call dependency graphs processed using GNN architectures (Nikolopoulos & Polenakis, 2015; Zhao, 2019), and
- Network-activity sequences captured and modeled by CNN or RNN variants (Alshoulie & Mehmood, 2025).

As illustrated in the central panel of Fig. 2., dynamic analysis provides improved resilience to static code obfuscation. Nevertheless, its effectiveness is constrained by sandbox evasion tactics, environment-aware malware, and the risk of incomplete behavioral traces when malicious actions are not triggered during execution (Bayer et al., 2009; *Cuckoo Sandbox: Open Source Automated Malware Analysis*, 2021; Alshmarni & Alliheedi, 2024).

Hybrid Analysis

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Hybrid approaches integrate both static and dynamic features to leverage their complementary strengths. Within the general workflow of Fig. 1., hybrid systems perform parallel static and dynamic encoding, followed by feature fusion and classification.

The right panel of Fig. 2. highlights common hybrid strategies, while Fig. 3. presents a more advanced attention-based fusion architecture (Alzaylaee et al., 2020; Cui et al., 2020). In this design, static and dynamic encoders produce independent representations that are subsequently aligned and weighted through a cross-modal attention mechanism. This enables the model to adaptively emphasize the most informative modality for each sample.

Hybrid systems consistently outperform single-modality detectors, particularly against evasive, obfuscated, or rapidly evolving malware variants (Alzaylaee et al., 2020). The flexibility introduced by attention mechanisms further enhances robustness by dynamically capturing dependencies across heterogeneous feature spaces, a principle central to the proposed MAD-FIT framework's design.

Recent architectures emphasize adaptive triggering of resource-intensive analysis. For instance, a 2025 hybrid system uses a self-evaluation agent to execute costly dynamic analysis on a remote server only when the confidence of the local static classifier drops, effectively managing computational resources while responding to concept drift (Augello et al., 2025a).

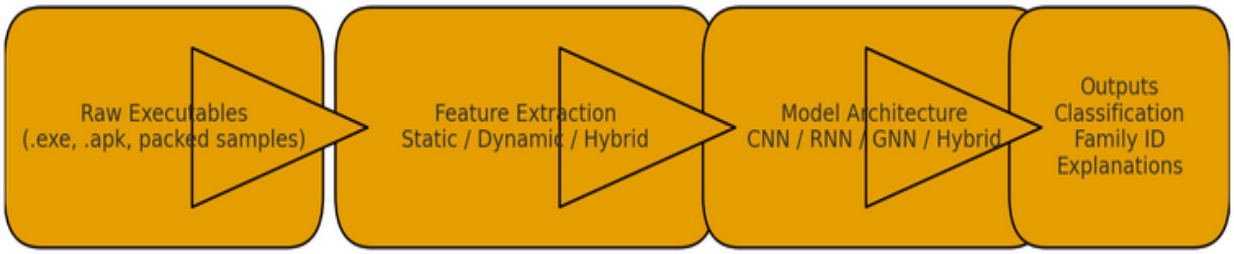


Fig. 1. End-to-end deep learning–based malware detection pipeline. The process begins with raw executable samples, followed by static, dynamic, or hybrid feature extraction techniques. Extracted features are then processed using deep learning architectures (e.g., CNN, RNN, GNN, or hybrid models) to produce final outputs such as malware classification labels and optional interpretability explanations.

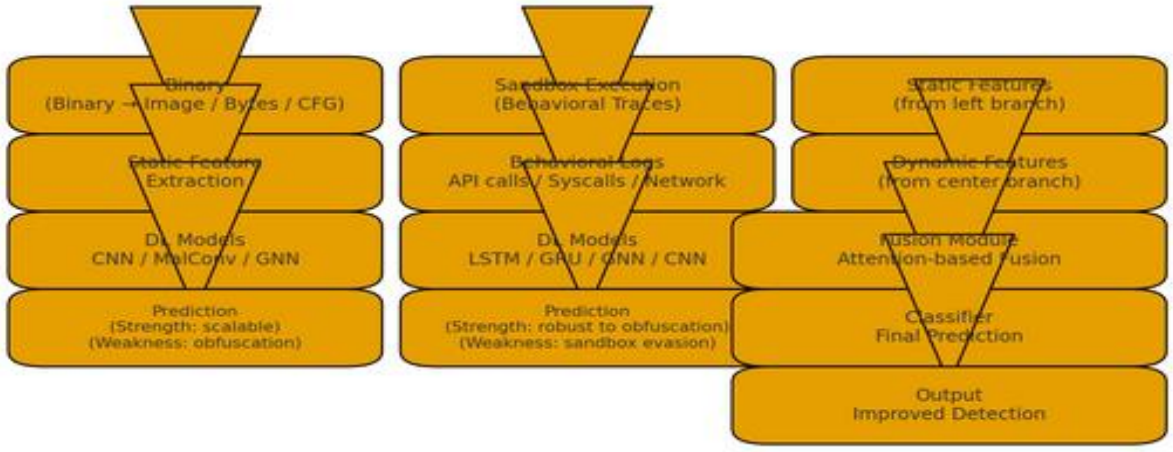


Fig. 2. Comparative workflow of static, dynamic, and hybrid deep learning-based malware analysis. Comparison of deep learning workflows for malware detection: (a) static analysis using features derived from executable binaries, (b) dynamic analysis based on behavioral traces acquired during sandbox execution, and (c) hybrid analysis integrating both modalities with attention-based or concatenation-based fusion.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

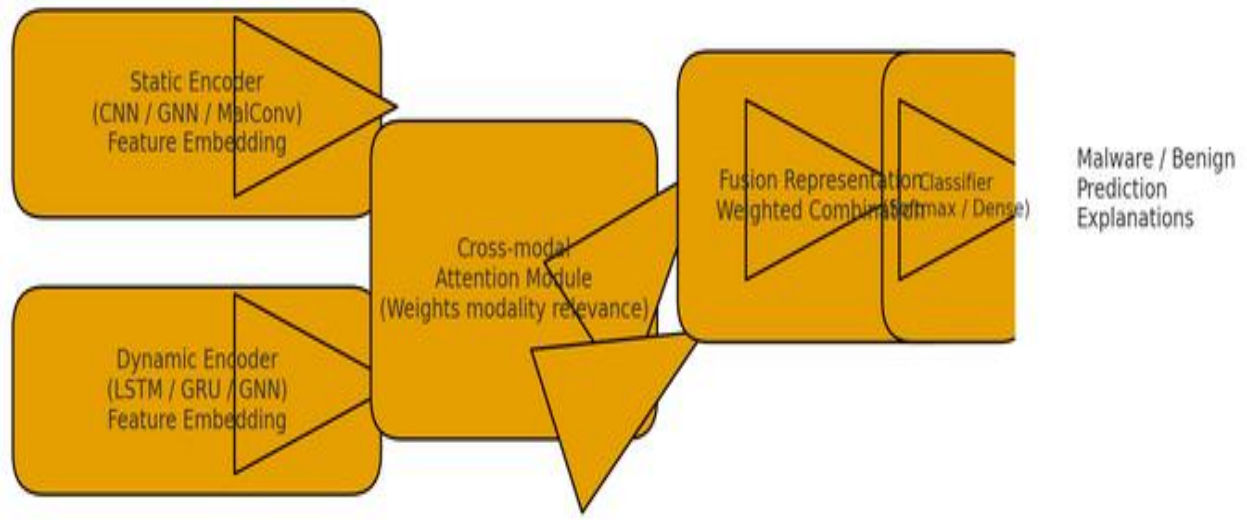


Fig. 3. Attention-based hybrid malware detection architecture. Static and dynamic feature encoders generate parallel representations, which are jointly processed by a cross-modal attention module to produce a fused embedding used for final classification.

Class Imbalance Handling in Malware Datasets

Class imbalance remains a persistent challenge in malware detection, significantly influencing the performance and generalization capability of machine learning and deep learning models. The nature of this imbalance and corresponding mitigation strategies are illustrated in Fig. 4 and Fig. 5, respectively.

Nature of Imbalance

Malware datasets typically exhibit a pronounced long-tailed distribution, wherein a small number of prevalent malware families account for the majority of available samples, while numerous minority families appear infrequently (Almajed et al., 2025; Panda et al., 2025). This distributional pattern is visualized in Fig. 4, which highlights the dominance of common families and the sharp decline in sample counts among rare families.

Rare families, despite their limited representation, often correspond to emerging, targeted, or higher-risk variants. Their underrepresentation poses a considerable challenge, as models may become biased toward majority classes (Almajed et al., 2025; Souza et al., 2025). Furthermore, dataset composition is frequently affected by vendor or source-specific collection biases, which amplify skewness and reduce the representativeness of the overall distribution.

Traditional Methods

Conventional approaches to mitigating class imbalance include oversampling, undersampling, synthetic minority oversampling (SMOTE) (Almajed et al., 2025), and cost-sensitive learning. While oversampling increases minority class presence, it often leads to overfitting due to the repetition of limited samples. Undersampling reduces dataset size and can cause information loss by discarding relevant majority-class samples. SMOTE attempts to generate synthetic samples but may introduce unrealistic instances in high-dimensional malware feature spaces (Almajed et al., 2025; Tuan et al., 2025). Cost-sensitive learning adjusts misclassification penalties but is often difficult to tune effectively, especially in dynamic threat environments.

These traditional strategies are summarized in the upper branch of Fig. 5, demonstrating their role in balancing data prior to model training.

Deep Learning Approaches

"Recent research increasingly leverages generative deep learning models, particularly generative adversarial networks (GANs) and variational autoencoders (VAEs), to augment minority malware families (Choi et al., 2023; Joshi et al., 2025). These models aim to generate synthetic samples that better reflect underlying distributions, thereby enhancing classifier robustness (Ajayi et al., 2025). However, several challenges remain:

- **Distributional Drift:** GAN-generated malware samples may suffer from distributional drift, reducing their realism and utility. While GANs can enhance datasets (Joshi et al., 2025), their output's fidelity for security tasks requires careful validation.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

- **Evaluation Gap:** Few studies incorporate security-oriented assessment metrics (e.g., behavioral fidelity, functional validation) to evaluate synthetic sample quality. Most validation relies on classifier performance rather than the samples' malicious properties (Choi et al., 2023).
- **Static Nature:** Existing augmentation techniques often treat imbalance as a static problem. They do not address dynamic class imbalance arising from concept drift, where malware families evolve or new variants emerge over time (McFadden et al., 2025; Roh et al., 2025). Truly adaptive systems must integrate drift-aware strategies to remain effective (Luo et al., 2024).

The lower branch of Fig. 5 illustrates the emerging generative modeling pipeline for imbalance mitigation.

Recent Advances and Synthesis

Recent research continues to highlight class imbalance as a primary impediment to model generalizability (Almajed et al., 2025). Solutions have evolved into three main categories:

- **Data-level techniques:** These include advanced synthetic oversampling strategies tailored for mobile malware (Almajed et al., 2025) and the use of Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) for augmentation (Almajed et al., 2025). A key development is the move towards dynamic quality assessment of generated samples to ensure their fidelity and usefulness.
- **Algorithm-level techniques:** Approaches include cost-sensitive learning, reweighted class-balanced loss functions (e.g., in DenseNet models), and architectures incorporating self-attention mechanisms to better discriminate minority classes (Almajed et al., 2025).
- **Hybrid and ensemble techniques:** There is a growing trend toward combining data- and algorithm-level methods. This includes hybrid CNN-LSTM models, federated learning frameworks designed for imbalanced data, and the integration of evolutionary optimization with deep learning for feature and hyperparameter tuning (Almajed et al., 2025).

Despite these advancements, a 2025 review concludes that a significant challenge remains the integration of imbalance correction with concept drift adaptation (Almajed et al., 2025). Most GAN-based augmentations generate samples based on historical distributions and do not dynamically adjust to the evolving feature space of new malware variants, creating a disconnect between data balancing and temporal model adaptation.

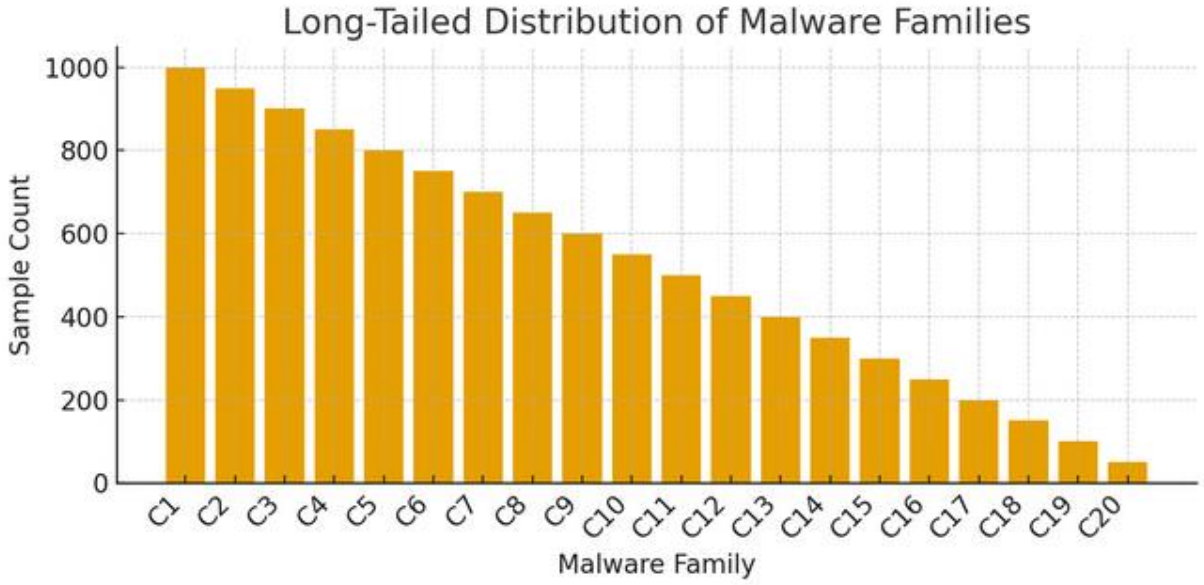


Fig. 4. Long-Tailed Malware Distribution. Illustration of the long-tailed distribution commonly observed in malware datasets. A small number of dominant families account for most samples, while numerous minority families are severely underrepresented.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

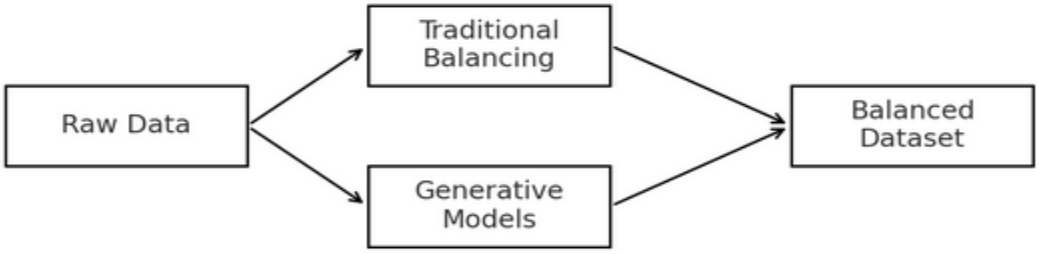


Fig. 5. Workflow of Imbalance-Handling Approaches. High-level workflow of class imbalance mitigation strategies. Raw data may be processed via traditional balancing techniques (e.g., oversampling, undersampling, SMOTE) or modern deep generative models (e.g., GANs, VAEs) to produce a more balanced dataset.

Concept Drift in Malware Detection

Concept drift constitutes a major challenge in malware detection systems, arising when the statistical properties of malware behavior evolve over time. Such drift may result from natural code evolution or deliberate adversarial manipulation, and it can significantly degrade the performance of static or continuously trained models. The primary forms of drift encountered in the malware domain are illustrated in Fig. 6.

Recent empirical studies confirm that concept drift is a pervasive and significant factor degrading the performance of ML-based Android malware detectors, affecting models across different feature types and algorithmic families (Sabbah et al., 2025). Furthermore, this drift is increasingly driven by deliberate adversarial manipulation, as evidenced by novel attack methods that can successfully evade leading ML detectors with high success rates and minimal, functionality-preserving perturbations (C. Li et al., 2025). Despite progress in defensive strategies like adversarial training, recent comprehensive evaluations show that many state-of-the-art defenses remain critically brittle when faced with sophisticated, binary-space-optimized attacks (Jafari & Shameli-Sendi, 2026).

Types of Drift Observed

Three drift patterns are commonly observed in malware datasets (see Fig. 6):

- **Sudden (abrupt) drift**, where the introduction of a new malware family or variant leads to immediate changes in feature distributions.
- **Gradual drift**, characterized by incremental code modifications or slowly evolving behavior, often associated with polymorphic or metamorphic evolution.
- **Recurring (cyclical) drift**, where previously seen malware families re-emerge after periods of inactivity, reflecting code reuse, reweaponization, or seasonal attack campaigns.

These drift types impose heterogenous temporal dynamics, complicating long-term model robustness and necessitating adaptive detection mechanisms.

Drift Detection Techniques

Drift detection approaches aim to identify when model performance degradation corresponds to underlying distributional change. An example of error-based drift detection over time is presented in Fig. 7.

Common techniques include:

- **ADWIN**, a statistically grounded, window-based detector capable of identifying abrupt changes but prone to excessive sensitivity in noisy behavioral environments.
- **DDM and EDDM**, which rely on monitoring classifier error rates. These methods offer useful stability under incremental drift but often exhibit detection lag under rapidly changing malware behaviors.
- **Divergence-based methods** (e.g., KL-divergence), which measure shifts between distributions but incur considerable computational overhead for high-dimensional static and dynamic features typical of malware.

While each approach provides distinct advantages, real-world deployment requires balancing sensitivity, robustness to noise, and computational feasibility.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Drift Adaptation Techniques

Once concept drift is detected, the model must adapt to the evolving data distribution. Recent literature highlights a shift toward more autonomous and deployment-oriented adaptation strategies. These approaches are summarized conceptually in Figure 7, which illustrates the workflow of contemporary drift-adaptation mechanisms.

- **Self-Supervised Test-Time Adaptation (TTA):** A significant advance is demonstrated by the 2025 MADCAT framework, which employs a Masked Autoencoder (MAE) to perform self-supervised fine-tuning on unlabeled test-time samples (Roh et al., 2025). This enables continuous adaptation to distributional shifts without requiring immediate labeled feedback, alleviating a key limitation of supervised retraining strategies (Roh et al., 2025).
- **Resource-Aware Hybrid Adaptation:** Recent work targeting mobile, edge, and IoT environments proposes multi-level adaptation pipelines in which a lightweight on-device static analyzer is monitored by a self-evaluation agent (Augello et al., 2025a). When confidence degradation signals drift, only high-risk samples are escalated to remote dynamic analysis. This selective-offloading design preserves energy and computational resources while maintaining robustness (Augello et al., 2025a).
- **Continual and Online Learning:** Traditional drift-adaptation strategies including incremental learning, ensemble refresh, and regularization-based methods such as Elastic Weight Consolidation (EWC)—remain widely used. Modern studies increasingly evaluate these approaches within federated or hybrid learning contexts to address distributional heterogeneity and privacy constraints (Augello et al., 2025a).

The integration of self-supervised test-time adaptation with traditional supervised and continual-learning mechanisms represents a promising hybrid direction (Roh et al., 2025). Furthermore, emerging architectures incorporate resource efficiency and selective analysis as design principles (Augello et al., 2025a), shifting the field from purely algorithmic solutions toward system-level resilience (see Figure 7).

Fig. 6. Types of Concept Drift. Overview of the primary types of concept drift observed in malware datasets, including sudden, gradual, and recurring forms.

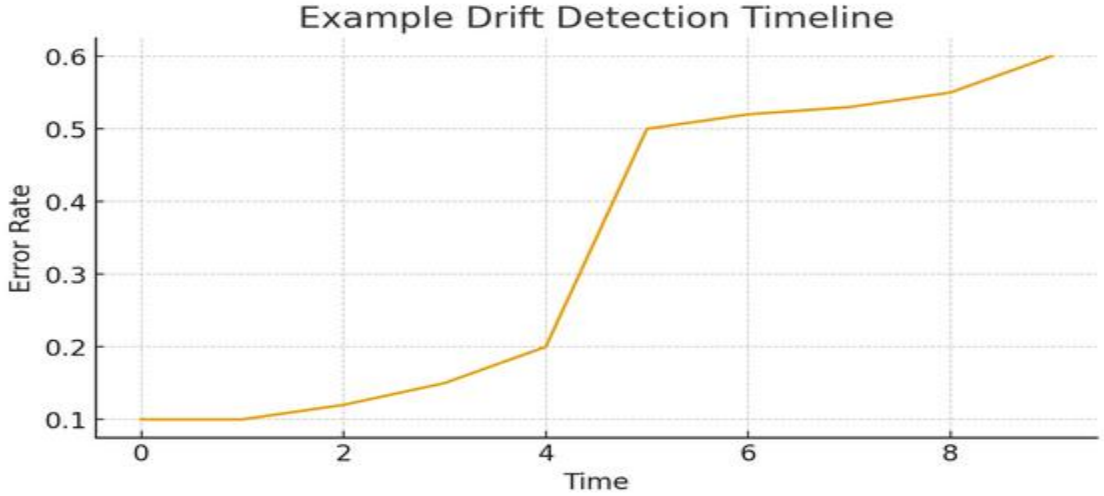


Fig. 7. Drift Detection Timeline. Example timeline illustrating drift detection based on error-rate monitoring. A sharp increase in error rate around time step 5 signals the onset of abrupt drift.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

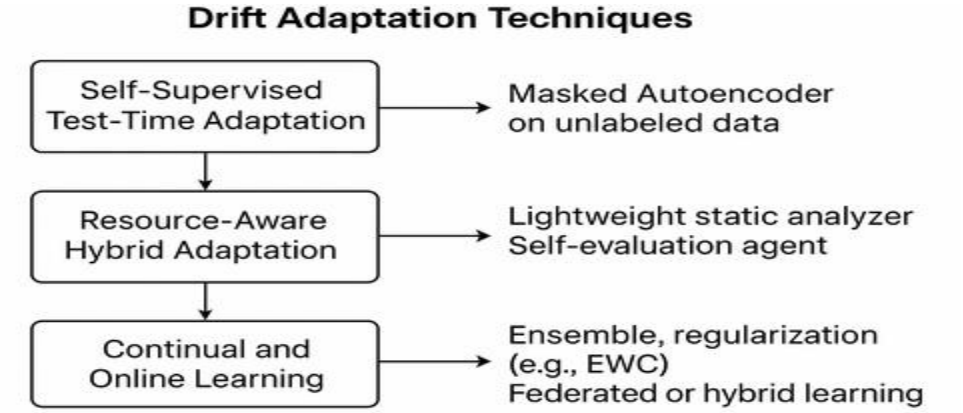
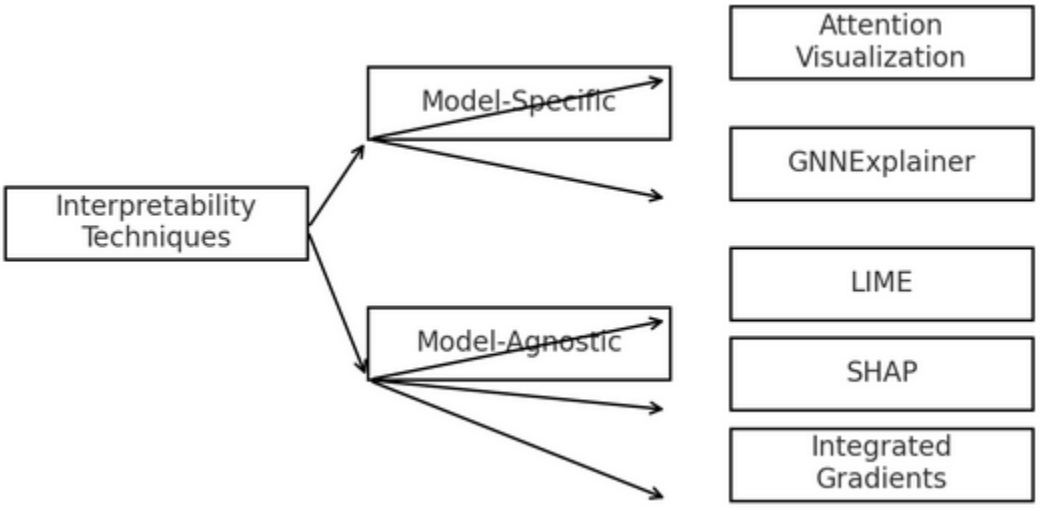


Fig. 8. Conceptual workflow of modern drift adaptation strategies. The diagram summarizes three principal mechanisms: (i) self-supervised test-time adaptation using masked autoencoders on unlabeled data, (ii) resource-aware hybrid adaptation combining on-device static analysis with selective remote dynamic evaluation, and (iii) continual and online learning approaches, including ensemble and regularization-based methods within federated or hybrid settings.

Interpretability in Malware Detection

As deep learning models increasingly drive automated malware detection, interpretability has emerged as a critical requirement for operational trust, regulatory compliance, and effective security workflows. The overarching landscape of interpretability techniques is summarized in Fig. 9.

A recent systematic review of XAI in cybersecurity, analyzing some studies, corroborates the field's fragmented state, noting that while techniques like LIME and SHAP are explored, a unified framework for integrating interpretability into operational security workflows remains lacking (Ofusori et al., 2025). However, emerging research demonstrates the feasibility of building inherently interpretable and lightweight models capable of generalizing to unseen malware variants, using techniques like feature importance analysis with SHAP (Madamidola et al., 2025). Other proposals move towards integrated, scalable frameworks that combine big data processing with explainability modules (e.g., Grad-CAM and SHAP) to provide transparent insights for administrators (Upender et al., 2025).



Fig, 9. Interpretability Taxonomy for Deep Learning–Based Malware Detection. A hierarchical taxonomy illustrating model-specific and model-agnostic interpretability techniques used in malware detection. The diagram categorizes major methods—including attention mechanisms, GNNExplainer, LIME, SHAP, and Integrated Gradients—and highlights their conceptual relationships.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Importance

Interpretability plays a central role in ensuring that malware detection systems are both trustworthy and operationally useful. First, transparent explanations enhance security analyst confidence, enabling experts to validate alerts and differentiate between true positives and false positives. Second, interpretability supports incident response workflows, allowing practitioners to identify malicious features, understand behavioral triggers, and prioritize containment actions. Third, evolving regulatory frameworks such as the General Data Protection Regulation (GDPR) and the EU AI Act, increasingly mandate explainability for automated decision-making, making interpretability a requirement rather than an optional enhancement.

Interpretability Techniques

Interpretability methods applied to malware detection can be grouped into two major families, as illustrated in Fig. 9: model-specific and model-agnostic techniques.

Model-Specific Techniques

Model-specific approaches leverage architectural components or structural properties inherent to the underlying detection model. Examples include:

- Attention-based visualization, which highlights influential byte regions, instruction sequences, or behavioral features that drive model predictions.
- GNNExplainer, which isolates key subgraphs, nodes, and edges in graph-based malware representations such as control-flow or call graphs.

These techniques often provide deeper insight into the internal representation learned by the model.

Model-Agnostic Techniques

Model-agnostic interpretability methods operate independently of model architecture and thus support broad applicability across diverse malware pipelines. Common approaches include:

- LIME, which generates local surrogate models to approximate feature contributions,
- SHAP, which provides Shapley-value–based importance scores, and
- Integrated Gradients, which measures feature attribution along a gradient-based path.

A comparative evaluation of these techniques, based on criteria such as scalability, granularity, and computational overhead, is presented in Fig. 10.

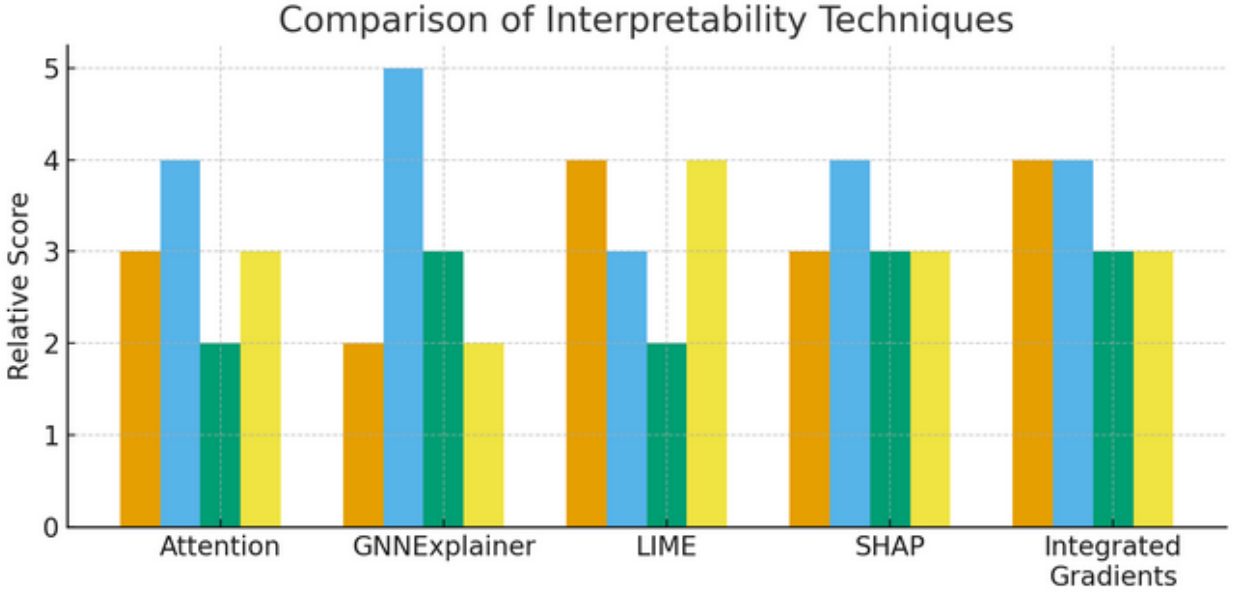


Fig. 10. Comparative Overview of Interpretability Techniques. A comparative chart summarizing key characteristics of commonly applied interpretability methods in malware detection. The figure contrasts scalability,

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

explanation granularity, computational cost, and model dependence across both model-specific and model-agnostic approaches.

Shortcomings in Existing Work

Shortcomings and Recent Synthesis

Despite notable progress, the application of explainable AI (XAI) in malware detection remains constrained by several unresolved challenges. A dedicated 2025 systematic review of XAI in cybersecurity (Ofusori et al., 2025) provides evidence that advancements have been fragmented and insufficiently aligned with operational requirements.

Fragmentation and Lack of Integration.

Most existing contributions apply techniques such as LIME, SHAP, and Grad-CAM as isolated post-hoc tools (Ofusori et al., 2025). Comprehensive frameworks that integrate interpretability throughout the malware analysis pipeline particularly those suitable for security operations center (SOC) environments are largely absent. This lack of architectural integration limits the practical utility of current XAI approaches (Ofusori et al., 2025).

Actionability Deficit.

A recurrent issue concerns the limited operational relevance of explanations. While many methods highlight salient features or API calls, they seldom articulate the broader behavioral or semantic implications of these indicators. Consequently, explanations often fail to provide analysts with meaningful or actionable intelligence.

Insufficient Validation.

The literature exhibits a marked shortage of human-subject evaluations and domain-specific assessment protocols. There is limited empirical evidence demonstrating that XAI methods improve analyst efficiency, decision accuracy, situation awareness, or trust calibration. This validation deficit weakens claims regarding the real-world effectiveness of proposed techniques (Iadarola et al., 2021; Ofusori et al., 2025).

Computational Constraints.

Model-agnostic approaches particularly perturbation-based methods such as LIME incur substantial computational overhead. This restricts their applicability in latency-sensitive or large-scale malware screening pipelines, where throughput and resource efficiency are critical.

Synthesis and Emerging Directions.

Foundational work, including early applications of Grad-CAM to image-based malware classification, established the feasibility of model-specific interpretability (Iadarola et al., 2021). Recent research trends have moved toward more holistic solutions, including multi-method XAI pipelines and the integration of interpretability mechanisms within adaptive detection systems such as resource-aware drift-handling architectures. These developments indicate a gradual shift from isolated interpretability tools toward unified, operationally grounded frameworks (Augello et al., 2025a).

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

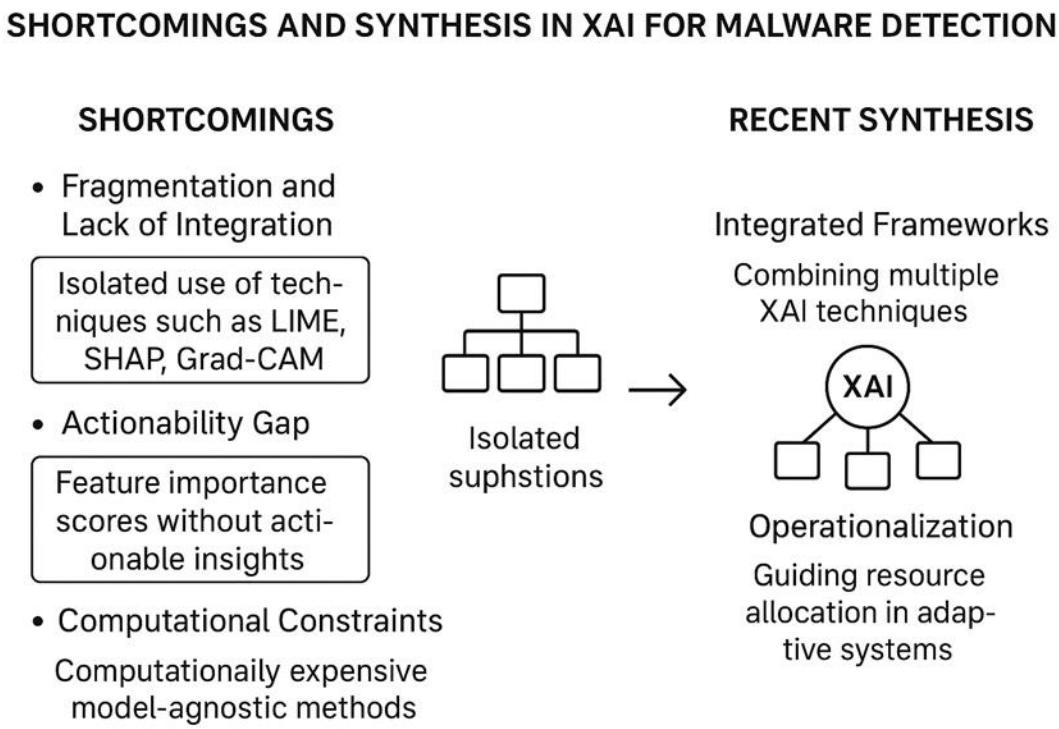


Figure 11. Interpretability Shortcomings and Recent Synthesis Overview. Illustration of the primary shortcomings and emerging synthesis directions in malware interpretability research, including fragmentation of XAI methods, limited actionability, deficient validation practices, and computational constraints. The schematic also highlights the recent shift toward integrated, multi-method interpretability pipelines.

Unified Synthesis and Design Requirements for Robust Malware Detection

The preceding sections establish that modern malware detection systems operate under three interacting pressures: severe class imbalance, continuous concept drift, and increasing demands for interpretability. While each challenge has been studied extensively in isolation, this review reveals that their combined impact is insufficiently addressed by existing approaches. Figure 12 conceptually summarizes this intersection and its implications for real-world deployment.

Interdependency of Core Challenges

A critical observation emerging from this review is that class imbalance, concept drift, and interpretability are not independent phenomena, but rather mutually reinforcing constraints:

Class imbalance amplifies concept drift effects, as minority malware families, often representing emerging threats, are precisely those most affected by temporal distributional shifts. Models biased toward majority classes thus exhibit delayed or failed adaptation to new attack vectors.

Concept drift undermines interpretability, as explanations derived from outdated representations may no longer reflect current malicious behaviors, leading to misleading or obsolete analyst insights.

Post-hoc interpretability methods struggle under imbalance and drift, since feature attributions are typically computed with respect to static training distributions that no longer hold in evolving environments.

These interactions expose a fundamental limitation of existing pipelines: optimizing for one challenge often degrades performance along another dimension.

Limitations of Current Integrated Solutions

Although recent works propose partial integrations—such as GAN-based augmentation for imbalance, self-supervised test-time adaptation for drift, or attention mechanisms for interpretability—these components are rarely co-designed. Instead, they are typically appended as modular fixes, resulting in:

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

- Temporal misalignment, where imbalance correction is performed offline while drift occurs online;
- Explanation fragility, where interpretability mechanisms are not updated alongside adapted model parameters; and
- Resource inefficiency, particularly in hybrid static–dynamic systems where interpretability and adaptation introduce prohibitive overhead.

Notably, no reviewed framework simultaneously satisfies (i) drift-aware learning, (ii) minority-sensitive representation, and (iii) operationally meaningful interpretability within a unified architecture.

Derived Design Requirements

From the systematic analysis conducted in Sections 3.1–3.5, the following design requirements emerge for next-generation malware detection systems:

1. **Multi-Modal Robustness**
   The architecture must integrate static, dynamic, and structural representations to ensure resilience against obfuscation, sandbox evasion, and behavioral sparsity.
2. **Drift-Aware Adaptation Without Label Dependence**
   Continuous adaptation should be achievable using unlabeled or weakly labeled data (e.g., via self-supervised or test-time adaptation), minimizing reliance on delayed analyst feedback.
3. **Minority-Sensitive Learning Under Temporal Evolution**
   Class imbalance handling must be dynamic, adjusting to evolving family distributions rather than relying on static augmentation strategies.
4. **Embedded Interpretability**
   Explanations should be architecturally integrated, not post-hoc, enabling alignment between model adaptation, feature importance, and analyst-facing insights.
5. **Resource-Aware Deployment**
   The system must support selective computation, escalating to expensive dynamic analysis or explanation generation only when confidence degradation or drift is detected.

Motivation for the Proposed MAD-FIT Framework

These requirements collectively motivate the proposed MAD-FIT (Multi-Modal Adaptive Drift- and Imbalance-Aware Framework with Integrated Transparency). Unlike prior approaches, MAD-FIT is designed from the outset to co-evolve detection, adaptation, and explanation mechanisms, ensuring that:

- Interpretability remains valid under drift,
- Minority classes are adaptively emphasized as threat landscapes change, and
- Computational resources are allocated intelligently across static and dynamic analysis stages.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

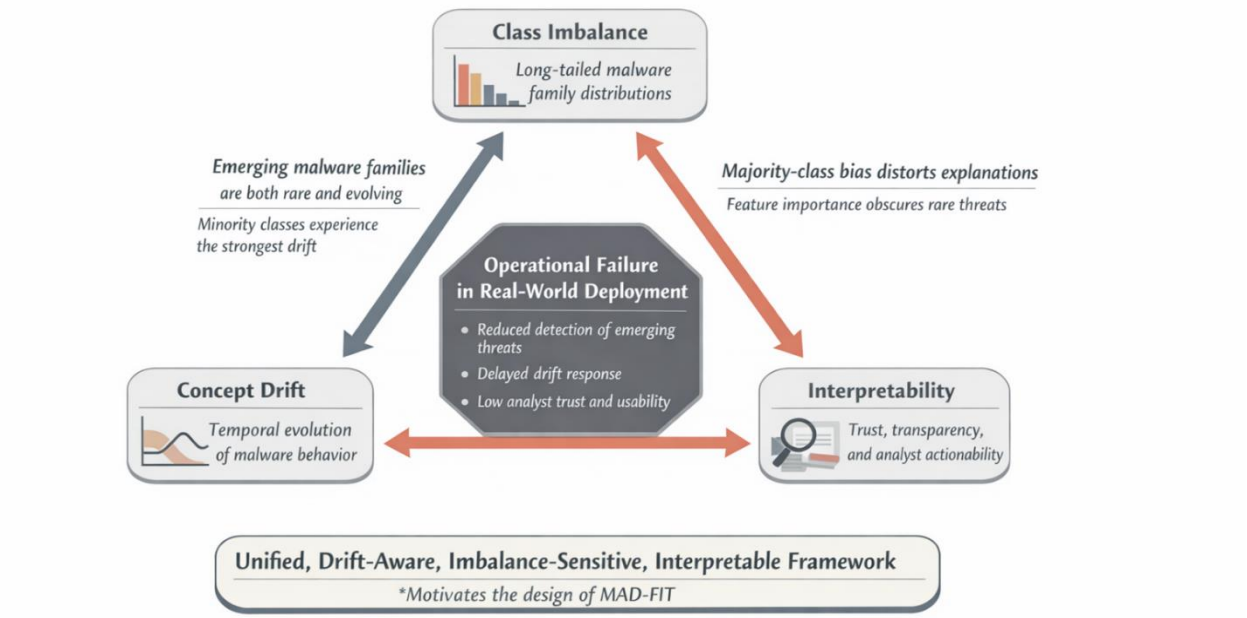


Figure 12. Unified challenge interaction in malware detection.
The diagram illustrates the interdependent relationship between class imbalance, concept drift, and interpretability in deep learning–based malware detection systems. Class imbalance amplifies drift effects on emerging malware families, concept drift degrades the validity of explanations, and biased interpretability further obscures minority threats. Their interaction leads to compounded operational failures, motivating the need for unified, adaptive, and interpretable frameworks such as MAD-FIT.

Comparative Analysis

This section provides a consolidated comparative analysis of state-of-the-art (SOTA) malware detection approaches across the three core challenge dimensions identified in Section 3: class imbalance, concept drift, and interpretability. Rather than evaluating models solely on predictive accuracy, this analysis emphasizes robustness, adaptability, and operational readiness, which are essential for real-world malware detection systems. A structured summary of these trade-offs is presented in Table 2, while key comparative insights are synthesized below.

Comparison Across Detection Paradigms

Malware detection techniques can be broadly categorized into traditional ML, deep learning single-modality, hybrid deep learning, and adaptive/continual learning frameworks. Each paradigm demonstrates strengths in specific dimensions but exhibits critical weaknesses when evaluated holistically.

Traditional machine learning models (e.g., SVM, RF, NB) offer low computational overhead and moderate interpretability, particularly when feature sets are manually curated. However, as established in Sections 3.1 and 3.4, these methods scale poorly under concept drift and rely heavily on static feature engineering, making them ill-suited for rapidly evolving malware ecosystems.

Single-modality deep learning approaches, whether static (e.g., CNN-based byte or image models) or dynamic (e.g., RNN-based API sequence models), achieve superior detection accuracy and improved generalization. Nonetheless, they remain vulnerable to evasion strategies targeting their respective modalities and typically lack mechanisms for online adaptation or minority-aware learning.

Hybrid deep learning systems represent a notable advancement by integrating static and dynamic features, often via attention-based fusion. As shown in Section 3.2.3, these models consistently outperform unimodal detectors under obfuscation and sandbox-evasion conditions. Despite this progress, most hybrid systems remain offline-trained and reactive, addressing imbalance and drift through periodic retraining rather than continuous adaptation.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Handling of Class Imbalance

Table 2 reveals that class imbalance mitigation is inconsistently addressed across SOTA methods. Traditional oversampling and cost-sensitive learning are widely applied but are largely static and dataset-dependent. Generative deep learning techniques (GANs, VAEs) offer improved minority representation but introduce distributional uncertainty and lack security-oriented validation.

Importantly, very few approaches couple imbalance handling with temporal awareness. As discussed in Section 3.3, most augmentation strategies assume stationary class distributions, failing to account for the fact that minority classes often correspond to emerging malware families subject to drift. This disconnect significantly limits long-term robustness.

Concept Drift Awareness and Adaptation

Concept drift remains one of the most under-addressed challenges in malware detection. While drift detection techniques such as ADWIN, DDM, and divergence-based methods are well-established, their integration into end-to-end malware pipelines is limited.

Recent adaptive frameworks, such as self-supervised test-time adaptation (e.g., MAE-based approaches) and resource-aware hybrid systems, demonstrate promising results (Section 3.4.3). However, these solutions often operate independently of imbalance correction and interpretability, leading to partial adaptation that does not fully resolve performance degradation on rare or evolving malware families.

Moreover, adversarially induced drift further exposes the brittleness of many adaptive defenses, as highlighted by recent evaluations showing high evasion success rates against state-of-the-art detectors (Li et al., 2025; Jafari & Shameli-Sendi, 2026).

Interpretability and Operational Utility

Interpretability analysis (Section 3.5) reveals a clear divide between research-focused explainability and operationally actionable explanations. Model-agnostic methods such as LIME and SHAP provide broad applicability but suffer from high computational cost and limited scalability. Model-specific approaches, including attention visualization and GNNExplainer, offer deeper insight but are tightly coupled to specific architectures.

Crucially, interpretability is rarely evaluated under drift or imbalance conditions. Explanations are typically generated post hoc, assuming stable feature distributions, which undermines their reliability in evolving threat environments. As a result, current XAI approaches often fail to support analyst trust, decision-making, or regulatory compliance in practice.

Integrated Trade-Off Analysis

The comparative evidence indicates a fundamental performance–adaptability–interpretability trade-off:

- Models optimized for accuracy and scalability (e.g., CNN-based static detectors) sacrifice adaptability and transparency.
- Drift-aware or adaptive systems often neglect minority sensitivity and explanation stability.
- Interpretable models tend to rely on simplified features or static assumptions, limiting detection robustness.

No reviewed approach simultaneously achieves:

1. Dynamic handling of class imbalance,
2. Continuous adaptation to concept drift, and
3. Integrated, low-overhead interpretability.

These gaps are summarized in Table 2, which highlights that existing solutions address these challenges in isolation rather than as an interconnected system-level problem.

Research Gap and Motivation

The comparative analysis confirms that current malware detection researches lack a unified framework capable of jointly addressing imbalance, drift, and interpretability within a single adaptive architecture. This limitation is not merely algorithmic but structural, stemming from the modular and retrospective design of existing solutions.

These findings directly motivate the proposed MAD-FIT framework, which is designed to:

- Integrate multi-modal representations,
- Perform drift-aware, minority-sensitive adaptation at test time, and
- Embed interpretability mechanisms aligned with model adaptation.

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Table 2. ***Comparative Analysis of State-of-the-Art Malware Detection Approaches***

Approach Category	Author(s) and Year	Representative Methods	Class Imbalance Handling	Concept Drift Handling	Interpretability Support	Key Limitations
Traditional ML (Static Features)	(Gibert et al., 2020; Halboumi et al., 2022)	SVM, RF, DT, NB, etc.	Cost-sensitive learning, basic resampling	None (static training)	High (feature-level)	Poor generalization, manual feature engineering, brittle under drift
Traditional ML (Dynamic Features)	(Ki et al., 2015)	HMMs, n-gram API models	Limited	None	Moderate	High false negatives under sandbox evasion
Static Deep Learning	(Raff et al., 2018)	CNN (binary images), MalConv	Typically ignored	Offline retraining only	Grad-CAM, saliency	Vulnerable to obfuscation and packing
Dynamic Deep Learning	(Nguyen et al., 2020)	RNN/LSTM/GRU on API calls	Rarely addressed	Partial (windowed retraining)	Attention visualization	Sandbox evasion, incomplete traces
Graph-Based DL	(Shokouhinejad, Razavi-Far, et al., 2025; Y. Zhang, 2021)	GNNs on CFGs / syscall graphs	Ignored	None	GNNExplainer	High computational and preprocessing cost
Hybrid DL (Static and Dynamic)	(Alzaylaee et al., 2020)	CNN–LSTM, Attention Fusion	Dataset-level balancing	Mostly offline	Attention-based	High resource cost, no continuous adaptation
Generative Augmentation	(Choi et al., 2023; Joshi et al., 2025)	GANs, VAEs	Synthetic minority samples	Static distributions	None	Distributional drift, weak security validation
Continual / Online Learning	(J. Li et al., 2020; Rebuffi et al., 2017; Sun et al., 2025)	Incremental learning, EWC, ensembles	Rarely addressed	Gradual drift handling	None	Catastrophic forgetting, label dependency
Self-Supervised TTA	(Roh et al., 2025)	MAE-based MADCAT	Not imbalance-aware	Test-time adaptation	None	No minority emphasis, opaque adaptation
Resource-Aware Hybrid Systems	(Augello et al., 2025a, 2025b)	Confidence-triggered analysis	Not explicit	Selective adaptation	Limited	Interpretability not integrated
XAI-Augmented Systems	(Abdallah et al., 2020; Akgündoğdu & Çelikbaş, 2025; Aliyavyil et al., 2022; Sar\i & Ac\i, 2025)	SHAP, LIME, Grad-CAM pipelines	Ignored	Assumes stationarity	High	High overhead, fragile under drift
Proposed Direction (MAD-FIT)		Multi-modal, drift-aware, interpretable	Dynamic, minority-sensitive	Self-supervised and continual	Embedded, adaptive	

Cite this article as:
Bodinga, A.B., Garko, A.B., Ibrahim, N.M., & Gabi. D. (2026). Deep learning-based malware detection under real-world constraints: A systematic review of class imbalance, concept drift and interpretability. *FNAS Journal of Computing and Applications, 3*(1), 87-92. https://doi.org/10.63561/jca.v3i1.1206

Research Gaps

Despite substantial advancements in deep learning–based malware detection, several critical gaps persist in the literature. These limitations impede the development of robust, adaptive, and interpretable detection frameworks. A summary of these research gaps is provided in Table 3.

Table 3 ***Research Gaps in Current Malware Detection Literature***

Research Gap	Description
Gap 1: Fragmented Solutions	Existing studies address class imbalance, concept drift, and interpretability independently, preventing the development of unified and operationally robust detection frameworks. This fragmentation persists, as noted in a recent 2025 SLR on XAI in cybersecurity, which concluded that while techniques like LIME and SHAP are explored, a unified framework for integrating interpretability into operational workflows remains absent (Ofusori et al., 2025).
Gap 2: Inadequate GAN Evaluation	GAN-based augmentation is commonly evaluated only through classifier performance, with limited assessment of structural, behavioral, or security relevance of generated samples.
Gap 3: No Adversarial Drift Modeling	Current drift detection and adaptation methods assume naturally occurring drift and do not account for adversarially induced distribution changes.
Gap 4: Lack of Integrated Interpretability Frameworks	A 2025 SLR on XAI in cybersecurity confirms that interpretability tools are applied in isolation and are rarely embedded into comprehensive, analyst-facing operational workflows (Ofusori et al., 2025). Most research focuses on generating explanations for model developers rather than designing usable interfaces for security analysts.
Gap 5: Limited Multi-Modal Fusion	Existing hybrid detection approaches seldom employ deeper attention-driven fusion of static, dynamic, and distributional features.
Gap 6: Limited Real-World Evaluation	Most studies rely on static, single-period datasets, with limited longitudinal, real-world, or deployment-oriented evaluation.
Gap 7: Absence of Holistic Evaluation Benchmarks	No standard benchmark exists to evaluate malware detection models simultaneously against the triad of class imbalance, concept drift (especially adversarial drift), and the need for actionable interpretability. Most studies test for one challenge using static, historically balanced datasets, failing to represent dynamic, real-world conditions (Almajed et al., 2025; Berrios et al., 2025).

Conceptual Framework

The findings of this review indicate that an effective next-generation malware detection system must move beyond isolated methodological advances and adopt an integrated, adaptive architecture. Such a framework should address class imbalance, concept drift, multi-modal heterogeneity, and interpretability in a cohesive manner. The key components of this conceptual framework are outlined below.

First, GAN-based dynamic data augmentation should be incorporated to mitigate both static and temporally evolving class imbalance. Unlike traditional oversampling techniques, generative augmentation can supply minority classes with structurally diverse samples that better reflect emerging malware behaviors.

Second, drift detection mechanisms combined with continual learning strategies are essential to maintain long-term model adaptability. This integration enables the system to recognize distributional changes induced by naturally evolving or adversarially manipulated malware and update its internal representations accordingly.

Third, the framework should employ attention-based multi-modal fusion, enabling the model to jointly leverage static features (e.g., binary structure), dynamic behavioral traces (e.g., API or system-call sequences), and distributional characteristics. Attention mechanisms allow the model to dynamically weight complementary signals across modalities, enhancing robustness and generalization.

Cite this article as:
Ossai, C., Odenema, U. A. & Omoyibo, S. E. (2026). Analysis of pathloss models for wireless network in Obiaruku, Delta State Nigeria. *FNAS Journal of Computing and Applications, 3*(1), 87-92.

Fourth, an integrated interpretability layer must be included to provide actionable, analyst-oriented explanations. Such a layer should unify model-specific and model-agnostic interpretability methods to support real-time decision-making, incident response, and compliance requirements.

Finally, the framework should be evaluated under explicit drift simulation protocols, capturing longitudinal, adversarial, and real-world deployment conditions that static offline benchmarks fail to represent.
Collectively, these components constitute the foundation of the proposed MAD-FIT (Malware Adaptive Detection with Fusion, Interpretation, and Training Dynamics) framework, which aims to deliver a unified, adaptive, and interpretable malware detection solution. The proposed framework aligns with the direction of cutting-edge research, incorporating principles from self-supervised test-time adaptation to handle label-scarce drift (Roh et al., 2025) and federated learning architectures to adapt collaboratively while preserving data privacy (Augello et al., 2025b).

Conclusion

This systematic literature review highlights substantial advancements in deep learning–based malware detection while also revealing persistent shortcomings that limit operational effectiveness. Although numerous studies demonstrate promising results under controlled experimental conditions, current approaches seldom address the combined challenges of class imbalance, concept drift, and model interpretability. The absence of integrated solutions leads to noticeable discrepancies between laboratory performance and real-world deployment requirements. The review underscores the need for malware detection systems that are not only accurate but also adaptive to evolving threats, robust to skewed data distributions, and transparent in their decision-making processes. These insights provide strong justification for developing a unified framework capable of jointly addressing these constraints. In response, the proposed MAD-FIT framework is positioned as a comprehensive, adaptive, and interpretable solution designed to meet the practical demands of contemporary malware detection environments.

Recommendations and Future Research Directions

Based on the findings of this systematic literature review, several recommendations are proposed to guide future advancements in deep learning-based malware detection research and practical implementation.

First, future studies should prioritize the development of unified malware detection frameworks that simultaneously address class imbalance, concept drift, and model interpretability within a single architecture. Current research largely treats these challenges independently, resulting in fragmented solutions that perform well under controlled experimental conditions but lack operational robustness. Integrated frameworks such as the proposed MAD-FIT architecture represent a necessary progression toward adaptive and trustworthy malware defense systems.

Second, greater emphasis should be placed on proactive concept drift adaptation and adversarial resilience. Existing drift-handling methods predominantly focus on reactive adaptation to distributional changes without sufficiently modeling malicious evolutionary intent. Future work should explore reinforcement learning, self-supervised adaptation, adversarial forecasting, and hybrid threat intelligence integration to enhance long-term robustness against zero-day and rapidly evolving malware threats.

Third, although generative models such as GANs and VAEs have shown potential in mitigating data imbalance, future research must establish rigorous security-focused validation protocols for synthetic malware generation. Evaluations should extend beyond classification improvements to include realism, behavioral fidelity, adversarial safety, and the prevention of synthetic bias introduction.

Fourth, interpretability should be advanced from isolated explainability techniques toward comprehensive human-centered XAI ecosystems. Future malware detection systems should integrate explainable AI directly into detection pipelines and validate interpretability outputs through collaboration with cybersecurity analysts and security operations teams. Such efforts are essential to ensure explanations are actionable, trustworthy, and operationally relevant.

Fifth, the research community should invest in the creation of continuously updated, large-scale benchmark datasets that accurately reflect real-world malware diversity, adversarial evolution, and severe class imbalance. The current dependence on static or outdated datasets limits the practical generalizability of many proposed models.

Sixth, federated and distributed learning paradigms should be further explored to support privacy-preserving, collaborative malware detection across geographically distributed environments. This direction may improve scalability and enable broader collective defense against emerging cyber threats.

Finally, future evaluations of malware detection frameworks should adopt more comprehensive performance criteria beyond conventional accuracy-based metrics. Metrics should include robustness to adversarial drift, minority-class detection performance, interpretability effectiveness, computational efficiency, and deployment scalability to better align academic research with operational cybersecurity demands.

In summary, future malware detection research must transition from isolated performance optimization toward holistic, adaptive, and interpretable defense ecosystems. Addressing these recommendations will be essential for bridging the persistent gap between laboratory success and real-world cybersecurity deployment, while providing the necessary foundation for frameworks such as MAD-FIT to achieve meaningful operational impact.

References

Abdallah, A., Maarof, M. A., & Zainal, A. (2020). Feature Selection and Explainable Intrusion Detection Using SHAP Values. *Journal of Information Security and Applications*, *55*, 102596.

Ajayi, B., Barakat, B., & McGarry, K. (2025). Leveraging VAE-Derived Latent Spaces for Enhanced Malware Detection with Machine Learning Classifiers. *ArXiv Preprint ArXiv:2503.20803*.

Akgündoğdu, A., & Çelikbaş, Ş. (2025). Explainable deep learning framework for brain tumor detection: Integrating LIME, Grad-CAM, and SHAP for enhanced accuracy. *Medical Engineering & Physics*, *144*, 104405. https://doi.org/https://doi.org/10.1016/j.medengphy.2025.104405

Aljurayyil, S., Al-Haj, A., & Farhat, W. (2022). Explainable deep learning for malware detection using SHAP. *ACM Workshop on AI and Security*, 1–10. https://doi.org/10.1145/3564292.3564294

Almajed, H., Alsaqer, A., & Frikha, M. (2025). Imbalance Datasets in Malware Detection: A Review of Current Solutions and Future Directions. *International Journal of Advanced Computer Science and Applications*. https://api.semanticscholar.org/CorpusID:276119764

Alshoulie, M., & Mehmood, A. (2025). Deep Learning Approaches for Malware Detection: A Comprehensive Review of Techniques, Challenges, and Future Directions. *IEEE Access*, *13*, 118652–118677. https://doi.org/10.1109/ACCESS.2025.3582875

Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices. *Computers \& Security*, *89*, 101663.

Aryal, K., Gupta, M., Abdelsalam, M., Kunwar, P., & Thuraisingham, B. (2025). A Survey on Adversarial Attacks for Malware Analysis. *IEEE Access*, *13*, 428–459. https://doi.org/10.1109/ACCESS.2024.3519524

Aslan, Ö., & Samet, R. (2020). A Comprehensive Review on Malware Detection Approaches. *IEEE Access*, *8*, 6249–6271. https://doi.org/10.1109/ACCESS.2019.2963724

Athiwaratkun, B., & Stokes, J. W. (2017). Malware Classification with LSTM and GRU Language Models and a Character-Level CNN. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2482–2486.

Augello, A., Paola, A. De, & Re, G. Lo. (2025a). Hybrid Multilevel Detection of Mobile Devices Malware Under Concept Drift. *Journal of Network and Systems Management*, *33*(2), 36. https://doi.org/10.1007/s10922-025-09906-3

Augello, A., Paola, A. De, & Re, G. Lo. (2025b). M2FD: Mobile malware federated detection under concept drift. *Computers \& Security*. https://doi.org/10.1016/j.cose.2025.103999

Bayer, U., Comparetti, P. M., Hlauschek, C., Kruegel, C., & Kirda, E. (2009). Scalable, Behavior-Based Malware Clustering. *Proceedings of the Network and Distributed System Security Symposium*, 8–11.

Berrios, S., Leiva, D., Olivares, B., Allende-Cid, H., & Hermosilla, P. (2025). Systematic Review: Malware Detection and Classification in Cybersecurity. *Applied Sciences*, *15*(14). https://doi.org/10.3390/app15147747

Brezinski, K., & Ferens, K. (2023). Metamorphic malware and obfuscation: a survey of techniques, variants, and generation kits. *Security and Communication Networks*, *2023*(1), 8227751.

Buriro, A. B., Luccio, F. V., & Yaqub, M. A. B. (2025). Balancing the Scales: Using GANs and Class Balance for Superior Malware Detection. *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, 2032–2039. https://doi.org/10.1145/3672608.3707800

Chakravarty, A. K., Raj, A., Paul, S., & Apoorva, S. (2019). A study of signature-based and behaviour-based malware detection approaches. *Int. J. Adv. Res. Ideas Innov. Technol*, *5*(3), 1509–1511.

Choi, A., Giang, A., Jumani, S., Luong, D., & Di Troia, F. (2023). Synthetic malware using deep variational autoencoders and generative adversarial networks. *EAI Endorsed Transactions on Internet of Things*, *10*.

Cuckoo Sandbox: Open Source Automated Malware Analysis. (2021).

Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G.-G., & Chen, J. (2020). Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics*, *16*(2), 1436–1444.

Cybersecurity Ventures. (2023). *2023 Cybersecurity Almanac: 100 Facts, Figures, Predictions, and Statistics*. https://cybersecurityventures.com/cybersecurity-almanac-2023/

F. Alshmarni, A., & A. Alliheedi, M. (2024). Enhancing Malware Detection by Integrating Machine Learning with Cuckoo Sandbox. *Journal of Information Security and Cybercrimes Research*, *7*(1), 85–92. https://doi.org/10.26735/wzng1384

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with Drift Detection. *Brazilian Symposium on Artificial Intelligence*, 286–295.

Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, *46*(4), 44:1-44:37. https://doi.org/https://dl.acm.org/doi/10.1145/2523813

Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malwGibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and. *Journal of Network and Computer Applications*, *153*(July 2019), 102526. https://doi.org/10.1016/j.jnca.2019.102526

Goodman, B., & Flaxman, S. (2017). European Union Regulations on Algorithmic Decision{-}Making and a ``Right to Explanation''. *AI Magazine*, *38*(3), 50–57.

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access*, *10*, 19572–19585. https://doi.org/10.1109/ACCESS.2022.3151248

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284.

Iadarola, G., Martinelli, F., Mercaldo, F., & Santone, A. (2021). Towards an interpretable deep learning model for mobile malware detection and family identification. *Computers & Security*, *105*, 102198. https://doi.org/https://doi.org/10.1016/j.cose.2021.102198

Jafari, M., & Shameli-Sendi, A. (2026). Evaluating the robustness of adversarial defenses in malware detection systems. *Computers and Electrical Engineering*, *130*, 110845. https://doi.org/10.1016/j.compeleceng.2025.110845

Joshi, C., Kumar, J., & Kumawat, G. (2025). Detection of unseen malware threats using generative adversarial networks and deep learning models. *Scientific Reports*, *15*(1), 34804. https://doi.org/10.1038/s41598-025-18811-3

Khan, S. H., Alahmadi, T. J., Ullah, W., Iqbal, J., Rahim, A., Alkahtani, H. K., Alghamdi, W., & Almagrabi, A. O. (2023). A new deep boosted CNN and ensemble learning based IoT malware detection. *Computers & Security*, *133*, 103385. https://doi.org/https://doi.org/10.1016/j.cose.2023.103385

Ki, Y., Kim, E., & Kim, H. K. (2015). A novel approach to detect malware based on API call sequence analysis. *International Journal of Distributed Sensor Networks*, *11*(6), 659101.

Kim, C., Chang, S.-Y., Kim, J., Lee, D., & Kim, J. (2023). Automated, Reliable Zero-Day Malware Detection Based on Autoencoding Architecture. *IEEE Transactions on Network and Service Management*, *20*(3), 3900–3914. https://doi.org/10.1109/TNSM.2023.3251282

Li, C., Zhiyuan, J., Yongjun, W., Tian, X., Yayuan, Z., & Yuhang, M. (2025). MiniMal: Hard-Label Adversarial Attack Against Static Malware Detection with Minimal Perturbation. In J. Kwok (Ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, {IJCAI-25}* (pp. 5589–5597). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2025/622

Li, J., Xue, D., Wu, W., & Wang, J. (2020). Incremental learning for malware classification in small datasets. *Security and Communication Networks*, *2020*(1), 6309243. https://doi.org/xiang Wang First published: 20 February 2020 https://doi.org/10.1155/2020/6309243

Lopez Pinaya, W. H., Vieira, S., Garcia-Dias, R., & Mechelli, A. (2020). Chapter 11 - Autoencoders. In A. Mechelli & S. Vieira (Eds.), *Machine Learning* (pp. 193–208). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-815739-8.00011-0

Luo, X., Liu, C., Gou, G., Xiong, G., Li, Z., & Fang, B. (2024). Identifying malicious traffic under concept drift based on intraclass consistency enhanced variational autoencoder. *Science China Information Sciences*, *67*(8), 182302. https://doi.org/10.1007/s11432-023-4010-4

Madamidola, O. A., Ngobigha, F., & Ez-zizi, A. (2025). Detecting new obfuscated malware variants: A lightweight and interpretable machine learning approach. *Intelligent Systems with Applications*, *25*, 200472. https://doi.org/10.1016/j.iswa.2024.200472

McFadden, S., Foley, M., D'Onghia, M., Hicks, C., Mavroudis, V., Paoletti, N., & Pierazzi, F. (2025). DRMD: Deep Reinforcement Learning for Malware Detection under Concept Drift. *ArXiv Preprint ArXiv:2508.18839*.

Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware Images: Visualization and Automatic Classification. *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, 1–7.

Nguyen, T., Patel, D., & Singh, R. (2020). Attention-based LSTM for malware behavior detection. *USENIX Workshop on Offensive Technologies (WOOT)*. https://www.usenix.org/conference/woot20

Nikolopoulos, S. D., & Polenakis, I. (2015). A graph-based model for malicious code detection exploiting dependencies of system-call groups. *Proceedings of the 16th International Conference on Computer Systems and Technologies*, 228–235.

Ofusori, L., Bokaba, T., & Mhlongo, S. (2025). Explainability and interpretability of artificial intelligence use in cybersecurity. *Discover Computing*, *28*(1), 241. https://doi.org/10.1007/s10791-025-09760-6

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., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *The BMJ*, *372*. https://doi.org/10.1136/bmj.n71

Panda, B., Bisoyi, S. S., Panigrahy, S., & Mohanty, P. (2025). Machine learning techniques for imbalanced multiclass malware classification through adaptive feature selection. *PeerJ Computer Science*, *11*, e2752.

Park, S., & Lee, K. (2018). Time-aware RNNs for malware sequence modeling. *AAAI Workshop on Artificial Intelligence for Cybersecurity*. https://aaai.org/

Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A. (2015). Malware Classification with Recurrent Networks. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1916–1920.

Patsakis, C., Arroyo, D., & Casino, F. (2025). The Malware as a Service Ecosystem. In D. Gritzalis, K.-K. R. Choo, & C. Patsakis (Eds.), *Malware: Handbook of Prevention and Detection* (pp. 371–394). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-66245-4_16

Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., & Cavallaro, L. (2019). {TESSERACT}: Eliminating experimental bias in malware classification across space and time. *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 729–746.

Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2018). Malware Detection by Eating a Whole EXE. *The Workshops of the Thirty-Second AAAI Conference on Artificial Intelligence*, 268–276. https://doi.org/10.13016/m2rt7w-bkok

Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental Classifier and Representation Learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001–2010.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Roh, E., Kaya, Y., Kruegel, C., Vigna, G., & Hong, S. (2025). MADCAT: Combating Malware Detection Under Concept Drift with Test-Time Adaptation. *ArXiv Preprint ArXiv:2505.18734*.

Sabbah, A., Jarrar, R., Zein, S., & Mohaisen, D. (2025). Empirical Evaluation of Concept Drift in ML-Based Android Malware Detection. *ArXiv Preprint ArXiv:2507.22772*. https://doi.org/https://doi.org/10.48550/arXiv.2507.22772

Sar\i, N. V., & Ac\i, 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*, e3258.

Saxe, J., & Berlin, K. (2015). Deep Neural Network Based Malware Detection Using Two Dimensional Binary Program Features. *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, 11–20.

Shokouhinejad, H., Higgins, G., Razavi-Far, R., Mohammadian, H., & Ghorbani, A. A. (2025). On the Consistency of GNN Explanations for Malware Detection. *Information Sciences*, *721*, 122603. https://doi.org/https://doi.org/10.1016/j.ins.2025.122603

Shokouhinejad, H., Razavi-Far, R., Mohammadian, H., Rabbani, M., Ansong, S., Higgins, G., & Ghorbani, A. A. (2025). Recent advances in malware detection: Graph learning and explainability. *ArXiv Preprint ArXiv:2502.10556*. https://doi.org/https://doi.org/10.48550/arXiv.2502.10556 Focus to learn more

Souza, J. V. S., Vieira, C. B., Cavalcanti, G. D. C., & Cruz, R. M. O. (2025). Imbalanced malware classification: an approach based on dynamic classifier selection. *2025 IEEE Symposium on Computational Intelligence in Security, Defence and Biometrics Companion (CISDB Companion)*, 1–5.

Sun, T., Daoudi, N., Pian, W., Kim, K., Allix, K., Bissyandé, T. F., & Klein, J. (2025). Temporal-Incremental Learning for Android Malware Detection. *ACM Trans. Softw. Eng. Methodol.*, *34*(4). https://doi.org/10.1145/3702990

Tang, A., Sethumadhavan, S., & Stolfo, S. J. (2014). Unsupervised Anomaly-Based Malware Detection Using Hardware Features. In A. Stavrou, H. Bos, & G. Portokalidis (Eds.), *Research in Attacks, Intrusions and Defenses* (pp. 109–129). Springer International Publishing.

Thomas, J., & Harden, A. (2008). Methods for the thematic synthesis of qualitative research in systematic reviews. *BMC Medical Research Methodology*, *8*(45). https://doi.org/10.1186/1471-2288-8-45

Tuan, T. A., Nguyen, P. S., Van, P. N., Hai, N. D., Trung, P. D., Son, N. T. K., & Long, H. V. (2025). A novel framework for cross-platform malware detection via AFSP and ADASYN-based balancing. *Computers and Electrical Engineering*, *128*, 110625. https://doi.org/https://doi.org/10.1016/j.compeleceng.2025.110625

Upender, T., Neelakantappa, M., Rao, C. P., Gera, J., Reddy, V. L., & Yamsani, N. (2025). CyberDetect MLP a big data enabled optimized deep learning framework for scalable cyberattack detection in IoT environments. *Scientific Reports*, *15*(1), 40865. https://doi.org/10.1038/s41598-025-24459-w

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

Zakeri, M., Faraji Daneshgar, F., & Abbaspour, M. (2015). A static heuristic approach to detecting malware targets. *Security and Communication Networks*, *8*(17), 3015–3027.

Zhang, X., Zhao, J., & LeCun, Y. (2021). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, *28*.

Zhang, Y. (2021). *Graph Neural Networks for Malware Detection: Methods and Applications*. University of California, Berkeley.

Zhao, B. (2019). System call dependence graph based behavior decomposition of Android applications. *International Journal of Network Security \& Its Applications (IJNSA) Vol*, *11*.