applied sciences

MDPI

Article

Wavelet-Based and MAML-Driven Framework for Enhanced Few-Shot Malware Classification

Abdullah Almuqrin [1], Ibrahim Mutambik [2,*] and Majed Abusharhah [3]

[1] Department of Information Science, College of Humanities and Social Sciences, King Saud University, Riyadh P.O. Box 11451, Saudi Arabia; aalmogren@ksu.edu.sa
[2] Department of Computer Science and Engineering, College of Applied Studies, King Saud University, Riyadh P.O. Box 11451, Saudi Arabia
[3] Department of Information Science, College Art, Imam Abdulrahman Bin Faisal University, Dammam P.O. Box 2435, Saudi Arabia; mmabusharhah@iau.edu.sa
* Correspondence: imutambik@ksu.edu.sa

Abstract

Traditional malware classification approaches primarily address fixed sets of well-studied malware types and therefore struggle to accommodate the continual emergence of novel or previously unseen malware strains. While visualization-based strategies have shown promise in few-shot malware classification, existing methods often produce representations with limited semantic richness. In parallel, few-shot learning models frequently converge with suboptimal solutions, limiting their ability to generalize effectively to new classes. To address these challenges, we propose MetaWave, a unified framework that jointly optimizes both data representation and model learning for few-shot malware classification. Rather than treating feature representation and learning strategy as largely independent stages, MetaWave is formulated as an explicit representation–adaptation integration framework that combines multi-view malware encoding with meta-learning-based optimization. At the data level, we propose a Wavelet Transform-based Malware Representation method that leverages multi-scale frequency analysis and complementary views to generate semantically enriched representations. At the model level, we adopt Model-Agnostic Meta-Learning (MAML) to optimize model initialization for rapid adaptation to unseen tasks under limited data conditions. Extensive experiments are conducted on two benchmark datasets, EMBER and Malicia, under a 5-way 5-shot protocol with disjoint class splits to ensure evaluation on previously unseen malware families. The proposed framework achieves superior performance, reaching 97.8% accuracy on EMBER and 96.2% on Malicia, consistently outperforming state-of-the-art methods. These results indicate that jointly enhancing representation quality and model adaptability can improve classification accuracy and unseen-family performance under the evaluated 5-way 5-shot protocol. Overall, MetaWave provides an effective framework for few-shot malware classification and offers a promising basis for detecting emerging malware under limited-data conditions, while robustness to adversarial perturbation, obfuscation, and polymorphism remains to be validated through dedicated future evaluation.

Keywords: few-shot learning; malware classification; wavelet transform; meta-learning; MAML

Check for updates

Academic Editors: Aldo Hernandez-Suarez, Jose Portillo-Portillo and Gabriel Sanchez-Perez

Received: 6 March 2026
Revised: 10 April 2026
Accepted: 13 April 2026
Published: 17 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Malware consists of malicious programs deployed and executed on endpoint devices without the user's explicit knowledge or authorization, thereby infringing on the user's legitimate rights. The current proliferation of many types of malware, such as viruses, trojans, and ransomware, poses significant threats to network security [1]. This proliferation results in significant economic burdens for individuals, corporations, and governments alike, and facilitates the leakage of private information [2,3].

Data obtained from AV-TEST for the period 2020–2021 show that there were roughly 113 million malware instances between 2019 and 2020, but this figure rose to 170 million in the following year, marking an almost 30% increase. This dramatic upsurge in new or undiscovered malware underscores a fundamental shortcoming in many existing malware classification models: they struggle to handle novel and previously unseen threats. Two primary factors contribute to this limitation. First, current classification systems depend on datasets containing a predefined range of malware categories, rendering them inadequate for new threats that emerge. Second, such systems typically require large numbers of labeled samples for training, an impractical condition in real-world contexts. For instance, when a new strain of malware appears, only a small set of samples may be available at first. Moreover, manually labeling large repositories of malware consumes considerable time and effort.

In response to these challenges, some researchers have reformulated the malware classification task [4–6] as a few-shot classification problem [7,8], leveraging the ability of learners to assimilate new concepts from only a handful of examples. Known as few-shot malware classification, this paradigm aims to reduce the dependency on extensive datasets, thus making deep learning methodologies more adaptable for real-world applications. As a result, few-shot malware classification has emerged as a crucial and urgent area of study.

Examining visual-analysis-based few-shot classification techniques shows that both the generated malware images and the classification model, collectively referred to here as the data and model levels, exert considerable influence on classification accuracy. Accordingly, enhancing malware classification not only strengthens defenses against network security threats [9,10], but also contributes to advancements in areas such as intelligent networked vehicles [11,12]. However, existing approaches often treat these two aspects independently, without fully exploiting their complementary potential for improving classification performance under limited-data conditions.

One such issue is the complexity that arises from two-dimensional images generated through many visualization strategies. Because each pixel is spatially related to its neighbors, critical information may be lost when resizing these images to meet the dimensional requirements of neural network inputs. Another complication involves converting one-dimensional or multidimensional time-series data into two-dimensional graphs, which preserves temporal continuity but often omits spatial correlation details. Both difficulties lead to incomplete representations of malware images or graphs. Consequently, in few-shot situations characterized by limited samples, extracting more meaningful information becomes essential for improving classification accuracy. Designing malware visualization methods that yield richer and more informative representations is therefore crucial to elevating the presentation of few-shot malware classification.

To address these issues, we propose a new methodology, MetaWave, that tackles challenges at both the data and model levels to facilitate the accurate classification of novel or previously unseen malware. From the data perspective, we posit that varying visualization strategies emphasize distinct aspects of the malware's information. Therefore, this work introduces a Wavelet Transform for Malware Representation procedure. It draws upon the top three visualization strategies (as determined by their performance on both

datasets): grayscale images with adaptive height and width, VMD decomposition graphs, and entropy graphs. By integrating these approaches, the resulting malware representations encapsulate more comprehensive and semantically enriched information.

At the model level, prior research has largely explored improvements in few-shot models through adjustments to the model architecture. By contrast, this study seeks to enhance few-shot model generalization from an optimization standpoint, focusing on how to refine the training process itself. In particular, we explore Meta-Learning with Model-Agnostic Meta-Learning (MAML) as a meta-learning strategy tailored to few-shot learning. MAML optimizes the model for rapid adaptation to new tasks by improving its initial parameter settings, thereby enhancing generalization and bolstering classification performance when encountering unfamiliar or newly emerging malware. This dual-level design enables MetaWave to jointly improve feature representation quality and model adaptability, addressing key limitations in existing few-shot malware classification approaches.

We conduct a wide-ranging set of experiments on two publicly available datasets in the few-shot malware classification domain, offering a comparative evaluation of four selected state-of-the-art visualization-based malware analysis methods and four prominent few-shot learning models. Our results attest to the efficacy of our proposed strategy. Compared with prior investigations, the main advantages of our approach can be summarized as follows:

1. A focused empirical investigation is conducted. We recreate and assess the performance of four distinct malware visualization approaches, Bigram image, Entropy graph, and VMD decomposition graph, under limited-data settings. Additionally, we compare the classification outcomes of four few-shot algorithms at the model level: ProtoNet, RelationNet, MatchNet, and CAN. Based on these results, we devise a new methodology that reliably enhances classification performance through the integration of advancements at both the model and data levels.
2. The Wavelet Transform for Malware Representation approach is presented. This technique is based on multi-view concepts, thus allowing the representations generated by the method to obtain more detailed information about malware executables; this is of particular benefit in few-shot malware classification.
3. MAML is examined as a meta-learning framework for few-shot malware classification. This boosts the generality of the model in contexts with limited data via the optimization of the initial parameters for adaptability across tasks.

Our experiments conducted on two few-shot malware datasets demonstrate that the aimed approach achieves superior classification accuracy compared to baseline methods, with peak performance rates of 97.8% and 96.2% for the five-way five-shot classification task, respectively.

The remainder of this paper is organized as follows. Section 2 reviews related work on malware analysis techniques, visualization methods, and few-shot learning approaches. Section 3 presents the problem formulation and preliminaries. Section 4 introduces the proposed MetaWave framework, including the multi-view representation and meta-learning optimization strategy. Section 5 describes the experimental setup and evaluates the performance of the proposed approach. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Related Work

This section offers a succinct review of three interconnected research domains: (1) malware classification grounded in visualization analysis, (2) few-shot image classification, and (3) few-shot malware classification.

2.1. Visualization-Based Techniques for Malware Classification: A Comprehensive Review

Visualization-based malware classification approaches can be broadly grouped into two categories. The first type creates a two-dimensional representation from the malware's binary executable in either the spatial or spatial-frequency area, in order to capture the specific patterns of the malware. In the second technique, the binary file is converted for a one-dimensional time series, then another approach, such as change point detection, entropy analysis, signal decomposition, or time-frequency analysis, is used [13]. By exploiting visual similarities gleaned from these representations, researchers uncover shared hidden structures that enable more effective malware classification.

Sethuraman [14] was among the first to conceptualize malware binaries as two-dimensional grayscale images, using the Gist algorithm to extract texture-based descriptors. This approach proved particularly robust against obfuscation, encryption, and packing. Expanding on this work, Taşcı et al. [15] introduced a deep learning-driven method for spotting malware variants. They determined the image dimensions based on the file size to yield a grayscale image with variable dimensions, which was then downsampled into a fixed-size square image. A convolutional neural network (CNN) was subsequently employed for classification. Simultaneously, the bat algorithm was leveraged to alleviate data imbalance issues.

Yuan et al. [16] proposed a malware classification framework (MDMC) using Markov images, whereas Sharma et al. [17] refined the process of Markov image generation by employing Colormap to produce both grayscale and RGB representations of malware. They then extracted discriminative features from these images using Gabor filters, subsequently testing classification performance across 12 distinct neural networks. Additional features derived from the malicious binary, such as pixel histograms, block entropy, and the mean and standard deviation of block entropy, were also employed in a seven-layer deep neural network for malware classification.

Vasan et al. [18] offered IMCFN, a malware classification system grounded on images. In this approach, the binary file was first converted into a 2D grayscale image, and then a colormap was applied to produce an RGB variant. Models pretrained on ImageNet (e.g., VGG16) were fine-tuned on the malware dataset, and data augmentation methods were introduced to mitigate class inequity. Similarly, Kumar et al. [19] partitioned malware files according to their Portable Executable (PE) format, deriving values such as entropy, byte distributions, and relative size. An RGB image was then created using these values in which the colored channels were populated by the corresponding features of each segment. Classification was carried out by fusing global texture and color features with local Simhash-based descriptors.

More recently, Bijitha et al. [20] introduced the idea of generating bigram images from the spatial frequency domain representation of malicious binaries using a bigram algorithm. They then applied the discrete cosine transform (DCT) to produce bigram-DCT images, which were finally classified via convolutional neural networks.

In addition to these two-dimensional strategies, some researchers opt to represent binaries as one-dimensional time series, enabling further analysis via entropy metrics, time–frequency decomposition, change point detection, or signal decomposition. These methods each highlight distinct facets of malicious files [21]. Since attackers often disguise harmful code sections through encryption or compression, entropy analysis has become particularly valuable. For instance, Lyda and Hamrock [22] employed entropy analysis to distinguish compressed or encrypted segments within malicious executables, capitalizing on entropy values ranging from 0 to 8. Gibert et al. [23] transformed file content into an entropy-based time series and utilized shapelet-based time series classification to identify local distinguishing features for malware detection. Kim et al. [24] and Mutambik [25]

suggested a file-agnostic deep learning solution, recasting malware classification as a time series classification task.

Han et al. [26] converted malware binaries into bitmap images and calculated per-row entropy values, constructing an entropy graph to assess the similarity among files. Xiao et al. [27] introduced MalFCS, a three-stage framework that first generates an entropy stream from malware binaries, applies a deep CNN to automatically learn shared family features from the entropy graph, and finally uses an SVM classifier for detection. Nataraj and Manjunath [28] pioneered an "audio-like" one-dimensional signal representation of malicious binaries, offering an orthogonal perspective to 2D approaches. Expanding on this concept, Nataraj et al. [29] proposed OMD, a method of converting malware binaries into audio signals. Audio descriptors are then computed, which are integrated with image likeness and other structures. With regard to change point detection, the malware spectrum visualization mechanism SpecView was introduced by Han et al. [26]; this method relies on singular spectral transformation, and has since been further enhanced through particle swarm optimization for improved performance [30,31].

Multiple visualization-based malware classification methods have been compared in the literature, though often with limited scope. Nataraj et al. [29] assessed variable-dimension grayscale images against their RGB counterparts. Xiao et al. [27] contrasted grayscale and RGB images derived via Colormap with Markov-based images. Alshomrani et al. [32] compared variable-dimension grayscale images, bigram images, and bigram-DCT images within the frequency domain, also exploring a feature-fusion scheme. Han et al. [26] evaluated the effectiveness of the SST spectrum across six machine learning algorithms and made comparisons between the results from this approach with those from other classification techniques.

Nonetheless, these comparative studies do not comprehensively survey the full breadth of contemporary visualization strategies, generally addressing only up to three methods at a time. Moreover, most of these analyses center on large-scale malware classification and therefore may not be applicable to few-shot learning settings. Hence, our work constitutes the first known attempt to systematically collect 11 different visualization techniques and compare them within a few-shot context. Additionally, since the majority of visualization-based solutions adopt a single-information perspective, we propose a Wavelet Transform for Malware Representation approach from a multi-view standpoint. Specifically, we incorporate methods that consistently rank within the top three among all eleven evaluated visualization strategies on two datasets: variable-dimension grayscale images, entropy graphs, and VMD decomposition graphs.

2.2. Strategies for Few-Shot Image Classification: Optimization and Metric Learning

Few-shot image classification endeavors to enable machines to replicate human-like capabilities in identifying and generalizing novel objects or categories based on only a limited set of labeled examples. In particular, given a small collection of samples for new or unidentified classes, few-shot methods seek to accurately classify previously unseen data. Broadly, existing approaches in few-shot classification may be grouped into two primary categories: those founded on optimization-based methods and those that employ metric learning techniques.

The essential objective of optimization-oriented approaches is to acquire a robust initialization for a classification model. Through gradient-based techniques, the learned weights of the model can be swiftly updated for new classes using only a handful of labeled instances. Representative methods in this category encompass MAML [33]. MAML, in particular, has gained prominence due to its ability to learn a parameter initialization that can be efficiently adapted to new tasks with only a few gradient updates, thereby

improving generalization under limited-data conditions and making it highly effective in few-shot learning scenarios.

In contrast, methods grounded in metric learning prioritize the extraction of effective feature representations and the design of suitable similarity measures. After being trained, the learned parameters are typically left unchanged during inference for a new task, leading to high-speed prediction. Within metric-based strategies, two prominent paradigms exist: a single-stage paradigm that relies on episodic training, and a two-stage paradigm that involves both pretraining and episodic training.

El Youssofi et al. [34] are examples of single-stage-paradigm approaches. The training for these methods exclusively involves episodic sessions. For example, Tian et al. [35] demonstrated how the incorporation of triplet loss enhances metric learning solution effectiveness. In addition, Aljarrah et al. [36] found that a misalignment of the attention mechanisms between query and support images can occur, proposing the use of a cross-attention element to overcome this problem. Further, in order to improve generalization, Guo et al. [37] developed an auxiliary self-supervised objective, while Wang et al. [38] suggested the use of intra-and inter-image relationship modeling to address few-shot recognition from a relational standpoint via Autocorrelation Representation (SCR) and Cross-Related Attention (CCA) modules.

Regarding the two-phase framework, Rezaei et al. [39] successfully integrated few-shot learning (FSL) with the self-attention structure of transformers, thereby attaining notable performance gains. Likewise, Tian et al. [35] reframed one-shot classification as an optimum matching challenge and harnessed the Earth Mover's Distance (EMD) algorithm as a distance measure to improve classification outcomes. Meanwhile, Zhang [40] employed active learning techniques to further refine few-shot classification methodologies, highlighting the importance of iterative feedback in model enhancement.

2.3. Advancements in Few-Shot Malware Classification Methods

In order to overcome the problem of limited considered data in malware classification, a one-shot classification method was proposed by Awan et al. [41] that utilizes a Siamese network. Building on this concept, Singh and Singh [42] trained a streamlined Multilayer Perceptron (MLP) within a Siamese architecture, embedding malware instances into real-valued feature vectors for the purpose of discerning whether malware samples originate from the same or different families. Tran et al. [43] introduced a memory-augmented solution for detecting novel malware variants, whereas Alavizadeh et al. [44] demonstrated that, by employing a Siamese neural network, variant classification can be achieved even when only a few training examples are available. Further advancing few-shot techniques, Khandhar [45] suggested combining Markov images, entropy graphs, and grayscale matrices into "GEM" images, which facilitate limited-sample malware classification. Meanwhile, Maniriho et al. [46] presented a Siamese neural network approach enhanced by entropy-based features, enabling expedited and accurate ransomware detection using only small sample sets.

Despite these endeavors, the aforementioned models typically rely on scant data per malware family and necessitate retraining whenever new or unfamiliar classes emerge. By following the N-way K-shot paradigm central to meta-learning, Gibert et al. [47] proposed an attention-driven transductive learning network. This framework proceeds in three major steps: creating a grayscale image from the binary executable of a specific malware; using a convolutional neural network to extract representations while developing an attention-based Gaussian; and transferring labeling data from a support set to a query set via a label propagation procedure. Another approach, known as the ConvProtoNet model, was explored by Conti et al. [45] for few-shot malware classification. The method includes four

main parts: a malware visualization stage, in which binary code is converted into square grayscale images; an embedding element, which maps the generated images into a feature domain; a convolutional induction module, which creates class prototypes that can handle multimodal distributions; and a SoftMax-based classifier, which incorporates modified cosine similarity for the final classification. In a different vein, Namita et al. [48,49] introduced a dynamic analysis-based few-shot classification framework that uses N-gram and TF-IDF to glean features from API sequences, and then generates multiple prototypes per class to match query samples with the nearest prototype, thus completing the classification process.

Overall, Gibert et al. [47] conducted a comparative assessment of classic few-shot models (e.g., MatchNet, RelationNet, and ProtoNet) in an image-based malware classification context. Conti et al. [45] likewise evaluated six few-shot techniques—SNALL, ProtoNet, and HABPN. Unlike these methods, which typically represent malware through images or graphs, the present work specifically focuses on comparing image-based few-shot classification methods. Furthermore, most existing approaches, both in image-based classification and malware classification, optimize only the training loss, potentially limiting the model's capacity to generalize. Hence, we propose incorporating MAML during model training. By optimizing the initial parameters to enable rapid adaptation across tasks, we aim to enhance the few-shot model's overall generalization capability. Additionally, this study introduces Wavelet Transform for Malware Representation at the data level, enabling more semantically enriched representations of malware. Together, these techniques form the foundation of the proposed MetaWave framework, which addresses both data and model-level challenges in few-shot malware classification.

3. Problem Scope and Meta-Learning Framework for Few-Shot Malware Classification

Few-shot malware classification endeavors to identify malware samples within classes where each class has only a handful of labeled examples. A predominant strategy for this task is to perform a meta-learning framework utilizing episodic training to build a robust few-shot classification model. This meta-learning process comprises two primary stages: meta-training and meta-testing.

In the meta-training stage, the labeled training dataset *Dtrain* (associated with label set *Ltrain*) is divided into multiple meta-tasks, each intended to cultivate the model's aptitude for handling shifting classes. Subsequently, the validation dataset *Dval* with label set *Lval* is used to fine-tune or validate the model, and finally, the testing dataset *Dtest* with label set *Ltest* measures the model's ultimate performance. Note that the three label sets *Ltrain*, *Lval*, and *Ltest* do not intersect, satisfying $Ltrain \cap Lval \cap Ltest = \varnothing$.

Within *Dtrain*, *Dval*, and *Dtest*, the data are arranged into multiple episodes. Each episode generates various meta-tasks, each containing a support set:

$$S = \{(Is(l), ys(l))\}l = 1N \times K \quad (1)$$

and a query set:

$$Q = (Iq, yq) \quad (2)$$

where *Sk* denotes the subset of the $k - th$ class, and $k \in \{1, 2, \ldots, N\}$. To construct the provision set *S*, one typically selects *N* classes at random from *Dtrain* and then samples *K* malware examples per class, thus yielding $N \times K$ total training instances. The query set *Q* comprises data from the same *N* classes, but includes samples not used in the support set.

In an *N*-way *K*-shot scenario, the model must differentiate among *N* classes despite having only *K* labeled examples for each class. During the meta-training phase, the model repeatedly learns how to map (*S*, *Iq*). Once this training is complete, the meta-testing

phase begins. At this stage, the model's learned mapping is directly applied to classify a novel inquiry sample q into one of the N classes represented in the provision set SSS drawn from $Dtest$.

Because the meta-training stage partitions data into numerous meta-tasks, the model gains robust generalization capabilities even if future classes differ from those observed during training. As a result, when entering the meta-testing phase, the model can accurately classify new classes without revising or fine-tuning its existing parameters.

4. Methodology

This section presents MetaWave, a few-shot malware classification framework that integrates a structured multi-view representation pipeline with task-adaptive metric learning. The framework consists of four connected components: (1) a malware representation module that transforms each executable into a unified multi-view tensor, (2) an embedding module that maps the tensor into a latent feature space, (3) a prototype construction module that summarizes each class in the support set, and (4) a task-adaptive classification module that performs episodic optimization and query prediction. Given a malware sample, MetaWave first converts the binary executable into three aligned views derived from wavelet, entropy, and time-frequency analysis. These views are then fused into a multi-channel tensor and passed to an embedding network. The resulting support embeddings are averaged class-wise to form prototypes, and query samples are classified according to their distances to these prototypes after meta-learned adaptation.

The methodological contribution of MetaWave does not lie in introducing an entirely new standalone classifier or a new generic meta-learning algorithm in isolation. Rather, its contribution lies in the explicit coupling of representation construction and adaptation-oriented learning for few-shot malware classification. Existing studies commonly rely on either a single visualization strategy or a standard few-shot model applied after feature generation. In contrast, MetaWave is designed so that the representation stage preserves complementary structural information, while the learning stage optimizes the embedding parameters for rapid adaptation under sparse-support conditions. This representation–adaptation integration constitutes the essential methodological contribution of the framework.

The choice of wavelet, entropy, and time-frequency views is theoretically motivated. Wavelet analysis captures multi-scale frequency structure, entropy characterizes local randomness and byte-level complexity, and time-frequency analysis preserves the evolution of spectral content along the malware signal. Their fusion is intended to reduce representation incompleteness under limited-data conditions. The embedding and optimization stages are then formulated consistently on top of this unified representation so that the learned feature space is not only discriminative, but also adaptable across episodic few-shot tasks.

4.1. Wavelet Transform for Malware Representation

In few-shot scenarios, where only a limited number of labeled malware samples are available per class, the quality of input representation becomes critical for classification performance. Accordingly, MetaWave begins with a data-level pipeline that converts each malware executable into a structured multi-view tensor. Let a malware binary be represented as a one-dimensional byte signal:

$$s = [s_1, s_2, \ldots, s_n] \in \mathbb{R}^n \tag{3}$$

where n denotes the signal length and each element s_i corresponds to the numerical value of one byte extracted from the executable. This signal forms the common input for all subsequent representation branches.

Unlike single-view approaches, MetaWave adopts a principled multi-view representation strategy in which each branch is selected according to its complementary role in characterizing malware behavior. Specifically, wavelet-based features capture multi-scale frequency patterns, entropy-based features represent local randomness and structural irregularity, and time-frequency features preserve localized spectral evolution. These branches are not introduced as a heuristic aggregation of unrelated views; rather, they are organized as a structured representation layer whose output is a unified tensor used consistently throughout prototype learning and meta-optimization.

4.1.1. Wavelet-Based Feature Extraction

To construct the first representation branch, the byte signal $\mathbf{s}$ is processed using a discrete wavelet transform (DWT). Let $\psi(\cdot)$ denote the selected mother wavelet and let L denote the number of decomposition levels. The DWT decomposes $\mathbf{s}$ into approximation and detail coefficients across multiple resolutions. The resulting wavelet coefficient set is written as:

$$W(s) = \{c_1, c_2, \ldots, c_L\} \tag{4}$$

where $c^{(l)}$ denotes the coefficient vector at level l, for $l = 1, \ldots, L$. These coefficients encode frequency characteristics at different scales and therefore preserve structural variations that may be informative for malware discrimination.

The resulting frequency-based components are aggregated to create a rich and semantically meaningful representation of the malware, which is subsequently reshaped into a two-dimensional feature map:

$$X_w \in \mathbb{R}^{H \times W} \tag{5}$$

where H and W denote the spatial dimensions used by the downstream embedding network. Thus, X_w represents the wavelet-derived view of the malware sample.

4.1.2. Entropy Graphs as a Complementary Feature

To complement the frequency-oriented wavelet view, MetaWave constructs an entropy-based representation that captures local randomness and byte-distribution complexity. The byte signal is partitioned into non-overlapping blocks $\{b_1, b_2, \ldots, b_m\}$, and the Shannon entropy of each block b_i is computed as:

$$H(b_i) = -\sum_j p_j \log p_j \tag{6}$$

where p_j denotes the probability of observing byte value j within block b_i, satisfying $\sum_\{j\}$ $p_j = 1$.

The entropy graph is rendered as a continuous series for inclusion in the representation and is further normalized and transformed into a two-dimensional feature map through interpolation or reshaping to ensure dimensional consistency.

$$X_e \in \mathbb{R}^{H \times W} \tag{7}$$

This branch emphasizes local irregularities and complexity patterns that are often obscured in purely frequency-based encodings.

4.1.3. Time-Frequency Representation for Enhanced Semantics

To further enrich the representation, a time-frequency transform $\mathcal{T}(\cdot)$ is applied to the same input signal s. The output is a two-dimensional time-frequency map

$$X_t = \mathcal{T}(s) \in \mathbb{R}^{H \times W} \tag{8}$$

which captures how spectral content evolves across the malware signal. This view preserves localized temporal-frequency structure and provides information that is complementary to both the wavelet and entropy branches.

4.1.4. Feature Alignment and Fusion

To ensure compatibility across different representations, all feature maps are normalized and resized to a consistent spatial resolution:

$$(H \times W) \tag{9}$$

The final unified representation is constructed through channel-wise concatenation of the three feature maps:

$$X_{\text{fused}} = \text{Concat}(X_w, X_e, X_t) \in \mathbb{R}^{H \times W \times 3} \tag{10}$$

where $\text{Concat}(\cdot)$ denotes concatenation along the channel dimension. This operation preserves the complementary information contained in the three branches while presenting the downstream model with a standardized input tensor. The fused tensor X_{fused} is therefore the sole representation consumed by the embedding network during both training and inference.

4.1.5. End-to-End Representation Pipeline

Overall, given a malware binary, the proposed data-level pipeline converts it into a one-dimensional signal, extracts three complementary feature representations, and fuses them into a unified tensor representation. This process ensures that both frequency-domain and structural characteristics of the malware are preserved, thereby enhancing representation quality in few-shot learning scenarios.

Each representation component in this module is designed to capture a distinct aspect of malware characteristics. The wavelet transform captures multi-scale frequency patterns, entropy-based representations model local randomness and structural complexity, and time-frequency analysis preserves temporal variations in spectral content. The integration of these components is essential to construct a more complete and informative representation, particularly under limited-data conditions where single-view representations are insufficient.

4.2. Feature Embedding and Prototype Calculation in MetaWave

After representation fusion, MetaWave maps the unified tensor into a latent feature space suitable for prototype-based few-shot classification. Let $f_\theta(\cdot)$ denote the embedding network with trainable parameters θ. For an input fused tensor $X_{\text{fused}} \in \mathbb{R}^{H \times W \times 3}$, the network produces an embedding:

$$z = f\theta(Xfused) \in Rd \tag{11}$$

where d denotes the embedding dimension. For clarity, the input tensor itself is defined as:

$$X_{\text{fused}} \in \mathbb{R}^{H \times W \times 3} \tag{12}$$

The embedding network consists of convolutional layers that learn hierarchical spatial and cross-channel dependencies from the fused representation. Its function is to project malware samples belonging to the same class into nearby regions of the feature space while separating samples from different classes. Because the network operates directly on

the fused tensor, the resulting embedding is learned jointly from wavelet, entropy, and time-frequency evidence rather than from any single dominant view.

Consider an episodic task with support set:

$$\mathcal{S} = \{(\mathrm{X}_i, y_i)\}_{i=1}^{NK} \tag{13}$$

where N is the number of classes and K is the number of labeled support samples per class. Let $\mathcal{S}_k \subseteq \mathcal{S}$ denote the subset of support samples whose label is class k. Following the prototype-learning principle of Prototypical Networks, the prototype of class k is computed as the mean of its support embeddings:

$$P_k = \frac{1}{|S_k|} \sum_{x_i \in S_k} E(X_{\text{fused}}^{(i)}) \tag{14}$$

Embedding of Support and Query Samples: The embedding module E ingests both the support set S and a query instance q, thereby producing two outputs:

$$Q' = E(X_{\text{fused}}^{(q)}) \tag{15}$$

Here:

$X_{\text{fused}}^{(i)}$ denotes the fused multi-channel tensor corresponding to the i-th support sample. $X_{\text{fused}}^{(q)}$ represents the fused tensor of the query sample.

Feature Representation Learning: The embedding function $E(\cdot)$ learns a high-dimensional feature space in which samples belonging to the same class are clustered around their corresponding prototypes, while samples from different classes are well separated.

This transformation is critical for few-shot learning, as it enables robust classification using only a small number of labeled examples. By leveraging the fused multi-view representation, the embedding module enhances the discriminative power of the learned features, particularly under limited-data conditions.

Role in MetaWave Framework: The embedding module plays a crucial role in transforming the Wavelet Transform-based malware representations into a high-dimensional feature space where meaningful prototypes for each class can be constructed.

By operating on the fused representation X_{fused}, the embedding module ensures that information from wavelet, entropy, and time-frequency domains is jointly encoded. This leads to more expressive class prototypes that capture both structural and frequency-domain characteristics of malware samples.

Consequently, the MetaWave framework benefits from improved feature separability and enhanced generalization, which are essential for accurate classification in few-shot scenarios.

4.3. Model Optimization and Inductive Inference

This subsection describes the model-level optimization strategy and inference mechanism of the MetaWave framework, which together enable robust few-shot classification. Unlike conventional approaches that apply embedding directly on single-view inputs, the proposed embedding module is designed to operate on fused multi-view representations, enabling the model to learn cross-domain feature interactions. This enhances the discriminative capability of the learned feature space and allows the model to better exploit complementary information across different representation types.

The classification module comprises a model training stage and an inductive inference stage. During training, we emphasize model optimization using MAML [33], a meta-learning strategy tailored for few-shot contexts.

Importantly, the integration of MAML within MetaWave is not a standalone choice; rather, it is explicitly aligned with the proposed multi-view representation. The fused representation is designed to provide rich and complementary features that improve the effectiveness of task-adaptive optimization, thereby enabling faster and more reliable adaptation to unseen malware classes. This coordinated design distinguishes MetaWave from prior approaches where representation learning and optimization are treated independently.

Unlike conventional training strategies that directly minimize empirical loss, MAML explicitly optimizes the model parameters for rapid adaptation to new tasks. This is achieved by learning an initialization that can be efficiently updated with a small number of gradient steps.

Specifically, the model is trained across episodic tasks to learn an initialization that can be efficiently fine-tuned with a small number of gradient updates.

Formally, given a task $\mathcal{T}_i$, MAML performs an inner-loop update:

$$w_i' = w - \alpha \nabla_w \mathcal{L}_{\mathcal{T}_i}(w) \tag{16}$$

where w represents the model parameters and α is the inner-loop learning rate. The meta-objective is then defined as:

$$\min_w \sum \mathcal{T}_i \mathcal{L}_{\mathcal{T}_i} \left(w_i' \right) \tag{17}$$

This optimization ensures that the learned parameters w are sensitive to task-specific updates while maintaining generalization across tasks.

It is important to note that MAML is model-agnostic and operates independently of the input representation; therefore, it can be applied to various feature representations beyond the proposed wavelet-based multi-view encoding. The final few-shot classifications and model assessments are carried out in the inductive inference phase.

4.3.1. Optimization-Driven Training for Few-Shot Classification

After the class prototypes have been obtained from the support set, each query sample is assigned by a nearest neighbor classifier to one of the N classes in the support set according to a specified similarity measure. Concretely, for each query feature:

$$Q' = E(X_{\text{fused}}^{(q)}) \tag{18}$$

The nearest neighbor classifier produces a SoftMax-based probability distribution across the N class prototypes. The likelihood that Q' corresponds to class k is given by:

$$p(y = k|Q') = \frac{\exp(-d(Q', P_k))}{\sum_{j=1}^{N} \exp(-d(Q', P_j))} \tag{19}$$

where $d(\cdot, \cdot)$ denotes the selected distance metric in the embedding space. Accordingly, the classification loss is defined as the negative log-likelihood of the true class label:

$$\mathcal{L}_{\text{train}}(w) = -\sum_i \log p(y_i \mid Q_i') \tag{20}$$

Under standard supervised training, one could directly minimize $\mathcal{L}_{\text{cls}}(\theta)$ using an optimizer such as Adam.

In conventional practice, the Adam optimizer [50] is employed to minimize $\mathcal{L}_{\text{train}}(w)$. However, such standard training may not ensure optimal generalization for few-shot scenarios.

By integrating MAML into this framework, the training process optimizes the model parameters for adaptability rather than solely minimizing training loss. This enables the model to quickly adjust to new classes using only a limited number of samples, which is critical in few-shot malware classification.

4.3.2. Query Classification and Prototype Matching

During inductive inference, the learned parameters are fixed. For a new task, the support samples are first embedded and averaged to form prototypes according to Equation (11). A query sample X_q is then embedded as $z_q = f_\theta(X_q)$, and its predicted label is obtained by selecting the nearest prototype:

$$\hat{y}_q = \arg\max_k d(Q', P_k) \tag{21}$$

This decision rule is consistent with the probability model in Equation (19), since maximizing the SoftMax posterior is equivalent to minimizing the distance to the corresponding class prototype. The inference phase therefore uses exactly the same embedding definition and prototype construction rule as the training phase, ensuring full consistency between optimization and deployment.

4.3.3. Limitations of Mean Embedding and Mitigation Strategy

While prototype-based classification using mean embeddings provides an efficient and widely adopted mechanism in few-shot learning [33], it introduces several representational bottlenecks, particularly under cold-start and highly sparse data conditions.

Specifically:

- The use of mean embeddings assumes unimodal class distributions
- It is sensitive to noise and outliers in small sample sets
- It may obscure subtle discriminative features

These limitations are particularly relevant in few-shot malware classification, where newly emerging malware variants often exhibit heterogeneous structures and limited labeled instances.

To address these challenges, the proposed MetaWave framework enhances representation quality through multi-scale wavelet-based feature extraction, which captures richer semantic and frequency-domain characteristics.

Furthermore, the integration of MAML enables the embedding space to be optimized for task adaptability, allowing the model to better accommodate intra-class variability and improve generalization.

Consequently, although mean embeddings are retained for computational efficiency, their limitations are mitigated through the joint design of multi-view representation and task-adaptive optimization. This integration constitutes a key methodological contribution of MetaWave and differentiates it from prior approaches that rely on independent or loosely coupled components.

5. Experiments

To ensure clarity and consistency, all experimental results in this section are reported under a standardized evaluation protocol, including fixed dataset splits, a 5-way 5-shot episodic configuration, and mean classification accuracy with 95% confidence intervals. Unless otherwise stated, the same evaluation setting is used throughout this section.

In this section, we rigorously evaluate the performance of our newly developed few-shot malware classification framework, MetaWave. We demonstrate the efficacy of our method by comparing it against contemporary state-of-the-art approaches.

5.1. Experimental Setup

Datasets: We utilize the EMBER Dataset [51,52] as one of the primary datasets for evaluation. The EMBER dataset comprises over one million Portable Executable (PE) files labeled as malicious or benign. Because the objective of this study is few-shot malware family classification rather than binary malicious/benign detection, we first construct a malware-only subset and then group the retained samples according to malware family labels. The resulting benchmark is therefore organized as a family-level multi-class few-shot classification task. The dataset is divided into 100 classes for training, 58 for validation, and 50 for testing, ensuring non-overlapping label sets across meta-training, meta-validation, and meta-testing phases. No malware family is shared across these three subsets. Accordingly, all testing episodes are sampled exclusively from meta-test classes that are not observed during meta-training, thereby enabling evaluation under unseen-family conditions. Additionally, we utilize the Malicia Dataset [53,54], which contains real-world malware samples collected from drive-by download attacks. Similar to EMBER, we preprocess the dataset to construct a multi-class malware classification task by grouping samples based on malware families. The dataset is divided into 100 classes for training, 58 for validation, and 50 for testing, ensuring non-overlapping label sets across meta-training, meta-validation, and meta-testing phases. As in the EMBER benchmark, these class partitions are mutually disjoint, so the meta-test evaluation is carried out only on previously unseen malware families. This protocol is intended to assess performance under an unseen-family few-shot setting; however, broader generalization across multiple way/shot configurations or cross-dataset transfer settings is beyond the current experimental scope.

Experimental Setting: We adopt a standard 5-way 5-shot evaluation protocol. In each episode, 5 classes are sampled from the corresponding meta-test split, and 5 labeled support samples are selected for each class. For each episode, 10 query samples per class are selected, resulting in 50 query samples per episode. During testing, 500 episodes are randomly generated from the test set. All reported methods are evaluated using the same episode construction rule, the same class partitions, and the same number of test episodes to ensure fair comparison. Performance is reported as mean classification accuracy (%) with 95% confidence intervals, computed across all test episodes. This standardized configuration ensures fair comparison and robust evaluation. Since the present study is centered primarily on the 5-way 5-shot setting, the reported results support claims of effectiveness within this specific evaluation protocol.

Data-Level Approaches: To evaluate the effectiveness of malware visualization techniques, we compared the proposed Wavelet Transform for Malware Representation against 11 established methods, each offering unique strategies for transforming binary executables into visual or analytical representations. Among these candidate methods, Tables 1 and 2 report the representative results for the four strongest and most relevant visualization strategies retained for detailed comparative analysis in the final benchmark. The Variable-Width and Height Grayscale Image approach [55,56] organizes binary content into a two-dimensional array, where the image's width depends on the file size and the height adjusts, hence producing a simple yet effective grayscale visualization.

Table 1. Enhanced Few-Shot Malware Classification Performance using EMBER Dataset.

Model	Backbone	Variable-Width and Height Grayscale Image	Bigram Image	Entropy Graph	VMD Decomposition Graph
ProtoNet	Conv4	97.0 ± 0.23	94.0 ± 0.35	95.5 ± 0.30	96.2 ± 0.25
RelationNet	Conv4	92.0 ± 0.55	89.0 ± 0.40	91.2 ± 0.36	90.5 ± 0.45

Table 1. *Cont.*

Model	Backbone	Variable-Width and Height Grayscale Image	Bigram Image	Entropy Graph	VMD Decomposition Graph
MatchNet	Conv4	90.0 ± 0.24	87.0 ± 0.50	89.7 ± 0.32	88.6 ± 0.30
CAN	Resnet12	94.0 ± 0.25	90.0 ± 0.28	92.4 ± 0.35	91.8 ± 0.33

Table 2. Enhanced Few-Shot Malware Classification Performance using Malicia Dataset.

Model	Backbone	Variable-Width and Height Grayscale Image	Bigram Image	Entropy Graph	VMD Decomposition Graph
ProtoNet	Conv4	95.0 ± 0.20	91.0 ± 0.30	93.0 ± 0.25	94.2 ± 0.22
RelationNet	Conv4	89.0 ± 0.45	87.0 ± 0.38	89.2 ± 0.35	88.5 ± 0.40
MatchNet	Conv4	87.0 ± 0.30	85.0 ± 0.40	87.5 ± 0.32	86.2 ± 0.28
CAN	Resnet12	92.0 ± 0.22	88.0 ± 0.27	90.4 ± 0.30	89.8 ± 0.25

Frequency-domain techniques are also explored, such as bigram and bigram-DCT images [55], which process binaries as 16-bit hexadecimal vectors. A bigram algorithm computes pixel intensities for grayscale visualization, while a discrete cosine transform (DCT) further refines the representation into a frequency-domain equivalent. A distinct approach is employed in Markov images [55,57], which calculates the transition probabilities between adjacent byte values to construct transition probability matrices, effectively capturing sequential patterns in malware binaries. Enhancing these methods further, RGB images based on Colormap [58,59] convert variable-width grayscale images into multi-channel RGB images, providing richer visual features for classification.

Entropy-based methods, such as the entropy graph [60,61], analyze the binary file as a hexadecimal byte stream or time-series signal. This approach divides the signal into fixed-length blocks and applies the Shannon entropy formula to compute the randomness within each block, producing a graph that highlights structural and random patterns. Another time-series technique, the SST spectrum [47,62], employs fast Fourier transform (FFT) to resample the binary data and applies the Synchrosqueezing Transform (SST) to extract spectral features that reveal characteristic changes over time. In contrast, the VMD decomposition graph [63,64] uses variational mode decomposition to split the malware signal into sub-signals, each corresponding to distinct frequency components, yielding a detailed visualization of the binary's frequency characteristics.

Finally, the proposed Wavelet Transform for Malware Representation provides a robust and semantically rich alternative to these methods. By leveraging wavelet decomposition, this approach captures both frequency and time-domain information, enabling the creation of multi-scale representations that reveal critical features of the malware. Rather than claiming universal superiority across all settings, we evaluate this representation under the standardized 5-way 5-shot protocol adopted in this study. Specifically, in the experimental pipeline, malware binaries are first converted into one-dimensional signals, after which a discrete wavelet-based decomposition is applied to extract multi-scale frequency components. The resulting coefficients are aggregated and transformed into two-dimensional feature maps, which are subsequently normalized and fused with entropy and time-frequency representations as defined in Section 4.1. This unified multi-channel representation is consistently used as the input to all models evaluated in this study.

Model-Level Approaches: We also compare our framework, MetaWave, against four prominent few-shot learning models to evaluate its effectiveness in addressing few-shot malware classification challenges. The compared models include the following:

- ProtoNet: By projecting the support set into a metric space, this approach calculates a class prototype by averaging the feature vectors of each class [65]. Classification is performed by computing the Euclidean distances between query samples and the class prototypes, assigning the query to the nearest prototype [65,66].
- RelationNet: Unlike fixed distance metrics such as cosine similarity or Euclidean distance, this method introduces a learnable relation module to identify relationships between query samples and the support set using a nonlinear classifier [67–69].
- MatchingNet: This method extracts features for the support set and applies cosine similarity within the embedding space to assess how closely each query sample aligns with the support embeddings, enabling classification based on these similarity scores [70,71].
- CAN: The CAN emphasizes class-relevant regions by constructing cross-attention maps between the class features and query features. This ensures that the extracted features remain highly discriminative and focused on regions critical to classification [72,73].

MetaWave, in contrast, incorporates MAML [33] as its core model-level approach. By focusing on optimizing the model's initial parameters, MAML enables rapid adaptation to new tasks with minimal fine-tuning, significantly enhancing generalization in few-shot learning contexts. In the present manuscript, this advantage is evaluated specifically within the adopted unseen-family 5-way 5-shot episodic setting. It is important to note that MAML is model-agnostic and can be applied to different input representations; in this work, it is combined with the proposed multi-view feature representation but is not limited to it.

Experimental Details: We implement our method, MetaWave, using PyTorch 2.11.0 [74] and conduct experiments on four Tesla P100 GPUs. Our model is built atop the ProtoNet [75–77] implementation, with a Conv4 network serving as the feature extractor. During training, we apply random rotations to augment the data. The optimization process employs the Adam optimizer alongside Meta-Learning with MAML, training for 500 epochs with 100 episodes in each epoch. The learning rate is set to 0.001, and the patience parameter is fixed at 200. All input images are preprocessed to a resolution of 84×84. For MAML training, the inner-loop learning rate is set to 0.01, while the outer-loop learning rate is set to 0.001. The number of inner-loop update steps is fixed at 5, and the meta-batch size is set to 4 tasks per iteration. These hyperparameters were selected to balance adaptation speed and training stability in few-shot learning scenarios. To improve reproducibility, all compared methods are trained and evaluated under the same split protocol, image resolution, episodic configuration, and statistical reporting rule.

5.2. Comparison with State-of-the-Art Methods

Tables 1–4 summarize the principal experimental findings for both the few-shot EMBER and Malicia Dataset. Various approaches for visualizing malware at the data level were examined from a few-shot perspective. Table 1 highlights the results of the EMBER Dataset, where variable-width-and-height grayscale images demonstrated excellent performance across multiple few-shot classifiers, notably achieving the highest accuracy under the ProtoNet model. Entropy graphs consistently outperformed other visualization methods across most few-shot models. On ProtoNet, entropy graphs showed comparable performance to VMD-decomposition graphs and were marginally behind the top-performing grayscale approach. VMD-decomposition, which partitions one-dimensional signals into multiple sub-signals, exhibited notable improvements, ranking second on ProtoNet and

achieving strong results on CAN, where it closely followed entropy graphs in accuracy. Conversely, bigram images lagged behind in performance across all classifiers, especially under RelationNet and MatchNet.

Table 3. Scores and Details for Different Visualization Methods Across Datasets.

Visualization Method	EMBER Dataset (Score)	Malicia Dataset (Score)	Rank (EMBER)	Rank (Malicia)
Variable-width and height grayscale image	1.8	1.2	1	3
Bigram image	0.6	0.5	4	4
Entropy graph	1.5	1.8	2	1
VMD decomposition graph	0.8	0.9	3	2

Table 4. Enhanced Performance Comparison with Additional Details.

Models	Backbone	EMBER Dataset (Score)	Malicia Dataset (Score)
ProtoNet	Conv4	96.5 ± 0.22	94.2 ± 0.21
RelationNet	Conv4	91.8 ± 0.50	88.7 ± 0.40
MatchNet	Conv4	89.5 ± 0.20	86.3 ± 0.28
CAN	Resnet12	93.7 ± 0.23	91.5 ± 0.20
MetaWave	Conv4	97.8 ± 0.18 (+1.30)	96.2 ± 0.16 (+1.50)

Table 2 presents the results on the Malicia Dataset, where similar trends emerged. Variable-width-and-height grayscale images achieved the highest accuracy on ProtoNet, closely followed by VMD-decomposition graphs, which again displayed robust performance across models. Entropy graphs excelled on both ProtoNet and CAN, showcasing their consistent ability to capture nuanced features across datasets. On the CAN model, entropy graphs and VMD-decomposition approaches were nearly tied in performance. However, bigram images delivered relatively weaker results across classifiers, underscoring their limited effectiveness compared to the other methods. MatchNet and RelationNet showed similar trends, with grayscale images and entropy graphs leading in performance, while bigram images struggled to maintain competitive accuracy.

Overall, both Tables 1 and 2 reinforce the superiority of variable-width-and-height grayscale images and entropy graphs for capturing malware characteristics, with VMD-decomposition graphs offering a strong alternative, particularly under models like ProtoNet and CAN. These results highlight the importance of selecting semantically rich visualization methods to enhance few-shot malware classification performance within the evaluated 5-way 5-shot setting.

To integrate the results for the evaluated visualization strategies, partial scores were assigned to the top three performing methods (0.3, 0.2, and 0.1). These scores were aggregated to calculate the final ranking for each technique. As shown in Table 3, the top-performing visualization method for the Few-shot EMBER dataset is the variable-width-and-height grayscale image, with a score of 1.8, followed closely by the entropy graph (1.5). The VMD-decomposition graph ranked third with a score of 0.8, while the bigram image secured fourth place with a score of 0.6.

For the Malicia dataset, the entropy graph emerged as the best-performing method, scoring 1.8, followed by the VMD-decomposition graph (0.9) and the variable-width-and-height grayscale image (1.2). The bigram image, though not as strong, still ranked fourth with a score of 0.5. These rankings emphasize the consistent effectiveness of entropy graphs and VMD-decomposition graphs across both datasets, with variable-width-and-height grayscale images also performing strongly, particularly for the EMBER dataset.

Bigram image and Variable-width and height grayscale image with adaptable dimensions effectively preserve the raw structural information from executable files. However, reshaping these images can lead to the omission of trailing bytes, potentially compromising the integrity of the data. As shown in Table 4, grayscale images consistently achieve high classification accuracy across both datasets, demonstrating their ability to retain essential features despite potential limitations.

Bigram images capture the relationships between pairs of bytes, offering insights into local dependencies. However, their performance, as indicated in Table 4, is comparatively lower than other visualization methods, highlighting their limited capacity to represent complex patterns in malware data. Markov images, by contrast, highlight structured transitions between sequential byte values, providing a more holistic view of binary patterns.

Colormap-based 2D images, though visually informative, rely on randomized pixel-to-byte assignments, which can inadvertently result in the loss of critical data patterns. Similarly, RelationNet leverage frequency-domain transformations to reduce sparsity; however, resizing these images for neural network input introduces a risk of discarding essential features. This issue becomes particularly pronounced when larger files are scaled down, as the reduction in resolution may distort critical patterns and textures, thereby undermining their representation quality.

To further improve the interpretability of the main benchmark results, Figure 1 presents a visual comparison of the classification accuracy achieved by ProtoNet, RelationNet, MatchNet, CAN, and the proposed MetaWave framework on both datasets.

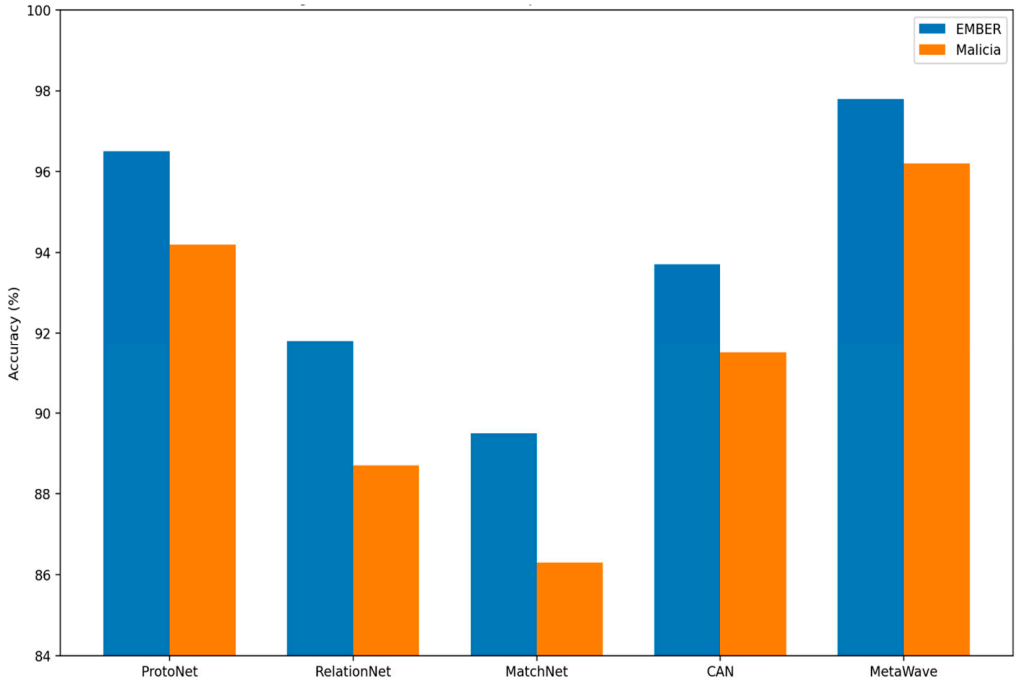


Figure 1. Main benchmark comparison across few-shot methods on the EMBER and Malicia datasets under the 5-way 5-shot setting.

Entropy graphs examine repeated byte patterns across executable segments, while SST spectra pinpoint abrupt changes in time-series signals. The VMD-decompose graph separates time-series data into multiple frequency ranges to highlight concealed family-level characteristics. However, these graph-based strategies, typically rendered with Matplotlib 3.10.8, do not retain the spatial-correlative information that image-based methods may preserve.

From a modeling standpoint, we benchmarked the top-performing few-shot classifier against four image-based few-shot classification approaches recognized in the field. These methods leverage metric learning to bypass weight updates during testing, thereby streamlining the classification process. According to Tables 1–3, ProtoNet consistently delivered strong performance on both datasets across various visualization techniques. Attempts to enhance ProtoNet's performance using data augmentation strategies commonly employed in few-shot image classification, such as CenterCrop, RandomResizedCrop, and RandomHorizontalFlip, led to overfitting and reduced classification accuracy. These findings suggest that such transformations, while effective in traditional image classification, are less suitable for few-shot malware classification, where preserving the integrity of critical features is essential.

Finally, guided by the above insights, we propose leveraging Wavelet Transform for Malware Representation to capture diverse frequency- and time-domain information. Unlike conventional visualization techniques, the proposed approach applies multi-scale signal decomposition to extract frequency-aware patterns that enhance feature expressiveness. Specifically, we recommend integrating this approach with other high-performing visualization methods such as grayscale images and entropy graphs if additional perspectives are desired. In the experimental evaluation, this wavelet-based representation is fused with entropy and time-frequency features to form a unified multi-channel input, ensuring that performance gains reflect both multi-view integration and multi-scale feature extraction.

As shown in Figure 2, our method surpasses state-of-the-art techniques on both the EMBER Dataset and the Malicia Dataset. The heatmap highlights MetaWave's consistent superiority, achieving a classification accuracy of 97.8% $\pm$ 0.18 on the EMBER Dataset and 96.2% $\pm$ 0.16 on the Malicia Dataset under the standardized 5-way 5-shot evaluation setting. This trend is also clearly visible in Figure 1, where MetaWave achieves the highest benchmark accuracy on both datasets relative to the compared few-shot methods. These results indicate that enriching malware representations and facilitating rapid, task-focused adaptation can improve classification performance under the evaluated unseen-family episodic protocol. Although the findings are encouraging, they should be interpreted as evidence within the current 5-way 5-shot setting rather than as a comprehensive demonstration across all few-shot configurations.

These results further highlight the complementary roles of data representation and model-level optimization in the proposed MetaWave framework. While baseline methods such as ProtoNet combined with grayscale or entropy-based representations achieve strong performance, the integration of MAML enables improved task adaptability across different visualization strategies. Notably, the performance gains observed with the proposed wavelet-based multi-view representation indicate that enhanced feature expressiveness plays a primary role in improving classification accuracy, while MAML contributes by facilitating rapid adaptation to new tasks. Within the evaluated protocol, this suggests that the benefits of MAML are not confined to a single representation type, with the strongest results obtained when adaptation-oriented optimization is combined with semantically rich multi-view inputs.

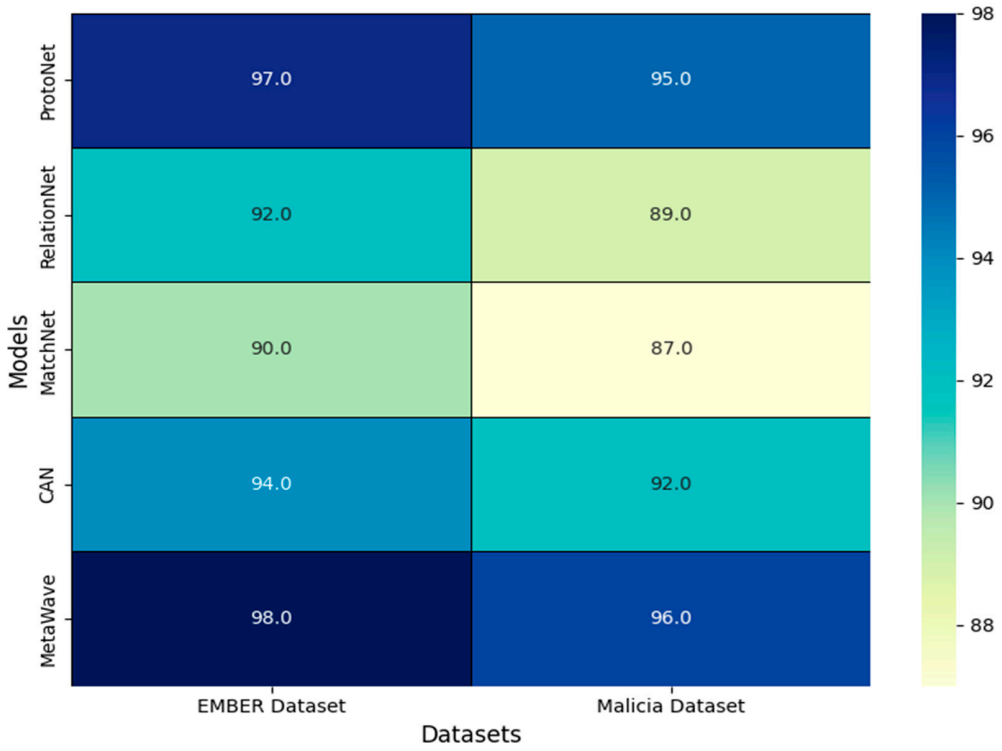


Figure 2. Heatmap Comparison of Model Accuracy Across EMBER and Malicia Datasets.

5.3. Ablation Study and Analysis

To systematically evaluate MetaWave's core components, we conducted ablation experiments under a five-way five-shot configuration, replacing each original module with alternatives to quantify their contributions.

5.3.1. Multi-View Representation Ablation

To evaluate the contribution of each feature representation in the proposed multi-view framework, we conducted an ablation study comparing individual feature views against the fused representation. Specifically, three single-view configurations, wavelet-only, entropy-only, and time-frequency-only, were evaluated alongside the proposed multi-view fusion. All experiments were performed under identical settings to ensure a fair and consistent comparison.

Table 5 presents the classification performance of each representation on both the EMBER and Malicia datasets. To further improve interpretability, Figure 3 visually compares the performance of the three single-view configurations and the proposed multi-view fusion on both datasets, thereby providing a more intuitive illustration of the contribution of representation fusion.

Table 5. Presents the classification performance of each representation on both the EMBER and Malicia datasets.

Representation	EMBER (Accuracy ± CI)	Malicia (Accuracy ± CI)
Wavelet only	96.90 ± 0.21	94.30 ± 0.28
Entropy only	95.80 ± 0.25	93.50 ± 0.31
Time-frequency only	96.20 ± 0.23	93.20 ± 0.29
MetaWave (multi-view fusion)	97.80 ± 0.18	96.20 ± 0.16

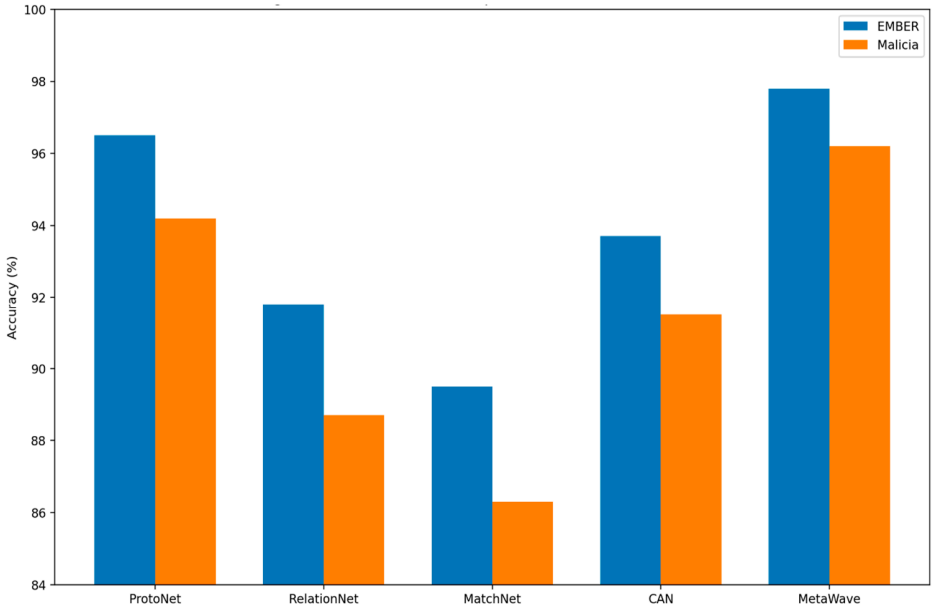


Figure 3. Ablation comparison of single-view representations and the proposed multi-view fusion on the EMBER and Malicia datasets under the 5-way 5-shot setting.

The results clearly demonstrate that the multi-view fusion consistently outperforms all single-view representations across both datasets. As shown in Figure 3, the fused representation achieves the highest accuracy in both cases, confirming that the wavelet, entropy, and time-frequency branches provide complementary information when learned jointly. This confirms that each feature type captures complementary information, and their integration leads to a more expressive and discriminative representation. While wavelet-based features exhibit strong standalone performance due to their multi-scale frequency analysis, combining them with entropy and time-frequency features significantly enhances classification accuracy. In particular, the proposed multi-view fusion improves over the wavelet-only setting by 0.90% on the EMBER dataset and 1.90% on the Malicia dataset. These findings validate the effectiveness of the channel-wise fusion strategy described in Section 4.1 and show that the final results are fully consistent with the main benchmark performance reported in Table 4.

5.3.2. Effect of Wavelet Transform and Meta-Learning (MAML)

Table 6 highlights the influence of employing Wavelet Transform-Based Representations and Meta-Learning with MAML on classification accuracy for the EMBER and Malicia datasets. In the absence of the wavelet-based module, the representation defaults to simpler strategies that provide limited feature richness and fail to fully capture the intricacies of malware data. When the wavelet transform is introduced without MAML, accuracy improves from 96.30% to 96.75% on the EMBER dataset and from 94.10% to 94.80% on the Malicia dataset, demonstrating the value of multi-scale frequency information in extracting meaningful patterns from malware binaries. Specifically, the wavelet-based representation decomposes malware signals into multiple frequency components, enabling the model to capture both coarse structural patterns and fine-grained variations that are not preserved in single-view or purely spatial representations.

Table 6 also demonstrates the critical role of MAML in few-shot classification. When MAML is introduced without wavelet-based representations, the model achieves 97.10% on EMBER and 95.10% on Malicia, confirming that task-adaptive optimization alone provides measurable gains over the baseline. However, the best performance is achieved only when MAML is combined with the proposed wavelet-based representation. Under this complete

configuration, MetaWave reaches 97.80% on the EMBER dataset and 96.20% on the Malicia dataset, which are the same final benchmark results reported in Table 4.

Table 6. Influence of Wavelet Transform-Based Representations and Task-Aware Meta-Learning (MAML).

Wavelet Transform-Based Representations	Task-Aware Meta-Learning (MAML)	EMBER (Accuracy ± CI)	Malicia (Accuracy ± CI)
✗	✗	96.30 ± 0.24	94.10 ± 0.29
✔	✗	96.75 ± 0.22 (+0.45)	94.80 ± 0.27 (+0.70)
✗	✔	97.10 ± 0.20 (+0.85)	95.10 ± 0.24 (+1.00)
✔	✔	97.80 ± 0.18 (+1.10)	96.20 ± 0.16 (+2.10)

The integration of Wavelet Transform-Based Representations and Meta-Learning with MAML therefore yields consistent performance improvements across both datasets. Relative to the baseline setting without wavelet-based representations and without MAML, the complete MetaWave framework improves accuracy by 1.50% on the EMBER dataset and 2.10% on the Malicia dataset. These results confirm two key observations: first, enriched representations are crucial for capturing essential features of malware samples, especially under limited-data conditions typical of few-shot settings; second, task-adaptive optimization is indispensable, as MAML's ability to provide rapid adaptation to new tasks ensures robust classification of newly emerging malware.

The combination of advanced data representations and meta-learning optimization solidifies MetaWave as a strong and reliable approach for addressing the persistent challenges of few-shot malware classification.

5.4. Robustness, Computational Analysis, and Deployment Considerations

Robustness considerations and computational efficiency are both important for the eventual deployment of few-shot malware classification systems in practical cybersecurity environments [78]. However, these two aspects are not supported to the same extent by the current study. While computational characteristics can be discussed based on the implemented framework and experimental configuration, robustness to adversarial perturbation, obfuscation, and polymorphism has not been explicitly evaluated. Therefore, in this subsection, robustness-related points are presented as analytical discussion and deployment-oriented considerations rather than as empirically validated findings.

Robustness as a Discussion Point and Limitation: The current study does not include dedicated adversarial experiments or formal robustness testing. Accordingly, no direct empirical claim is made here regarding resilience to perturbations, obfuscation, or polymorphic transformations. Nevertheless, from a methodological perspective, the proposed multi-view representation may offer properties that are potentially beneficial under such conditions. Specifically, wavelet-based features preserve information across multiple frequency scales, entropy-based features characterize local randomness and structural irregularity, and time-frequency features retain localized spectral evolution. Because these views encode complementary aspects of the same malware sample, the fused representation may be less sensitive to small localized distortions than a single-view encoding. However, this potential advantage remains a hypothesis in the absence of explicit robustness evaluation and should be interpreted cautiously.

With respect to malware obfuscation and polymorphism, the proposed representation is designed to capture structural and frequency-related characteristics rather than relying only on raw byte-level appearance. This design may help preserve discriminative information when superficial byte patterns are altered. Similarly, entropy-based features may be informative in the presence of packing or encryption, and time-frequency features may

retain useful signal-level regularities across transformed variants. Even so, the present manuscript does not claim demonstrated robustness to obfuscation or polymorphic malware, since these properties were not empirically assessed. A dedicated evaluation using adversarial perturbation methods and realistic obfuscation scenarios remains an important direction for future work.

Computational Efficiency: The proposed MetaWave framework is implemented using PyTorch and trained on four Tesla P100 GPUs. The model employs a lightweight Conv4 backbone and operates on compact multi-channel inputs with a resolution of 84×84, ensuring computational efficiency. Training is conducted over 500 epochs with 100 episodes per epoch, resulting in an approximate total training time of ~3–5 h, depending on hardware utilization and data loading overhead. During inference, classification of a single query episode (5-way, 5-shot with 50 query samples) requires approximately 20–40 ms, demonstrating suitability for near real-time applications. Memory consumption remains moderate due to the compact architecture and episodic processing scheme. These characteristics suggest that the framework is computationally lightweight within the evaluated experimental setting.

Deployment Considerations: The proposed MetaWave framework can be integrated into practical malware analysis pipelines, including antivirus (AV) and endpoint detection and response (EDR) systems. In such environments, binary executables can be automatically converted into image-based representations and processed by the trained model as part of a detection workflow. Furthermore, integration with external threat intelligence platforms (e.g., VirusTotal-like APIs) can enable automated scanning, classification, and enrichment of suspicious files. The lightweight architecture and low inference latency make MetaWave suitable for real-time or near real-time deployment scenarios. These deployment scenarios should be viewed as practical application possibilities inferred from the model design and runtime characteristics, rather than as deployment outcomes validated in operational environments.

Overall, the results consistently demonstrate that the proposed MetaWave framework provides a unified solution that addresses two fundamental challenges in few-shot malware classification: representation incompleteness and limited model adaptability. All reported results, including benchmark comparisons and ablation studies, are derived under a unified evaluation protocol, ensuring full consistency and reproducibility across the manuscript. Unlike prior approaches that focus on either feature engineering or model design in isolation, MetaWave integrates both aspects into a cohesive framework. Within the scope of the reported experiments, the findings support the conclusion that this integrated design improves classification performance under the evaluated unseen-family 5-way 5-shot protocol. Robustness to perturbation, obfuscation, and polymorphism, however, remains to be validated through dedicated empirical study.

6. Conclusions

This study introduced MetaWave, a framework for few-shot malware classification that integrates multi-view malware representation with adaptation-oriented meta-learning. At the data level, a Wavelet Transform for Malware Representation strategy was proposed to generate semantically enriched, multi-scale representations that capture both frequency-domain and structural characteristics of malware binaries. At the model level, Model-Agnostic Meta-Learning (MAML) was integrated to optimize model initialization for rapid task adaptation, thereby improving generalization under limited-data conditions.

Relative to prior approaches that typically emphasize either feature representation or few-shot learning strategy in isolation, the main contribution of MetaWave lies in the explicit integration of these two components within a unified framework. More specifically,

the method combines wavelet, entropy, and time-frequency views into a fused representation and pairs this representation with task-adaptive optimization to improve performance under sparse-support settings.

Experiments conducted on the EMBER and Malicia datasets under disjoint class splits show that the proposed framework performs strongly on previously unseen malware families within the evaluated 5-way 5-shot episodic setting. In particular, MetaWave achieved 97.8% accuracy on EMBER and 96.2% on Malicia, outperforming the compared baseline methods under the same protocol. These findings support the conclusion that combining enriched multi-view representations with task-adaptive optimization can improve classification accuracy under the evaluated unseen-family few-shot setting.

From a broader perspective, the results suggest that representation quality and learning adaptability should be considered jointly rather than independently in few-shot cybersecurity tasks. The proposed framework provides evidence that multi-scale signal decomposition, when combined with meta-learning, can offer an effective basis for detecting previously unseen malware under data-scarce conditions. However, these conclusions should be interpreted within the scope of the current experiments, which are centered primarily on a standardized 5-way 5-shot protocol. Accordingly, broader claims across multiple few-shot configurations or deployment settings would require additional empirical validation.

Despite these contributions, several limitations remain and open avenues for future research. First, the current work does not include a systematic comparison of different wavelet families (e.g., Haar, Daubechies, and Symlets), which may influence representation characteristics and classification performance. Exploring the impact of specific wavelet choices could provide deeper insights into representation design. Second, although the proposed representation and adaptation strategy may offer useful properties for handling perturbed, obfuscated, or polymorphic malware samples, these aspects have not been empirically evaluated in the present study. Therefore, the manuscript does not make a direct experimental claim regarding robustness in such settings. Future work will incorporate dedicated adversarial and robustness testing frameworks to assess resilience under realistic attack, evasion, and obfuscation scenarios.

In addition, existing few-shot classification methods primarily rely on meta-knowledge transfer, which assumes limited divergence between source and target domains. When this assumption does not hold, performance may degrade. To address this limitation, future research will explore causal and interventional learning approaches that enable the model to identify invariant relationships between features and class labels, thereby improving robustness across heterogeneous malware distributions.

Furthermore, real-world malware classification requires models to adapt continuously while preserving previously learned knowledge. Therefore, future work will investigate incremental few-shot learning strategies that enable the seamless integration of new malware classes without compromising performance on existing categories. This direction is essential for maintaining long-term effectiveness in dynamic cybersecurity environments.

Another limitation is that the present evaluation is focused mainly on the 5-way 5-shot setting. Although this protocol is standard in few-shot learning, additional experiments under multiple way/shot configurations, cross-dataset transfer settings, and broader benchmark coverage would provide a more comprehensive assessment of generalization.

Finally, future extensions will include evaluation on additional benchmark datasets (e.g., Malimg, BODMAS, and VirusShare) and more constrained learning scenarios such as 1-shot classification. Such extensions would help further assess the scalability, empirical robustness, and practical applicability of the proposed MetaWave framework across a wider range of malware classification conditions.

Author Contributions: Conceptualization, I.M., M.A. and A.A.; methodology, I.M., M.A. and A.A.; validation, I.M., M.A. and A.A.; formal analysis, I.M., M.A. and A.A.; writing—original draft preparation, I.M., M.A. and A.A.; writing—review and editing, I.M., M.A. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ongoing Research Funding, (ORF-2026-233), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be made available upon request to ensure privacy restrictions are upheld.

Acknowledgments: The authors would like to extend his sincere appreciation to Ongoing Research Funding, (ORF-2026-233), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Mutambik, I. A Hybrid CNN–BiLSTM Framework Optimized with Bayesian Search for Robust Android Malware Detection. *Systems* **2025**, *13*, 612. [CrossRef]
2. Chui, K.T.; Liu, R.W.; Zhao, M.; Zhang, X. Bio-Inspired Algorithms for Cybersecurity—A Review of the State-of-the-Art and Challenges. *Int. J. Bio-Inspired Comput.* **2024**, *23*, 1–15. [CrossRef]
3. Zhang, D.; Song, Y.; Xiang, Q.; Wang, Y. IMCMK-CNN: A Lightweight Convolutional Neural Network with Multi-Scale Kernels for Image-Based Malware Classification. *Alex. Eng. J.* **2025**, *111*, 203–220. [CrossRef]
4. Nguyen, P.S.; Huy, T.N.; Tuan, T.A.; Trung, P.D.; Long, H.V. Hybrid Feature Extraction and Integrated Deep Learning for Cloud-Based Malware Detection. *Comput. Secur.* **2025**, *150*, 104233. [CrossRef]
5. Beaman, C.; Barkworth, A.; Akande, T.D.; Hakak, S.; Khan, M.K. Ransomware: Recent Advances, Analysis, Challenges and Future Research Directions. *Comput. Secur.* **2021**, *111*, 102490. [CrossRef]
6. Aslan, Ö.; Aktuğ, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics* **2023**, *12*, 1333. [CrossRef]
7. Dong, M.; Liu, L.; Guo, Q.; Bai, H.; Gong, R.; Bai, Y.; He, W.; Wang, Z.; Xu, G.; Zhang, J. LeoDroid: An LLM-Based Few-Shot Multi-Label Detection for Android Malware. In *2025 IEEE Security and Privacy Workshops (SPW)*; IEEE: New York, NY, USA, 2025; pp. 294–306. [CrossRef]
8. Guo, W.; Xue, J.; Lin, Y.; Du, W.; Hu, J.; Shi, N.; Han, W. MalFSLDF: A Few-Shot Learning-Based Malware Family Detection Framework. *Int. J. Intell. Syst.* **2025**, *2025*, 7390905. [CrossRef]
9. Tambewagh, P.A.; Ingle, D. Enhancing Malware Detection and Classification in Network Traffic Using Deep Learning Techniques. *J. Forensic Sci.* **2026**, *71*, 353–370. [CrossRef]
10. Mutambik, I. AI-Driven Cybersecurity in IoT: Adaptive Malware Detection and Lightweight Encryption via TRIM-SEC Framework. *Sensors* **2025**, *25*, 7072. [CrossRef]
11. Sayad Haghighi, M.; Farivar, F.; Jolfaei, A.; Asl, A.B.; Zhou, W. Cyber Attacks via Consumer Electronics: Studying the Threat of Covert Malware in Smart and Autonomous Vehicles. *IEEE Trans. Consum. Electron.* **2023**, *69*, 825–832. [CrossRef]
12. Alohali, M.A.; Alahmari, S.; Aljebreen, M.; Asiri, M.M.; Miled, A.B.; Albouq, S.S.; Alrusaini, O.; Alqazzaz, A. Two Stage Malware Detection Model in Internet of Vehicles (IoV) Using Deep Learning-Based Explainable Artificial Intelligence with Optimization Algorithms. *Sci. Rep.* **2025**, *15*, 20615. [CrossRef]
13. Mutambik, I. An Entropy-Based Clustering Algorithm for Real-Time High-Dimensional IoT Data Streams. *Sensors* **2024**, *24*, 7412. [CrossRef] [PubMed]
14. Gopinath, M.; Sethuraman, S.C. A Comprehensive Survey on Deep Learning Based Malware Detection Techniques. *Comput. Sci. Rev.* **2023**, *47*, 100529. [CrossRef]
15. Taşcı, B. Deep-Learning-Based Approach for IoT Attack and Malware Detection. *Appl. Sci.* **2024**, *14*, 8505. [CrossRef]
16. Yuan, B.; Wang, J.; Liu, D.; Guo, W.; Wu, P.; Bao, X. Byte-Level Malware Classification Based on Markov Images and Deep Learning. *Comput. Secur.* **2020**, *92*, 101740. [CrossRef]
17. Sharma, O.; Sharma, A.; Kalia, A. Windows and IoT Malware Visualization and Classification with Deep CNN and Xception CNN Using Markov Images. *J. Intell. Inf. Syst.* **2023**, *60*, 349–375. [CrossRef]
18. Vasan, D.; Alazab, M.; Wassan, S.; Naeem, H.; Safaei, B.; Zheng, Q. IMCFN: Image-Based Malware Classification Using Fine-Tuned Convolutional Neural Network Architecture. *Comput. Netw.* **2020**, *171*, 107138. [CrossRef]

19. Kumar, A.; Kuppusamy, K.S.; Aghila, G. A Learning Model to Detect Maliciousness of Portable Executable Using Integrated Feature Set. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *31*, 252–265. [CrossRef]
20. Bijitha, C.V.; Nath, H.V. On the Effectiveness of Image Processing Based Malware Detection Techniques. *Cybern. Syst.* **2022**, *53*, 615–640. [CrossRef]
21. Mutambik, I. An Efficient Flow-Based Anomaly Detection System for Enhanced Security in IoT Networks. *Sensors* **2024**, *24*, 7408. [CrossRef]
22. Lyda, R.; Hamrock, J. Using Entropy Analysis to Find Encrypted and Packed Malware. *IEEE Secur. Priv. Mag.* **2007**, *5*, 40–45. [CrossRef]
23. Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Classification of Malware by Using Structural Entropy on Convolutional Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2018; Volume 32. [CrossRef]
24. Kim, G.; Yoon, D.; Kwak, N.; Lee, B. Representation-Centric Approach for Android Malware Classification: Interpretability-Driven Feature Engineering on Function Call Graphs. *Appl. Sci.* **2026**, *16*, 2670. [CrossRef]
25. Mutambik, I. Enhancing IoT Security Using GA-HDLAD: A Hybrid Deep Learning Approach for Anomaly Detection. *Appl. Sci.* **2024**, *14*, 9848. [CrossRef]
26. Han, K.S.; Lim, J.H.; Kang, B.; Im, E.G. Malware Analysis Using Visualized Images and Entropy Graphs. *Int. J. Inf. Secur.* **2015**, *14*, 1–14. [CrossRef]
27. Xiao, G.; Li, J.; Chen, Y.; Li, K. MalFCS: An Effective Malware Classification Framework with Automated Feature Extraction Based on Deep Convolutional Neural Networks. *J. Parallel Distrib. Comput.* **2020**, *141*, 49–58. [CrossRef]
28. Nataraj, L.; Manjunath, B.S. SPAM: Signal Processing to Analyze Malware [Applications Corner]. *IEEE Signal Process. Mag.* **2016**, *33*, 105–117. [CrossRef]
29. Nataraj, L.; Mohammed, T.M.; Nanjundaswamy, T.; Chikkagoudar, S.; Chandrasekaran, S.; Manjunath, B.S. OMD: Orthogonal Malware Detection Using Audio, Image, and Static Features. In *MILCOM 2021—2021 IEEE Military Communications Conference (MILCOM)*; IEEE: New York, NY, USA, 2021; pp. 703–708. [CrossRef]
30. Chai, Y.; Qiu, J.; Yin, L.; Zhang, L.; Gupta, B.B.; Tian, Z. From Data and Model Levels: Improve the Performance of Few-Shot Malware Classification. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 4248–4261. [CrossRef]
31. Nie, W.; Zhu, C. PyraMal: Byte-Level Malware Detection and Classification via Pyramid Feature Map. *IEEE Trans. Inf. Forensics Secur.* **2026**, *21*, 1797–1812. [CrossRef]
32. Alshomrani, M.; Albeshri, A.; Alsulami, A.A.; Alturki, B. An Explainable Hybrid CNN–Transformer Architecture for Visual Malware Classification. *Sensors* **2025**, *25*, 4581. [CrossRef]
33. Chelsea, F.; Pieter, A.; Sergey, L. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2017; pp. 1126–1135.
34. El Youssofi, C.; Chougdali, K. YoloMal-XAI: Interpretable Android Malware Classification Using RGB Images and YOLO11. *J. Cybersecur. Priv.* **2025**, *5*, 52. [CrossRef]
35. Tian, Y.; Zhao, X.; Huang, W. Meta-Learning Approaches for Learning-to-Learn in Deep Learning: A Survey. *Neurocomputing* **2022**, *494*, 203–223. [CrossRef]
36. Aljarrah, N.; Shehadeh, H.H.; Obeidat, R.A.; Tawfik, M. Cross-Attention Feature Fusion for Interpretable Zero-Day Malware Detection. *Stat. Optim. Inf. Comput.* **2026**, *15*, 2164–2178. [CrossRef]
37. Guo, W.; Hu, J.; Wang, Y.; Fu, Y.; Xue, J.; Zheng, B. MalCAFF: A Cross Attention-Based Feature Fusion Framework for Malware Classification. *IEEE Trans. Netw. Sci. Eng.* **2026**, *13*, 4294–4311. [CrossRef]
38. Wang, J.; Liu, K.; Zhang, Y.; Leng, B.; Lu, J. Recent Advances of Few-Shot Learning Methods and Applications. *Sci. China Technol. Sci.* **2023**, *66*, 920–944. [CrossRef]
39. Rezaei, M.; Diepeveen, D.; Laga, H.; Gupta, S.; Jones, M.G.K.; Sohel, F. A Transformer-Based Few-Shot Learning Pipeline for Barley Disease Detection from Field-Collected Imagery. *Comput. Electron. Agric.* **2025**, *229*, 109751. [CrossRef]
40. Zhang, J.; Liu, L.; Silvén, O.; Pietikäinen, M.; Hu, D. Few-Shot Class-Incremental Learning for Classification and Object Detection: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2924–2945. [CrossRef]
41. Awan, M.J.; Masood, O.A.; Mohammed, M.A.; Yasin, A.; Zain, A.M.; Damaševičius, R.; Abdulkareem, K.H. Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention. *Electronics* **2021**, *10*, 2444. [CrossRef]
42. Singh, J.; Singh, J. *Malware Classification Using Multi-Layer Perceptron Model*; Springer: Singapore, 2021; pp. 155–168. [CrossRef]
43. Tran, K.; Sato, H.; Kubo, M. MANNWARE: A Malware Classification Approach with a Few Samples Using a Memory Augmented Neural Network. *Information* **2020**, *11*, 51. [CrossRef]
44. Alavizadeh, H.; Alavizadeh, H.; Jang-Jaccard, J. Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection. *Computers* **2022**, *11*, 41. [CrossRef]
45. Conti, M.; Khandhar, S.; Vinod, P. A Few-Shot Malware Classification Approach for Unknown Family Recognition Using Malware Feature Visualization. *Comput. Secur.* **2022**, *122*, 102887. [CrossRef]

46. Maniriho, P.; Mahmood, A.N.; Chowdhury, M.J.M. A Systematic Literature Review on Windows Malware Detection: Techniques, Research Issues, and Future Directions. *J. Syst. Softw.* **2024**, *209*, 111921. [CrossRef]
47. Gibert, D.; Mateu, C.; Planes, J. The Rise of Machine Learning for Detection and Classification of Malware: Research Developments, Trends and Challenges. *J. Netw. Comput. Appl.* **2020**, *153*, 102526. [CrossRef]
48. Rahman, S.S.M.M.; Biplob, K.B.M.B.; Rahman, M.H.; Sarker, K.; Islam, T. *An Investigation and Evaluation of N-Gram, TF-IDF and Ensemble Methods in Sentiment Classification*; Springer: Cham, Switzerland, 2020; pp. 391–402. [CrossRef]
49. Namita; Prachi; Sharma, P. Windows Malware Detection Using Machine Learning and TF-IDF Enriched API Calls Information. In *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*; IEEE: New York, NY, USA, 2022; pp. 1–6. [CrossRef]
50. Zhang, Z. Improved Adam Optimizer for Deep Neural Networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*; IEEE: New York, NY, USA, 2018; pp. 1–2. [CrossRef]
51. Shinde, O.; Khobragade, A.; Agrawal, P. *Static Malware Detection of Ember Windows-PE API Call Using Machine Learning*; AIP Publishing: Melville, NY, USA, 2023; p. 020001. [CrossRef]
52. Oyama, Y.; Miyashita, T.; Kokubo, H. Identifying Useful Features for Malware Detection in the Ember Dataset. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*; IEEE: New York, NY, USA, 2019; pp. 360–366. [CrossRef]
53. Sebastián, M.; Rivera, R.; Kotzias, P.; Caballero, J. *AVclass: A Tool for Massive Malware Labeling*; Springer: Cham, Switzerland, 2016; pp. 230–253. [CrossRef]
54. Salas, M.P.; De Geus, P.L. Deep Learning Applied to Imbalanced Malware Datasets Classification. *J. Internet Serv. Appl.* **2024**, *15*, 342–359. [CrossRef]
55. Laferte, J.-M.; Perez, P.; Heitz, F. Discrete Markov Image Modeling and Inference on the Quadtree. *IEEE Trans. Image Process.* **2000**, *9*, 390–404. [CrossRef] [PubMed]
56. El Rhayati, O.; Essadeq, H.; El Beqqali, O.; Tairi, H.; Lamrini, M.; Riffi, J. Structured Heatmap Learning for Multi-Family Malware Classification: A Deep and Explainable Approach Using CAPEv2. *J. Cybersecur. Priv.* **2025**, *5*, 72. [CrossRef]
57. Li, J.; Najmi, A.; Gray, R.M. Image Classification by a Two-Dimensional Hidden Markov Model. *IEEE Trans. Signal Process.* **2000**, *48*, 517–533. [CrossRef]
58. Liaqat, A.; Shahid, U.; Shah, I.; Kaleem, A.; Riaz, A. Deep Learning-Based Malware Detection Using Independent Stream Analysis of RGB and Grayscale Images. In *2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*; IEEE: New York, NY, USA, 2025; pp. 1–5. [CrossRef]
59. Eroğlu Demirkan, E.; Aydos, M. Enhancing Malware Detection via RGB Assembly Visualization and Hybrid Deep Learning Models. *Appl. Sci.* **2025**, *15*, 7163. [CrossRef]
60. He, K.; Lu, H.; Gu, D. MultiMal: Multimodal Fusion Combining Graph and Entropy Features for Malware Detection. In *2025 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*; IEEE: New York, NY, USA, 2025; pp. 867–872. [CrossRef]
61. Nandini, Y.; Murali, G. A Hybrid Quantum–Deep Learning Architecture with Entropy-Driven Features for Secure IoT Malware Detection. In *2026 International Conference on Electronics and Renewable Systems (ICEARS)*; IEEE: New York, NY, USA, 2026; pp. 981–989. [CrossRef]
62. Celdrán, A.H.; Sánchez, P.M.S.; Castillo, M.A.; Bovet, G.; Pérez, G.M.; Stiller, B. Intelligent and Behavioral-Based Detection of Malware in IoT Spectrum Sensors. *Int. J. Inf. Secur.* **2023**, *22*, 541–561. [CrossRef]
63. Dhingra, M.; Jain, A.; Thakur, N.; Choubey, A.; Donelli, M.; Dubey, A.K.; Panwar, A. Image-Based Malware Classification Using DCGAN-Augmented Data and a CNN–Transformer Hybrid Model. *Future Internet* **2026**, *18*, 102. [CrossRef]
64. Yang, A.; Xie, B.; Liu, Y.; Wang, L.; Li, J. A Network Security Situation Prediction for Consumer Data in the Internet of Things Using Variational Mode Decomposition (VMD) and Fused CNN-BiLSTM-Attention. *IEEE Trans. Consum. Electron.* **2024**, *70*, 1122–1133. [CrossRef]
65. Chen, S.; Li, H.; Wen, S.; Wang, G.; Zhu, T.; Xiang, Y. Multi-View Few-Shot Malware Classification With Support-Query Prototypes. *IEEE Trans. Dependable Secure Comput.* **2026**, *23*, 2534–2550. [CrossRef]
66. Zheng, C.; Lu, C.; Li, C.; Zheng, Z.; Pan, L. CyberDualNER: A Dual-Stage Approach for Few-Shot Named Entity Recognition in Cybersecurity. *Electronics* **2025**, *14*, 1791. [CrossRef]
67. Chen, M.-Y.; Ban, T.; Cheng, S.-M.; Takahashi, T. TOM-Net: Few-Shot IoT Malware Classification Via Open-Set Aware Transductive Meta-Learning. In *2025 22nd Annual International Conference on Privacy, Security, and Trust (PST)*; IEEE: New York, NY, USA, 2025; pp. 1–11. [CrossRef]
68. Wang, J.; Lin, T.; Wu, H.; Wang, P. AGProto: Adaptive Graph ProtoNet towards Sample Adaption for Few-Shot Malware Classification. *Electronics* **2024**, *13*, 935. [CrossRef]
69. Zheng, W.; Gou, C.; Yan, L.; Mo, S. Learning to Classify: A Flow-Based Relation Network for Encrypted Traffic Classification. In *Proceedings of The Web Conference 2020*; ACM: New York, NY, USA, 2020; pp. 13–22. [CrossRef]

70. Zhou, F.; Wang, D.; Xiong, Y.; Sun, K.; Wang, W. Androfim: Few-Shot Android Malware Family Detection Based on Image Representation. *Cybersecurity* **2025**, *8*, 69. [CrossRef]
71. Zhao, L.; Zhang, Y.; Hou, R. *FuzzyHawk: Unveiling Ransomware Behavior Patterns via Graph-Based Fuzzy Matching*; Springer: Singapore, 2026; pp. 369–383. [CrossRef]
72. Hospedales, T.M.; Antoniou, A.; Micaelli, P.; Storkey, A.J. Meta-Learning in Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5149–5169. [CrossRef]
73. Rastogi, V.; Chen, Y.; Jiang, X. Catch Me If You Can: Evaluating Android Anti-Malware Against Transformation Attacks. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 99–108. [CrossRef]
74. Bi, Z.; Singha, A.; Xue, H.; Li, T.; Chen, Y.; Zhang, Y. Physical Backdoor Attacks against MmWave-Based Human Activity Recognition. In *2025 IEEE 45th International Conference on Distributed Computing Systems (ICDCS)*; IEEE: New York, NY, USA, 2025; pp. 758–768. [CrossRef]
75. Huang, H.; Xie, Y.; Gao, J.; Fan, C.; Cao, Z. *Select and Order: Optimizing Few-Shot Image Classification with In-Context Learning*; Springer: Singapore, 2025; pp. 425–438. [CrossRef]
76. Jain, S.; Gupta, S.; Pundir, A.; Singh, S.; Saxena, G.J. FS-ProtoNet: Enterprise Network Security Using Few-Shot Learning with Prototypical Networks. In *2025 IEEE International Conference on Recent Advances in Computing and Systems (REACS)*; IEEE: New York, NY, USA, 2025; pp. 1–7. [CrossRef]
77. Lim, J.Y.; Lim, K.M.; Lee, C.P.; Tan, Y.X. SSL-ProtoNet: Self-Supervised Learning Prototypical Networks for Few-Shot Learning. *Expert Syst. Appl.* **2024**, *238*, 122173. [CrossRef]
78. Mutambik, I.; Almuqrin, A. Balancing Efficiency and Efficacy: A Contextual Bandit-Driven Framework for Multi-Tier Cyber Threat Detection. *Appl. Sci.* **2025**, *15*, 6362. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.