Hybrid CNN-MLP for Malware Detection on Heterogeneous Features

Sabre Didi[1] and Geoff Nitschke[2]

Cape Peninsula University of Technology, Cape Town , South Africa
didis@cput.ac.za
University of Cape Town, Cape Town, South Africa
gnitschke@cs.uct.ac.za

Abstract. Malware attacks remain a critical cyber-security concern, necessitating robust detection solutions for both individuals and organizations. Deep learning methods, including Convolutional Neural Networks (CNNs) and Multi-Layer Perceptrons (MLPs), have become integral to malware detection and classification. However, real-world malware datasets, such as Elastic Malware Benchmark for Empowering Researchers (EMBER), present heterogeneous features that differ in type, scale, and representation. This heterogeneity poses a significant challenge for traditional deep-learning methods, which often assume homogeneity in the input features, resulting in suboptimal learning and detection performance. Moreover, the rapid evolution of malware strains introduces further complexity, as conventional models struggle to adapt to novel patterns.
This study proposes a hybrid Multi-Branch MLP and CNN model that handles heterogeneous malware feature groups through dedicated branches to better capture local and global patterns. The model adapts to evolving, heterogeneous malware data and outperforms conventional models, as demonstrated using the EMBER dataset.

Keywords: Convolutional Neural Network · Deep Learning · Multi-Layer Perceptron · Multi-branch MLP · Malware · Portable Executable Files.

1 Introduction

Recently, the surge in malware threats has prompted extensive research into effective detection methods, with a particular focus on dynamic and static analyses [1, 2]. Static analysis examines features of binary programs without execution [3], offering a resource-efficient and secure approach, but often fails to accurately capture the behavior of runtime malware. In contrast, dynamic analysis monitors program execution to extract behavioral features [4], providing richer information but requiring customized runtime environments, such as virtual machines, which can be computationally expensive for large-scale datasets.

Windows remains a predominant platform for malware attacks, and most threats target Windows PE files [5]. Consequently, this study focuses on adaptive classifiers for Windows PE malware, addressing both detection efficiency and robustness. Traditional signature-based detection methods [6] are limited to known malware, highlighting the need for approaches capable of generalizing to unseen or zero-day malware [7].

The multi-layer perceptron (MLP) is a widely adopted supervised neural network architecture, trained using backpropagation, and has been shown to be particularly effective in malware detection due to its ability to capture complex, nonlinear relationships in high-dimensional data [8, 9].

To further improve feature representation, researchers have proposed multi-branch MLP architectures, where separate branches process distinct groups of features before their outputs are combined. This approach has been successfully applied in intrusion and malware detection tasks, with multi-branched perceptron networks achieving near-optimal detection rates and attention-enhanced MLP variants surpassing 99% accuracy in benchmark evaluations [10].

Beyond malware detection, the multi-branch MLP has been explored as a general solution for training on complex and high-dimensional datasets. In many real-world applications, the abundance of nonlinear features complicates the learning of effective decision boundaries. Multi-branch designs address this by decomposing the input into multiple parallel shallow MLPs, each responsible for a feature subset, while a selector or aggregator branch integrates these outputs to improve classification performance [11].

In recent years, deep learning (DL) has emerged as a powerful tool for malware detection [12], leveraging the capacity of neural networks to model complex, nonlinear relationships across diverse feature sets. However, real-world malware datasets, such as EMBER [13], present heterogeneous features, including byte histograms, entropy measures, string metadata, headers, sections, imports and exports, and data directories that differ in type, scale, and representation. Traditional CNNs and MLPs often struggle to learn effectively from such heterogeneous data, limiting detection performance and adaptability in fast-evolving malware landscapes [14–16].

To overcome these challenges, this study proposes a Hybrid Multi-Branch MLP and CNN framework, where each feature group is processed by a dedicated branch. The CNNs extract local patterns from histogram and entropy features, while MLP branches learn representations from structured and categorical metadata. These branches are then merged to capture complementary relationships across the heterogeneous dataset, providing a unified representation for malware classification.

The key contribution of this research is the demonstration of the effectiveness of a Hybrid CNN+MLP architecture in learning from heterogeneous malware features, enabling accurate and robust detection of Windows PE malware across diverse and complex datasets.

2 Research Aim and Questions

2.1 Research Aim

The aim of this research is to investigate the effectiveness of a hybrid neural network architecture, combining CNNs and MLPs, for improving malware detection accuracy on structured and distributional features in the EMBER 2018 dataset. Specifically, the study explores whether leveraging modality-specific processing for histogram and entropy features via CNNs, alongside tabular metadata features via MLPs, can outperform conventional single-architecture approaches.

2.2 Research Questions

This study seeks to answer the following research questions:

1. Evaluate the hybrid CNN+MLP model task-performance in terms of accuracy, precision, recall, F1-score, and ROC-AUC compared to standalone MLP and CNN models on the EMBER 2018 dataset.
2. Evaluate which feature groups (histogram, entropy, metadata, imports and exports, sections) contribute most significantly to the performance of the hybrid model.
3. Test if the hybrid CNN+MLP architecture demonstrates superior generalization across unseen data, especially in the presence of class imbalance, compared to traditional models.
4. Evaluate the capability of the hybrid architecture to reduce specific error types (false positives and false negatives) relative to standalone MLP or CNN architectures, particularly for malware samples with subtle statistical or structural anomalies.

3 Methods and Experiments

This study investigates the effectiveness of three neural network architectures for malware detection using the EMBER 2018 dataset [13]. The dataset provides a heterogeneous feature space that includes both high-dimensional structured vectors (including imports, exports, headers and strings) and grid-based feature representations (such as histograms and entropy). Each feature group has distinct characteristics, motivating the use of models tailored to its specific type.

3.1 Dataset

The 2018 EMBER (Elastic Malware Benchmark for Empowering Researchers) dataset [13] is a publicly available benchmark resource compiled to advance machine learning-based malware detection. It encompasses a comprehensive set of features extracted from Windows Portable Executable (PE) files, covering

both benign and malicious samples. This ensures that the evaluation of classification models is both reproducible and standardized. In total, the dataset contains feature representations from approximately 1.1 million PE files, comprising 900,000 training instances (300,000 malicious, 300,000 benign, and 300,000 unlabeled) and 200,000 test instances, evenly divided between 100,000 malicious and 100,000 benign samples.

EMBER was released by Endgame and researchers at MITRE to provide a standardized, large-scale dataset for malware detection. It addresses limitations of previous datasets by offering consistent feature extraction, a diverse set of malware families, and metadata suitable for both traditional machine learning and deep learning methods.

Heterogeneous Features: The dataset contains multiple types of features that capture complementary aspects of PE files:

- *Histogram and Entropy Features:* Each with 256 values representing byte distributions and entropy measures. These features represent local byte-level patterns and are compatible with CNN after reshaping into 16×16 single-channel images.
- *Metadata Features:* General file metadata such as size, number of sections, linker version, and image base. These tabular features capture high-level key file properties that often distinguish benign files from malware.
- *Header, Section, Imports and Exports Features:* There are structured numeric and categorical features that capture the program's structure, including section names, imported and exported functions, and other header-level details. These features provide additional semantic information that complements the raw byte-level statistics.

3.2 Dataset preparation

After filtering out unlabeled samples (label = -1), features were normalized or reshaped according to the model architecture. Specifically, histogram and entropy features were reshaped to 16×16 images for CNN processing, while tabular metadata features remained as vectors for MLP processing. This heterogeneous structure allows hybrid CNN+MLP models to exploit both spatial byte-level patterns and high-level semantic metadata for improved malware detection performance.

3.3 Neural Network Architectures

We implemented and evaluated three neural network architectures under identical training conditions to assess their performance on the EMBER dataset: (i) a Multi-Branch MLP, (ii) a pure CNN, and (iii) a hybrid CNN+MLP model. Each model was trained with the Adam optimizer and sparse categorical cross-entropy loss for 50 epochs (batch size 256), using 20% of the training data for validation. To address class imbalance, class weights were computed dynamically.

3.3.1 Multi-Branch MLP: The EMBER dataset provides 2381 static features per binary, which we partitioned into nine feature groups: histogram (256), entropy (256), strings (104), general metadata (10), header (62), section (255), imports (1280), exports (128), and data directory (30). Each group was normalized independently using `StandardScaler`.

Each feature group was processed by a dedicated dense block consisting of a ReLU-activated dense layer, batch normalization, and dropout ($p = 0.3$). The number of hidden units was scaled by input dimensionality: 256 (imports), 128 (histogram, entropy, section), 64 (strings, header, exports), and 32 (general, data directory). Outputs from all branches were concatenated and passed through two dense layers (512 and 256 units, each with Batch Normalization and Dropout), followed by a softmax output layer for binary classification.

3.3.2 Pure CNN Model: To exploit spatial patterns in malware features, all EMBER feature groups were reshaped into two-dimensional matrices with a single channel, for example histogram (16×16), entropy (16×16), imports (40×32), strings (13×8).

Each branch consisted of two convolutional blocks, each comprising a Conv2D layer, max-pooling (with pool size adapted to the feature dimensions), batch normalization, and dropout ($p = 0.3$), followed by flattening. The resulting feature maps from all branches were combined and passed through fully connected layers with 512 and 256 units, each with ReLU activation and dropout ($p = 0.4$), then passed to a softmax output layer for binary classification.

3.3.3 Hybrid CNN+MLP (Proposed): The proposed hybrid architecture integrates convolutional neural networks (CNNs) for statistical feature distributions with multi-layer perceptrons (MLPs) for structured metadata.

- **CNN branch:** Histogram (256) and entropy (256) features were reshaped into 16×16 grayscale matrices. Each input passed through two convolutional blocks followed by flattening to produce feature embeddings.
- **MLP branch:** Structured metadata consisted of strings (104), general metadata (10), headers (62), sections (255), imports (1280), exports (128), and data directory (30). Each feature group was normalized and passed through a dense block consisting of fully connected layers, batch normalization, and dropout. The number of units in each block was scaled according to input dimensionality as given in the Algorithm 1.

Outputs from CNN and MLP branches were combined and passed through two fully connected layers, followed by a softmax output layer, as illustrated in Figure 1.

3.4 Rationale for Hybrid Architecture

The hybrid CNN+MLP leverages complementary inductive biases:

- CNN capture local relationships and spatial dependencies in statistical distributions (histogram and entropy).
- MLPs model nonlinear relationships in structured tabular metadata, headers, imports/exports, and sections.
- Feature fusion enables joint reasoning over spatial and structured representations, improving detection accuracy and generalization.

This design, illustrated in Figure 1, integrates spatial and structured perspectives into a unified classification framework.

Algorithm 1 Hybrid CNN + MLP for EMBER Malware Classification

Require: EMBER dataset features $X \in \mathbb{R}^{N \times F}$, labels $y \in \{0, 1\}^N$, epochs E, batch size B
Ensure: Trained hybrid CNN+MLP model
1: **Split features into groups:** Histogram, Entropy, Strings, General metadata, Header, Section, Imports, Exports, Data Directory
2: **Reshape CNN features:** Histogram and Entropy $\rightarrow 16 \times 16 \times 1$
3: **Normalize remaining feature groups**
4: **Define CNN branches:** Apply two Conv2D+ReLU layers with MaxPooling and Dropout, then Flatten (for Histogram and Entropy)
5: **Define MLP branches:** For each non-CNN feature group, apply Dense+ReLU, BatchNorm, Dropout
6: **Combine all branches:** Concatenate CNN and MLP outputs
7: **Fusion MLP:** Two Dense+ReLU layers with BatchNorm and Dropout
8: **Output layer:** $y_{\text{pred}} \leftarrow \text{Dense}(2, \text{Softmax})$
9: **Train model:** Minimize sparse categorical cross-entropy over E epochs with batch size B and class weights
10: **return** Trained hybrid CNN+MLP model

3.5 Performance Evaluation

A stratified 80/20 train-test split was applied and subsequently, 20% of the training set was used as validation data for model tuning. Each model was trained five times with random seed. The Table 3 shows the evaluation metrics that were applied to evaluate performance, that includes accuracy, precision, recall, F1-score and ROC-AUC. This approach enables fair comparison of malware detection methods.

Table 1: Hybrid CNN + MLP Architecture for Malware Detection

Layer (Type)	Output Shape	Kernel / Units	Activation	Dropout
CNN Branches (applied to both `hist_input` (16×16×1) and `entropy_input` (16×16×1))				
Conv2D	(16, 16, 32)	3 x 3	RelU	0.3
MaxPooling2D	(8, 8, 32)	2 x 2	-	-
Conv2D	(16, 16, 32)	3 x 3	RelU	0.3
MaxPooling2D	(4, 4, 64)	2 x 2	-	-
Flatten	(16, 16, 32)	-	-	
Conv2D	(1024,)	-	-	-
MLP Branches (per feature group input, with Batch Normalization)				
strings-input	(10,) - Dense	64	RelU	0.3
general-input	(10,) - Dense	32	RelU	0.3
header-input	(62,) - Dense	64	RelU	0.3
section-input	(255,) - Dense	128	RelU	0.3
imports-input	(128,) - Dense	256	RelU	0.3
exports-input	(128,) - Dense	64	RelU	0.3
datadir-input	(30,) - Dense	32	RelU	0.3
Fusion Layers (after concatenation, 3200 features, with Batch Normalization + Dropout)				
Dense	-	512	RelU	0.3
Dense	-	512	RelU	0.3
Dense (Output)	2	-	Softmax	-

3.6 Experiments

Experiments aimed to validate the following hypotheses:

- A hybrid CNN+MLP architecture Figure 1 that processes histogram and entropy features via CNN and metadata features via MLP achieves malware classification performance (accuracy, precision, recall, F1-score, ROC-AUC) superior to standalone CNN or MLP architectures on the EMBER 2018 dataset.
- Feature-specific machine learning method that combines CNN and MLP to handle heterogeneity in the dataset improves generalization and robustness, leading to reduced false positives and false negatives compared to single-architecture baselines.

Table 2: Method and Experiment Parameters.

Experiments	
Training data	Section 3.1
Training, validation, testing	80%, 10%, 10% data subsets
Initialization (weights)	Random
Epochs	50
Batch Size	256
Runs	5
Validation Split	0.2
Evaluation (fitness) metric	Table 3

– The hybrid CNN+MLP model demonstrates statistically significant improvements in ROC-AUC and F1-score, confirming the advantage of modality-aware architectures in malware detection.

Our malware dataset comprises Windows PE binary files, where the dataset, denoted as D, consists of two classes (benign and malicious) with N total instances (Section 3.1). The training set T is drawn from D and employed to train the classifiers:

$$T_i \subseteq N_i \subseteq D_i, \quad i \in 1, 2 \tag{1}$$

This ensures a training process that allows models to learn discriminative patterns and characteristics across feature groups. Method and experiment parameters are presented in Table 2, while evaluation metrics are defined in Table 3.

We conducted three sets of experiments:

Benchmark CNN Classifier: A re-implementation of a CNN architecture inspired by prior malware detection studies [15], adapted to process histogram and entropy feature representations with a convolutional architecture. This model serves as a strong baseline for evaluating malware classification performance.

Multi-Branch MLP Classifier: A multi-input architecture where each feature group from EMBER is passed through its own dense branch before feature fusion. This setup evaluates the discriminative power of structured tabular features without convolutional processing.

Hybrid CNN+MLP Classifier (Proposed): Our main contribution integrates convolutional branches for histogram and entropy features (reshaped as grayscale images) with multi-branch MLPs for all remaining tabular feature groups (Algorithm 1). Fusion layers combine all the representations learned before classification. The design and parameter configuration of our proposed Hybrid CNN+MLP architecture are outlined in Table 1 and Figure 1.

Table 3: Malware classification performance metrics.

Metric	Formula	Description
True Positive (TP)	$tp = \sum_1^n$	Total malware correctly predicted
False Positive (FP)	$fp = \sum_1^n$	Total benign incorrectly predicted
True Negative (TN)	$tn = \sum_1^n$	Total benign correctly predicted
False Negative (FN)	$fn = \sum_1^n$	Total malware incorrectly predicted
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	Rate of correct predictions
Precision	$\frac{TP}{TP+FP}$	Positive Predictive values
Recall	$\frac{TP}{TP+FN}$	True positive rate
F1-score	$\frac{2.Precision.Recall}{Precision+Recall}$	Harmonic mean of precision and recall
ROC-AUC score	$\int_0^1 \mathrm{TPR}(\mathrm{FPR})\, d(\mathrm{FPR})$	Probability the model ranks a positive above a negative

For each model, classification performance was measured using accuracy, precision, recall, F1-score, and ROC–AUC. To mitigate variance due to random initialization, all results were averaged over 5 runs with different seeds.

Table 4: Architectural Performance on the EMBER 2018 Dataset

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC (%)**
MLP	97.46 ± 0.15	97.5 ± 0.11	97.6 ± 0.2	97.6 ± 0.2	98.4 ± 0.3
CNN	75.72 ± 3.56	83.0 ± 0.4	71.7 ± 0.3	70.6 ± 0.3	82.13 ± 0.22
Hybrid	97.62 ± 0.08	97.8 ± 0.09	97.8 ± 0.1	97.7 ± 0.13	99.6 ± 0.08

4 Results and Discussion

The experimental results indicate that the Hybrid CNN+MLP architecture consistently achieves faster convergence and higher classification accuracy compared to the standalone CNN and MLP models (Figure 2). The MLP-only model exhibits high accuracy, closely following the hybrid approach. The CNN-only model converged quickly but plateaued early, suggesting limited

capacity to fully exploit tabular metadata features. In contrast, the MLP-only model demonstrated stable convergence but lacked fine-grained representational power for histogram and entropy features. By combining CNN branches for histogram and entropy with MLP branches for metadata, the Hybrid model achieved both accelerated convergence and higher final validation accuracy. Notably, the validation accuracy curve of the CNN-only model fluctuates more compared to the smoother trajectories observed for the MLP and Hybrid models, which is consistent with known behaviors in deep learning models when trained on heterogeneous feature types.

Across multiple validation runs ($n = 5$), the Hybrid CNN+MLP model achieved an average validation accuracy of 97.71 ± 0.08, whereas the MLP-only baseline reached 97.45 ± 0.15 and the CNN-only baseline reached 75.72 ± 3.56. These results suggest a strong performance advantage for the Hybrid and MLP models compared to CNN.

4.1 Statistical Evaluation

To evaluate performance differences, multiple statistical tests were conducted. The paired t-test comparing Hybrid and MLP yielded $t = 4.27$, $p = 0.0235$, indicating a significant improvement at the 5% level, whereas the Wilcoxon signed-rank test ($W = 0.000$, $p = 0.125$) was not significant, reflecting the conservative nature of non-parametric methods.

The Hybrid model significantly outperformed CNN, as indicated by the Mann–Whitney U test ($U = 16.000$, $p = 0.0286$). Across all three models, the Friedman test revealed significant differences ($\chi^2 = 8.000$, $p = 0.0183$). Post-hoc Mann–Whitney tests confirmed that Hybrid exceeded both MLP ($p = 0.0294$) and CNN ($p = 0.0286$), and MLP outperformed CNN ($p = 0.0294$). Paired t-tests for MLP vs. CNN and Hybrid vs. CNN ($p = 0.0011$ for both) further highlighted highly significant differences, whereas Wilcoxon tests remained non-significant.

Overall, these analyses show that the Hybrid model consistently outperforms CNN and offers modest gains over MLP, though the difference between Hybrid and MLP is not significant under conservative non-parametric tests.

4.2 Quantitative Performance Summary

Table 4 summarizes the quantitative metrics for all three architectures. The Hybrid CNN+MLP model outperformed both baselines across key metrics including accuracy, precision, recall, F1-score, and ROC-AUC. Notably, the Hybrid model achieved a ROC-AUC of 0.996, demonstrating superior discriminatory power in distinguishing benign from malicious samples by jointly leveraging histogram and entropy distributions and metadata features.

The hybrid approach reduces false negatives observed in CNN-only models (malware with subtle distributional anomalies) and false positives observed in MLP-only models (benign samples with unusual entropy patterns). These results illustrate that combining convolutional and fully connected pathways effectively

exploits complementary inductive biases present in the EMBER 2018 feature set.

4.3 Contributions

The key contribution of this research is the empirical evidence that integrating convolutional processing for distributional features with fully connected layers for metadata significantly enhances malware classification performance compared to standalone CNN or MLP architectures. As summarized in Table 4, the Hybrid CNN+MLP model consistently outperformed both baselines across all evaluation metrics, achieving the highest accuracy (97.6%), precision (97.8%), recall (97.8%), F1-score (97.7%), and ROC-AUC (99.6%). The model also demonstrated faster convergence, smoother validation trajectories, and improved robustness under class imbalance conditions.

These contributions directly address the research questions posed in this study:

- The Hybrid CNN+MLP achieved superior accuracy and ROC-AUC compared to both MLP and CNN baselines. Statistical tests confirmed that the Hybrid model significantly outperformed CNN, while its improvement over MLP was more modest and less consistent under non-parametric evaluation. This demonstrates that hybridization offers a tangible performance gain over single architectures.
- By combining convolutional layers for byte-distributional features (histogram, entropy) with fully connected layers for structured metadata (sections, imports, exports), the Hybrid architecture leveraged complementary feature spaces. The results show that neither CNN nor MLP alone fully captured this heterogeneity, highlighting the added value of integrating distributional and metadata signals.
- The Hybrid model showed smoother validation curves and superior performance under class imbalance, confirming stronger generalization. Its higher ROC-AUC and more stable precision–recall trade-offs indicate robustness when applied to unseen or skewed datasets, thereby answering the generalization question affirmatively.
- The Hybrid architecture reduced both false positives and false negatives compared to standalone baselines. In particular, it was more effective at detecting malware with subtle structural or statistical anomalies, demonstrating that hybridization improves robustness against difficult-to-classify samples.

In summary, the Hybrid CNN+MLP architecture not only improves classification performance but also enhances generalization and robustness under class imbalance conditions. By aligning the model architecture with the multi-modal nature of EMBER features, the hybrid approach provides a practical and effective solution for malware detection in real-world scenarios.

5 Conclusion

This study demonstrated and rigorously evaluated a hybrid CNN+MLP architecture for malware detection on the EMBER 2018 dataset. The proposed model leveraged CNN branches to extract features from histogram and entropy distributions, while MLP branches processed tabular metadata, enabling joint reasoning across heterogeneous feature types.

The results establish a robust framework for adaptive, multi-modal malware detection. Future work will investigate extending hybrid architectures to additional malware families, incorporating richer feature sets, and applying techniques such as data augmentation and adversarial training to improve generalization. The long-term goal is to develop self-adapting malware classifiers capable of maintaining high detection performance against evolving threats [17], contributing to the broader vision of autonomous and efficient computational and cyber-physical systems [18–20].

References

1. D. Gibert, C. Mateu, and J. Planes, "The Rise of Machine Learning for Detection and Classification of Malware: Research Developments, Trends and Challenges," *Journal of Network and Computer Applications*, vol. 153, p. 102526, 2020.
2. M. Gopinath and S. Sethuraman, "A Comprehensive Survey on Deep Learning based Malware Detection Techniques," *Computer Science Review*, vol. 47, p. 100529, 2023.
3. R. Lo *et al.*, "MCF: A Malicious Code Filter," *Computers Security*, vol. 14, no. 6, pp. 541–566, 1995.
4. K. Rieck *et al.*, "Automatic Analysis of Malware Behavior using Machine Learning," *Journal of Computer Security*, vol. 19, no. 4, pp. 639–668, 2011.
5. X. Ling *et al.*, "Adversarial attacks against Windows PE Malware Detection: A Survey of the State-of-the-Art," *Computers Security*, vol. 128, p. 103134, 2023.
6. T. Alsmadi and N. Alqudah, "A Survey on Malware Detection Techniques," in *2021 International Conference on Information Technology (ICIT)*, pp. 371–376, 2021.
7. K. Aryal, M. Gupta, and M. Abdelsalam, "A Survey on Adversarial Attacks for Malware Analysis," 2022.
8. J. Singh and R. Banerjee, "A study on single and multi-layer perceptron neural network," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 35–40, 2019.
9. I. Ben Abdel Ouahab, E. Lotfi, and M. Bouhorma, *Image-Based Malware Classification Using Multi-layer Perceptron*, pp. 453–464. 10 2021.
10. X. Larriva-Novo, C. Sánchez-Zas, V. A. Villagrá, M. Vega-Barbas, and D. Rivera, "An approach for the application of a dynamic multi-class classifier for network intrusion detection systems," *Electronics*, vol. 9, no. 11, 2020.
11. O. Zelený and T. Fryza, "Multi-branch multi layer perceptron: A solution for precise regression using machine learning," in *2023 33rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–5, 2023.
12. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
13. H. S. Anderson and P. Roth, "Ember: An open dataset for training static pe malware machine learning models," 2018.
14. A. Bensaoud, J. Kalita, and M. Bensaoud, "A survey of malware detection using deep learning," *Machine Learning with Applications*, vol. 16, p. 100546, 2024.
15. B. Chen *et al.*, "Adversarial examples for cnn-based malware detectors," *IEEE Access*, pp. 54360–54371, 2019.
16. J. Jiang and M. Stamp, "Multimodal techniques for malware classification," 2025.
17. S. Didi and G. Nitschke, "Evolutionary deep-learning malware classifiers," in *2025 IEEE Symposium on Computational Intelligence in Security, Defence and Biometrics Companion (CISDB Companion)*, pp. 1–5, 2025.
18. D. Nagar, A. Furman, and G. Nitschke, "The Cost of Big Brains in Groups," in *Proceedings of the 2019 Conference on Artificial Life*, (Newcastle, United Kingdom), pp. 404–411, MIT Press, 2019.
19. J. Watson and G. Nitschke, "Evolving Robust Robot Team Morphologies for Collective Construction," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, (Cape Town, South Africa), pp. 1039–1046, IEEE Press, 2015.

20. C. Mailer, G. Nitschke, and L. Raw, "Evolving Gaits for Damage Control in a Hexapod Robot," in *Proceedings of the Genetic and Evolutionary Computation Conference*, (Lille, France), pp. 146–153, ACM, 2021.

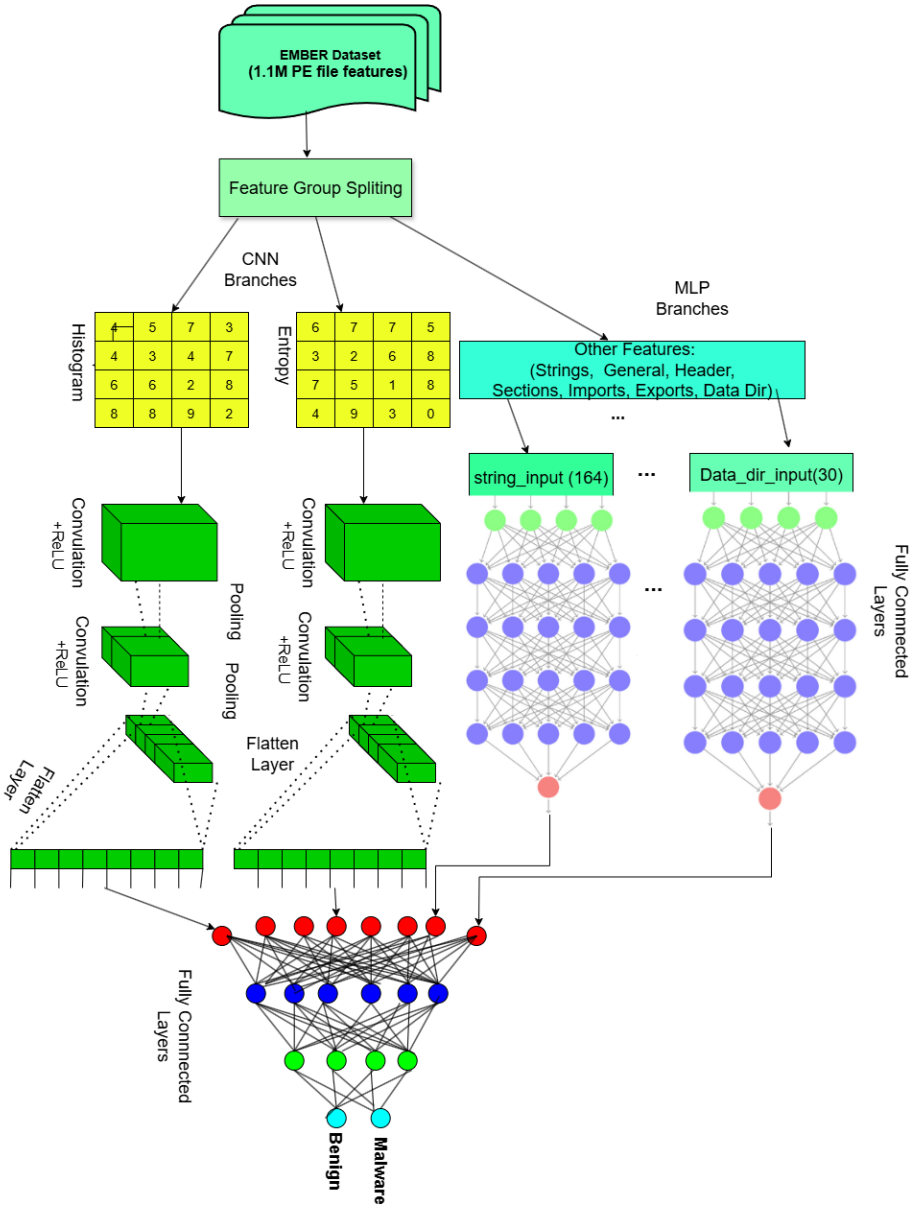

Fig. 1: Hybrid CNN+MLP Architecture: Multi-branch design where CNN branches (left) extract spatial patterns from histogram and entropy features, while MLP branches (right) process structured metadata, header, imports, exports, and section features. Feature representations from all branches are merged for joint learning, enabling the model to leverage both local patterns and global feature dependencies for robust malware classification.

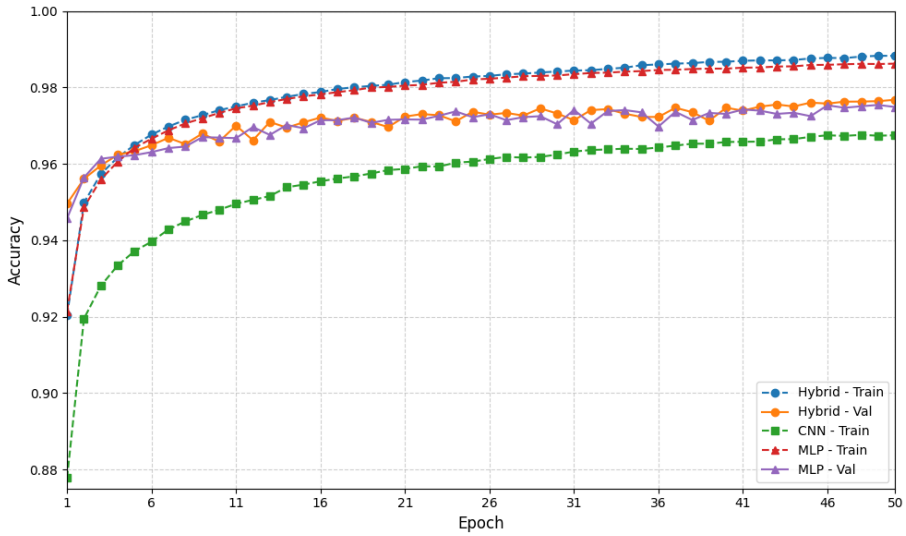

Fig. 2: Average training and validation accuracy over 5 runs for the classifiers: Hybrid (CNN+MLP), MLP-only, and CNN-only. Each plot shows both training and validation performance curves, except for the CNN, which exhibits noticeably lower validation accuracy.