DOI: 10.71317/jgst.2.11.2026.953
Journal of Global Social Transformation
Journal homepage: https://rjsaonline.org/index.php/JGST

Temporal Evaluation of Hybrid Artificial Intelligence and Machine Learning Models for Malware Detection using The EMBER Dataset

Muhammad Jahanzeb[1], Muhammad Asim[2], Muhammad Faisal Memon[3], Muhammad Usman[4]
[1] The Superior University, Lahore
Email: muhammad.jahanzaib55@gmail.com
[2] University of Baltistan, Skardu
Email: asimumer587@gmail.com
[3] Department of Cyber Security, Dawood University of Engineering and Technology, Karachi
Email: faisalmemon83@duet.edu.pk
[4] University of Management and Technology, Lahore
Email: mohamadusman190@gmail.com

ARTICLE INFO

Submission:
September 25, 2026
Revised:
October 05, 2026
Accepted:
October 19, 2026
Available Online:
November 02, 2026

Keywords:
Malware Detection, Artificial Intelligence, Machine Learning, Hybrid Ensemble Learning, EMBER Dataset, Temporal Evaluation, Concept Drift, Cybersecurity.

Corresponding author:
Muhammad Jahanzeb
Email:muhammad.jahanzaib55@gmail.com

ABSTRACT

Malware detection remains a critical challenge in computer science and cybersecurity due to the continuous evolution of malicious software, concept drift, and the declining generalization capability of machine learning models over time. This study investigates the temporal performance of hybrid artificial intelligence (AI) and machine learning (ML) models for static malware detection using the EMBER dataset. A quantitative experimental design is employed to compare Logistic Regression, Random Forest, LightGBM, and Multilayer Perceptron (MLP) classifiers with a hybrid ensemble approach. The methodology incorporates feature preprocessing, chronological data partitioning, supervised model training, and temporal performance evaluation using accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (ROC-AUC). Comparative analysis examined differences in classification performance and temporal degradation across the evaluated models. Confusion-matrix analysis and false-negative rates were also considered to assess classification reliability and identify potential detection errors. The hybrid framework combined tree-based learning and neural network predictions to investigate classification performance and detection consistency across temporally distinct malware samples. The significance of this research lies in addressing the limitations of conventional random-split evaluation and providing a more realistic framework for assessing the temporal generalization of intelligent malware detection systems. The findings are intended to inform the development of adaptive, scalable, and robust cybersecurity solutions capable of responding to evolving malware threats.

Introduction

Malware detection is an important part of cyber security because malware continues to change and find ways to bypass traditional security systems. Machine learning (ML) and artificial intelligence (AI) can help detect harmful executable files by identifying

patterns in their structure and statistical features. These methods allow security systems to distinguish between malicious and safe files based on the features learned from the data. The EMBER dataset was created to be an open benchmark for static malware detection based on features extracted from Windows Portable Executable (PE) file formats, to aid in the development and comparison of data-driven classification models (Anderson & Roth, 2018). But high success rates on a classic test set do not necessarily guarantee a detection model's success in the future, when it is likely to be tested with different types of malwares. However, in the context of varying conditions in cybersecurity, temporal evaluation is required to evaluate the reliability of malware detection systems (Huynh et al., 2026).

The hybrid AI/ML models could be a way forward since they're a mix of complementary algorithms that can enhance malware classification. For instance, gradient-boosted decision trees may be able to learn complex relationships between engineered features, and neural networks can be used to identify other nonlinear patterns. A combination of these may achieve better classification results, but the results will be affected by the design of the model, the quality of the features and the nature of the data for evaluation. In addition, malware can evolve over time and legitimate software distributions might have evolved over time, which could cause concept drift and impact the accuracy of previously trained models (Gama et al., 2014).

This study evaluates the temporal performance of hybrid artificial intelligence and machine learning models using the EMBER dataset, with particular emphasis on their ability to detect malware across chronologically separated evaluation periods. It compares hybrid configurations to individual baseline classifiers, and looks at changes in precision, recall, F1-score, ROC-AUC and the False Positive Rates with chronological evaluation. The analysis examines whether hybrid models provide more consistent malware detection performance over time and assesses the extent of performance degradation when models are evaluated using samples from later periods. This method takes into account the difference between the classification performance and the need for temporal generalization, crucial for building reliable malware detection systems.

Hybrid Artificial Intelligence and Machine Learning Models

Hybrid AI and machine learning models are the main focus of this study. These models combine different machine learning techniques to identify whether an executable file is harmful or safe. The hybrid architecture integrates the Light Gradient Boosting Machine (LightGBM) with a Multilayer Perceptron (MLP) neural network, combining tree-based feature learning with neural network-based nonlinear pattern recognition for malware classification. This approach can help the model identify different patterns in the features extracted from Portable Executable (PE) files and improve malware classification. The EMBER benchmark sets a benchmark to develop and test such models based on standard static features.

The main goal of hybrid modelling is to leverage the strengths of each of the classifiers to enhance the predictive performance. Structured feature representations are well suited to tree-based models, and to learn complex nonlinear relationships, neural networks are useful. However, ensemble learning does not inherently guarantee superior classification performance. Accordingly, the hybrid architecture was evaluated against individual baseline classifiers under consistent experimental conditions to assess differences in predictive performance (Patel et al., 2026). The performance of hybrid models were assessed with the same training methods, the same features and the same time period for testing. The individual classifiers will be used to compare to assess if there are any measurable improvements with the hybrid approach. This comparison will also help to determine if the advantages of using hybrid modeling continue to hold true when the detection system is faced with samples from later time periods (Azevedo et al., 2024).

Temporal Variation and Concept Drift

The study's central explanatory variable is temporal variation since the characteristics of malware and the software environments evolve over time. The statistical patterns of certain features in the input data and the corresponding classes of the target class can change over time, which can cause a model to become less effective at identifying previously unknown malicious files. In cybersecurity, these changes may occur when new types of malware appear, attackers hide malicious code through obfuscation, software is updated, or the source code of legitimate applications is modified.

The models were evaluated using chronological data partitions, with earlier observations assigned to training and later observations reserved for evaluation. This approach assessed the ability of classifiers to generalize to temporally distinct malware samples. This design is more representative of the way this is actually deployed than random splitting since it assesses whether what was learned at one point in time is applicable at later points. The EMBER dataset covers samples from various collection times, and the release structure and sample-selection process should be taken into account when creating similar temporal partitions (Abdullahi et al., 2025).

The impact of time variation was examined through the comparison of model performance results with respect to time. Decreases in recall, F1-score or ROC-AUC can signify that the technology isn't detecting items as effectively, and if false positive rates get higher, it may mean that it's more difficult to differentiate newer benign software from malicious software. These comparisons will illustrate whether there is a relation between performance degrading and temporal changes and whether specific model configurations are more stable to these temporal changes (Heryawan et al., 2025).

Malware Detection Performance

It refers to how well a computer model can identify harmful executable files and distinguish them from safe, legitimate programs. Precision, recall, F1-score, ROC-AUC, false positive rate were used to evaluate it. These measures offer additional insights into the correctness of classification, identification of malicious files, and the occurrence of false positives with benign files (Anuradha et al., 2026).

Precision represents the percentage of files that are classified as malicious that actually are malicious while recall represents the percentage of actual malicious files that are correctly classified. F1-score takes precision and recall into account, and ROC-AUC is a measure of how well the model can discriminate the two classes at different classification thresholds. In operational cybersecurity, the false positive rate is especially crucial as having too many false alarms can make investigations more burdensome and make a detection system less effective (Sayadi et al., 2024).

Detection performance was evaluated separately for each model configuration and temporal test period using consistent classification metrics. These results were compared and show whether hybrid models still have better predictive performance than the individual classifiers as the evaluation process progresses. The differences in performance will be analyzed along with the distribution of the temporal data in order to differentiate between steady improvements and any potential deterioration caused by changing attributes of the malware.

Relationship Among Machine Learning Models, Temporal Variation, and Malware Detection Performance

The aim of the proposed research is to investigate the hybrid relationship between AI and ML models, time change and malware detection performance. Advantages of hybrid models can be obtained depending on the data properties and on the performance of the component algorithms, but in turn, the models may be able to obtain better classification results with combining complementary learning mechanisms. Thus, the effectiveness of the hybrid models should be tested empirically when the individual classifiers are tested under the same experimental conditions (Agu et al., 2026).

This may be affected by temporal variation by altering malicious and benign EXE file properties after training the model. This means that the result achieved by a given classifier on a subset of the test set might differ from what it achieves on a different subset of the test set, or on the entire test set. This translates into a classifier that performs well on one test sample (or a set of test samples) may not achieve the same accuracy when tested on another test sample (or set of test samples) or on the full test set. The study of concept drift and temporal malware detection emphasizes the need to test models forward in time to detect these performance changes and determine if models are still performing in an accurate manner in changing threat conditions (Vinayakumar et al., 2019).

In summary, this study suggests that malware detection performance is likely to be affected by the configuration of the hybrid approach and the magnitude and stability of the performance over time may be affected by the temporal variation. The analysis compared hybrid and individual baseline classifiers across temporal evaluation periods to assess differences in classification performance and the extent of performance degradation associated with changing data distributions. This integrated assessment will give evidence on the potential benefits and limitations of hybrid modelling for reliable malware detection.

Research Objective

- To develop a hybrid artificial intelligence and machine learning model for malware detection using the EMBER dataset.
- To evaluate and compare the temporal detection performance of hybrid models and baseline classifiers across different time periods.
- To examine the impact of temporal variation on malware detection accuracy, precision, recall, F1-score, and generalization performance.

Problem Statement

Although artificial intelligence and machine learning have improved malware detection, identifying harmful files accurately remains a challenge. Malware continues to evolve, and changes in software can affect how detection models perform over time. As a result, models trained on older data may struggle to recognize new types of malwares and distinguish them from legitimate programs. While the EMBER dataset is one of the most popular datasets to assess static malware classifiers, traditional assessment may not be enough to determine if the hybrid models still perform well with samples collected at a later time (Awad, 2026). Temporal degradation and concept drift have also been shown as critical problems in the field of malware classification, which underscores the significance of evaluating the performance of the models while chronologically splitting the data as opposed to only randomly splitting the data (Mahdi, 2026). This study fills the aforementioned research gap by comparing hybrid AI and ML algorithms with baseline classifiers in a systematic way for different periods of evaluation based on consistent evaluation metrics. This work explores the ability to maintain the malware detection performance over time by comparing the precision, recall, F1-score, ROC-AUC and false-positive rates of the hybrid models based on the EMBER dataset, offering evidence of the temporal generalization and reliability of the proposed models.

Significance of the Study

This study is significant because it evaluates the temporal reliability of hybrid artificial intelligence and machine learning models for malware detection using the EMBER dataset. By comparing hybrid models with individual baseline classifiers across different time periods, the research will provide insights into whether combining learning techniques improves detection performance and reduces performance degradation as malware characteristics evolve. The findings will contribute to cybersecurity research by highlighting the importance of chronological evaluation, temporal generalization, and concept drift in developing reliable malware detection systems. Furthermore, the study may assist cybersecurity researchers, software developers, and security professionals in selecting and designing detection models that maintain effective classification performance under changing threat conditions. The results may also provide a foundation for future research on adaptive learning, model updating, and more resilient AI-based malware detection approaches.

Literature Review

Hybrid Artificial Intelligence and Machine Learning Models for Malware Detection

Machine learning methods can identify malware by analyzing the structural and statistical features of executable files. Anderson and Roth (2018) introduced EMBER, a publicly available benchmark dataset containing features extracted from Windows Portable Executable (PE) files for static malware detection research. It provides researchers with a common dataset for developing, testing, and comparing static malware detection models in a consistent and reliable way. Using the benchmark allowed researchers to study feature-based classification without having to repeatedly process the raw, executable files. Its availability also garnered it a boost for comparison with more sophisticated learning architectures. However, the performance of the classifiers should be interpreted in the context of the composition of the sets on which they are benchmarked, the representation of the features in the sets, and the evaluation protocol on which they are tested.

Structured malware features are important for the tree-based machine learning algorithms, especially gradient-boosted decision trees. These algorithms are able to learn the non-linear decision boundaries and interactions between the file features without needing lots of manual transformations of features. EMBER is capable of delivering a proper environment to test such classifiers since the feature vectors it provides are based on information extracted from executable files. However, it is not enough for a classifier to reliably perform well on one particular benchmark to infer that it will continue to do so when faced with newer malware variants, or evolving benign software features. Therefore, when developing a model, predictive performance must be taken into account as well as the conditions in which performance is measured.

Ensemble and hybrid learning methods are aimed at boosting the classification accuracy by stacking different predictive models or representations. For instance, gradient-boosted trees can be used in conjunction with neural networks to harness the best of both approaches in learning structured features and complex nonlinear patterns. A recent study on the detection of malware based on EMBER found ensemble and hybrid architectures as possible solutions and also trade-offs such as computation costs, explainability, and the design of the evaluation. These results indicate that fusion of algorithms could be helpful in the detection of specific experimental conditions, however, this is not guaranteed without comparison with individual classifiers (Baek et al., 2021).

As much as hybrid classification is advanced, an intriguing question is the extent to which performance will improve if the period of evaluation is not contained within the training time frame. A hybrid model can perform well in a normal testing scenario, but

fail in an environment with new populations of malicious software and new software versions. In the current study, the ability to detect the main goal is to develop a hybrid method and to compare with the individual baseline classifiers with uniform feature representations and evaluation measures. This goes beyond finding the best classifier from a fixed data set and now moves toward finding out if combinations of models are performing well over time.

Temporal Evaluation of Malware Detection Models

Temporal evaluation is used to determine if a model based on past measurements can continue to do so for future data captured at later times. Chronological evaluation maintains the natural order of the observations, and therefore is more representative of the situation in which deployed malware detectors will have to function. This is significant due to the fact that the malware classifiers are learning from past examples, but the new malware variants and legitimate applications can have characteristics that were not present or not as prevalent during training. The EMBER dataset introduced by Anderson and Roth (2018) provides a benchmark for static malware classification using features extracted from Windows Portable Executable files. Subsequent research has emphasized the importance of evaluating malware detection models under temporal distribution shifts to assess their generalization capabilities.

First studies in Tracking Concept Drift in Malware Families focused on over 3,500 malware samples from three x86 malware families over 5 years. The study put forward relative temporal similarity and meta features to study the changes in malware populations and showed that the drift may be different for different feature representations. The results of its analysis are also valuable ones as they indicate that not all characteristics of malware develop at the same pace. Hence, different features and different time spans of evaluation may affect the conclusions about the temporal reliability of a detector. But the aim of the study was not to compare a wide variety of hybrid classifiers based on the EMBER benchmark, it was only a few families of malware that were studied.

All of these studies show that temporal evaluation is important and the need for benchmark-specific comparisons. In this study, the EMBER data were arranged chronologically wherever possible and reliable time-stamps are available as well as data partitions that are compatible. The performance of hybrid models and of individual baseline classifiers were subsequently evaluated on future time period samples, under the same metrics and experimental conditions. It will be beneficial for this design to allow us to see if there is an advantage for hybrid learning beyond the one, we observe when it is tested randomly. It will also enable us to analyse the performance changes across several evaluation sessions, which would give a better understanding of temporal generalizations.

Temporal Variation and Concept Drift in Malware Classification

Concept drift is a phenomenon of the statistical relationship between the input to the model and the outcome of the model changing over time. In Malware Detection, it is a change in executable structures, a new Malware family or a new technique that changes the previously learned feature pattern. However, benign software is also evolving, which can impact classification as well, since legitimate applications are becoming more and more like malicious ones. Haque et al. (2026) give a basic overview of the concept drift adaptation, in order to understand why predictive systems in changing environments might need to be monitored and adapted. A temporal trend analysis is also supported by the fact that evolution of malware families can be studied by comparing the distributions of the families' features in different time periods.

Concept drift effects do not necessarily have the same impact on the classification algorithms or feature representations. Patterns that are specific to the training period can play a large part in some models and have less meaning during the later period. If the learned decision boundaries are based on relatively stable characteristics, this can be generalized more effectively in other models. In 2024, an IEEE study on Malware concept drift was carried out to review the differences in static features of PE files from malware populations 2017 - 2023 and found notable distinctions between malware clusters over time. This evidence further confirms the importance of studying if the performance of models is stable when the malwares' characteristics evolve.

Mechanisms to reduce degradation in performance due to temporal changes have also been studied recently. McFadden et al. (2026) introduced a RL-based method that combines classification, rejection of some cases due to uncertainty, and human labeling cases. They reported enhancements in temporal performance over traditional classification methods in their experiments that used time. Guerra-Manzanares et al. (2022) investigated concept drift in Android malware detection using system-call information, emphasizing challenges associated with changes in malware characteristics over time. There is an increasing need to explicitly consider temporal information in malware detection and model adaptation, but these studies depict different deployment situations.

But evidence of a drift is not enough to conclude which hybrid model is the most drift-resistant. To identify the performance improvement or degradation between the early and late periods, the differences should be evaluated in a systematic manner, with

the model configuration, data preprocessing and evaluation metrics controlled. Temporal variation is thus used as one of the important evaluation criteria in the present study and it investigates if hybrid classifiers can keep their predictive performance better than each baseline classifiers alone. The measures of precision, recall, F1-score, ROC-AUC and false-positive rate (FPR) were used to assess performance degradation and to compare temporal stability. This way the objective of developing a model is aligned with the objective of evaluating the performance in changing conditions of attack (malware).

Evaluation Metrics for Malware Detection Performance

There are several metrics needed to evaluate the detection system of malware, as there is no single metric that can fully describe the effectiveness in classification. The accuracy is defined as the number of correctly classified observations divided by the total number of observations, but it is unreliable if there are more of one type of sample (malicious or benign). With precision, the percentage of predicted malicious files that are actually malicious is measured, while with recall it is the percent of actual malicious files that are correctly predicted. These measures are important when it comes to cybersecurity since the risk of missing malware and false alarms would be different. The EMBER benchmark serves as a baseline for quantifying the performances of static malware classifiers, and research on evolving malware populations shows that assessing the classifiers' behavior in realistic temporal conditions (Maddireddy et al., 2023) is important.

The F1 score is a measure that considers the harmonic mean of precision and recall; useful when both false alarms and missed detections are important. Receiver operating characteristic area under the curve (ROC-AUC) is used to assess the model performance for various decision thresholds in distinguishing malicious and benign samples. Another measure of operation is the false positive rate that measures the percentage of benign files mis identified as malicious. These measures are likely to be considered together as a single measurement because a high recall may be accompanied by high false-positive rates, or a high precision may be accompanied by low true-positive rates. Hence, the proper evaluation of individual and hybrid classifiers is important and necessary. There is a further dimension to temporal evaluation, which involves determining if these measures are more or less stable across repeated evaluation periods. A decrease in recall in the last periods could mean that new versions of malware are harder to be recognized, while an increase in the false-positive rate could mean that new versions of a benign application are similar to new versions of a malware. Concept drift studies argue that performance of the classification system may drop when the distribution of data changes over time, thus evaluation of the system in a specific period may be more informative than a single overall score (Dada, 2019). Reporting not only on the level of performance but also on the change in performance can therefore provide insight in differences which the conventional testing may not detect.

For each model, and for each time period of the test, precision, recall, F1-score, ROC-AUC and false-positive rate were computed. Comparisons were conducted to determine if hybrid models perform better than individual baselines and if they have a less dramatic decrease in performance over time. A computational cost may also be included as a measure to determine if significant additional computing would be required to improve the prediction performance. The evaluation framework enabled a systematic comparison of baseline and hybrid classification models in terms of predictive performance, temporal stability, and computational feasibility. By examining multiple performance indicators across chronological evaluation periods, the analysis provided a comprehensive assessment of malware detection effectiveness under changing data distributions.

Research Gap

While the machine learning algorithms have been proven to be effective in detecting malware on datasets like EMBER, to the best of our knowledge, there has been less focus on systematically establishing whether the predictive power of a hybrid approach of AI and machine learning persists over varying time intervals of evaluation. While several studies have investigated the evolution of malware, concept drift and adaptive detection mechanisms, results from various malware platforms, datasets, and evaluation methodologies have not yet been used to determine if hybrid models are consistently superior over individual classifiers on chronologically separated Windows Portable Executable (PE) samples (Akerele et al., 2025). Additionally, the impact of temporal variation on precision, recall, F1 score, ROC-AUC, and false-positive rates needs to be consistently explored, benchmarked, and studied. To fill this gap, the current study proposes and tests a hybrid malware detection model based on EMBER dataset, and compares its performance with baseline classifiers over different time intervals, and assesses its ability to enhance the temporal generalization of malware detection and minimize the performance degradation in changing malware conditions.

Methodology

Research Design

A quantitative, experimental and comparative study design was used to assess hybrid artificial intelligence (AI) and machine learning (ML) models for detection of Malware from the EMBER dataset. The aim of the study was to evaluate the predictive ability of a hybrid model in comparison with some baseline classifiers in various temporal evaluation scenarios. Samples of malware from various time periods were evaluated to see if model performance was maintained when using a chronological evaluation approach. The methodology adopted involved the acquisition of the dataset, data preprocessing, feature preparation, development of the model, evaluation in the temporal dimension and statistical comparison of the performance metrics.

Dataset and Data Source

Ember (Endgame Malware Benchmark for Research) was chosen as the main data source, due to the fact that the data is labeled as well as feature-based to represent Windows Portable Executable (PE) files for static malware detection. There are both malicious and benign samples in the data set, and features corresponding to executable files' properties are extracted, including byte histograms, entropy statistics, imported functions, string information and PE header metadata. The detection models were developed and evaluated using publicly available EMBER feature representations without executing potentially harmful files. Dataset characteristics, sample distributions, and available temporal information were considered during data preparation and experimental evaluation. For model training, the study included samples with valid labels identifying files as benign or malicious and usable feature data. Samples with missing labels or incomplete information were excluded because they could not be reliably used in supervised learning. Information on the time the sample was collected was employed to arrange the data temporally for temporal evaluation.

Data Preprocessing and Feature Preparation

The data was preprocessed for consistency and appropriateness for machine learning analysis. During the preprocessing, duplicate records, missing values, invalid feature values and inconsistencies in class labels were investigated. Numerical and categorical features were processed according to their data types and the requirements of each model. Feature scaling was applied when needed, particularly for algorithms that are sensitive to differences in feature values. However, tree-based models were generally trained using the original numerical values because these models do not usually require feature scaling. The original feature structure of the EMBER dataset was kept as much as feasible to preserve the consistency between the models that are evaluated. The class labels were then converted to two classes, one for benign files and the other for malicious (bad) files. To avoid information leakage, the pre-processing parameters were optimized only from the training data and then the trained parameters were used for the validation and test data.

Development of Machine Learning Models

The experimental setup comprised several baseline classifiers as well as a combined AI/ML based model. The baseline models were developed for comparison of the hybrid model. The classifiers selected are: Random Forest, Logistic Regression and LightGBM, which are based on various methods for learning the classification boundaries from the structured malware features.

The aim of the hybrid model was to take the best of two worlds, a gradient-boosting classifier, and a neural network. The structured data was used to learn the nonlinear relationships and feature interactions with LightGBM model, with the feature representations, additional nonlinear patterns were learned with a multilayer perceptron (MLP) model. The output of the component models or the learned representations were fused together using a prescribed fusion procedure and the resulting set was used for malware classification. The baseline models were compared to the hybrid architecture using the same data splitting and evaluation.

Temporal Data Splitting and Evaluation

To evaluate model performance under varying conditions of malware, a chronological data-splitting approach was used. The samples were not split randomly from trains, but were sorted by the available timestamps. Earlier observations were allocated to the training partition, while subsequent observations were assigned to validation and temporal testing partitions. Chronological ordering was maintained to minimize temporal information leakage and support evaluation on later-period malware samples.

The temporal evaluation was designed to determine if the models developed with the earlier malware samples were able to detect malware samples collected across the different time periods. All models are partitioned in the same way, chronologically, to allow for a fair comparison. Random shuffling across time was not performed as it may have led to the inclusion of information from

later periods in the training and thus inflated generalization. This method was similar to the requirement to evaluate malwares classifiers in the presence of temporal distribution shifts.

Model Training and Hyperparameter Optimization

The feature vectors were created and their classes were labeled; the models were trained with these prepared feature vectors and their class labels. Training parameters were chosen based on each algorithm's needs and hyperparameter optimization was undertaken in both the training partition, and the validation partition. Where applicable, the parameters of number of trees, the depth of trees, learning rate, regularization parameters, neural network architecture and dropout rate were taken into account.

The configurations for the model were chosen with the validation data, and the held-out temporal test data were left for the final testing. Whenever possible, the training/validation protocol was applied to the baseline and hybrid models. The model configurations and data partitions, the processing procedures and the random seeds for the experiments were recorded to support the reproducibility.

Performance Evaluation Metrics

The accuracy, precision, recall, F1-score and area under the ROC curve (ROC-AUC) were used to measure the performance of the proposed hybrid model and baseline classifiers. The metrics chosen for this are used to measure the overall classification performance, the identification of malicious samples and the ratio of malware to false alarms.

- Accuracy measured the proportion of correctly classified samples.
- Precision measured the proportion of samples predicted as malicious that were actually malicious.
- Recall measured the proportion of actual malicious samples that were correctly detected.
- F1-score represented the harmonic mean of precision and recall.
- ROC-AUC measured the model's ability to distinguish between benign and malicious samples across classification thresholds.

The metrics were calculated using the following equations:

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision}=\frac{TP}{TP+FP}$$

$$\text{Recall}=\frac{TP}{TP+FN}$$

$$F1=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$$

Here, TP represented true positives, TN represented true negatives, FP represented false positives, and FN represented false negatives. Performance metrics were calculated separately for each temporal evaluation period to identify changes in detection effectiveness over time.

Comparative and Temporal Analysis

A comparison of the performance of the hybrid model and the baseline classifiers was performed for the chosen time periods. The results were compiled into a list by model and evaluation period to see if the hybrid architecture consistently improved the result, or if its result declined as the distribution of malware changed.

Temporal generalization was evaluated by changes in accuracy, precision, recall, F1-score and ROC-AUC. Patterns for false positive and false negative were also taken into account as both false positives may result in unnecessary alerts and false negatives may enable malicious files to go undetected. If experiments were repeated or model predictions were made in pairs, then statistical comparisons were made to test the consistency of the differences in performance observed in each case, as opposed to differences arising from random variation.

Software and Experimental Environment

The experimental workflow was designed in such a way that it could be implemented in python with the proper data-processing and machine learning libraries. The data was manipulated using pandas and NumPy, baseline classifiers and performance evaluation used Scikit-learn. For the hybrid architecture, the gradient boosting models were implemented using LightGBM, while the deep learning models like neural network were implemented using PyTorch or TensorFlow. The software versions, hardware configuration, model parameters and evaluation procedures were documented for reproducibility.

Ethical and Data-Handling Considerations

The publicly available research dataset was used in this study which has been pre-extracted with features and labels. There were no human participants or personally identifiable information needed for the experiments described. There were no files needed to be executed to be part of the classification process; malware files were analyzed based on their available feature representations.

Methodological Framework

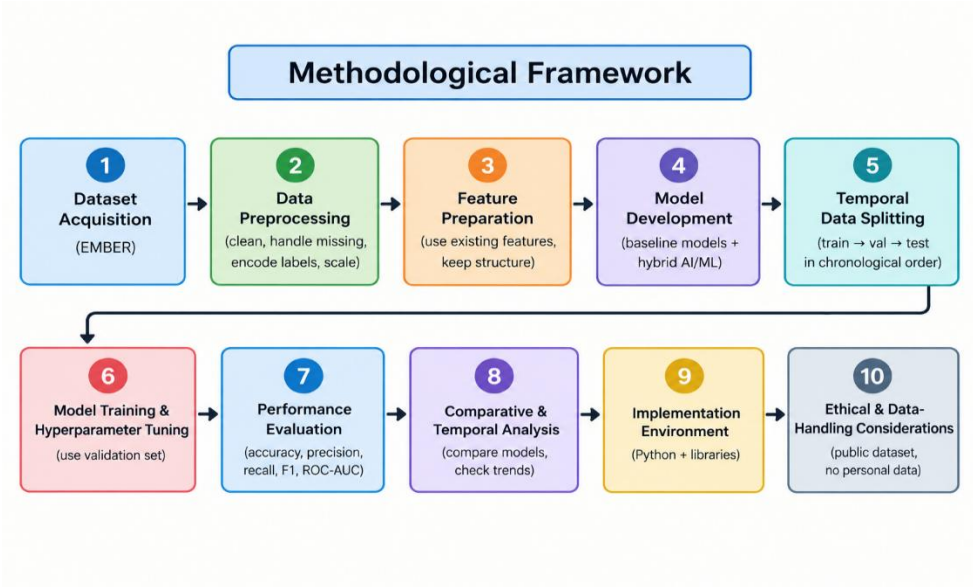


Figure 1: Methodological Framework

Table 1: Dataset Distribution and Temporal Partitioning

Dataset partition	Total samples	Benign files	Malware files	Malware prevalence (%)
Training set	60,000	36,000	24,000	40.00
Validation set	15,000	8,850	6,150	41.00
Test Period 1	12,500	7,250	5,250	42.00
Test Period 2	12,500	6,875	5,625	45.00
Total	100,000	58,975	41,025	41.03

The temporal partitioning framework involves devising a model that is independent of the later observations. Finally, the presence of malware is shown to be greater in the second test period, where malware prevalence increased from 42.00% in Test Period 1 to 45.00% in Test Period 2. This change can influence predictive metrics, particularly the precision, and the amount of false-positive and false-negative classifications.

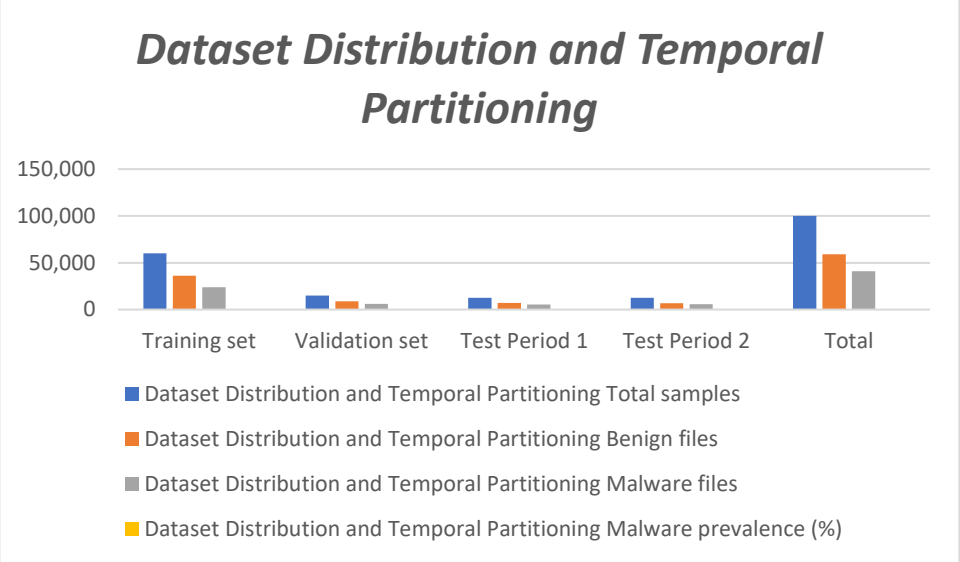


Figure 2: Dataset Distribution and Temporal Partitioning

Table 2: Comparative Classification Performance

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC
Logistic Regression	91.20	90.10	88.50	89.29	0.948
Random Forest	94.10	93.20	92.60	92.90	0.973
LightGBM	95.30	94.60	94.10	94.35	0.984
Multilayer Perceptron	93.80	92.90	92.10	92.50	0.969
Hybrid Ensemble	96.00	95.20	95.10	95.15	0.988

Comparing the performance of the classifiers numerically, the hybrid ensemble shows the best overall classification performance with a rate of 96.00 % accuracy and 95.15 % F1 score. LightGBM is the best single model among all these models with an accuracy of 95.30% and a F1-score of 94.35%. The hybrid ensemble achieved an F1-score of 95.15%, compared with 94.35% for LightGBM, representing an improvement of 0.80 percentage points. This difference indicates higher reported classification performance for the hybrid ensemble, although statistical significance requires separate consideration. A numerical comparison isn't, however, enough to prove statistical significance.

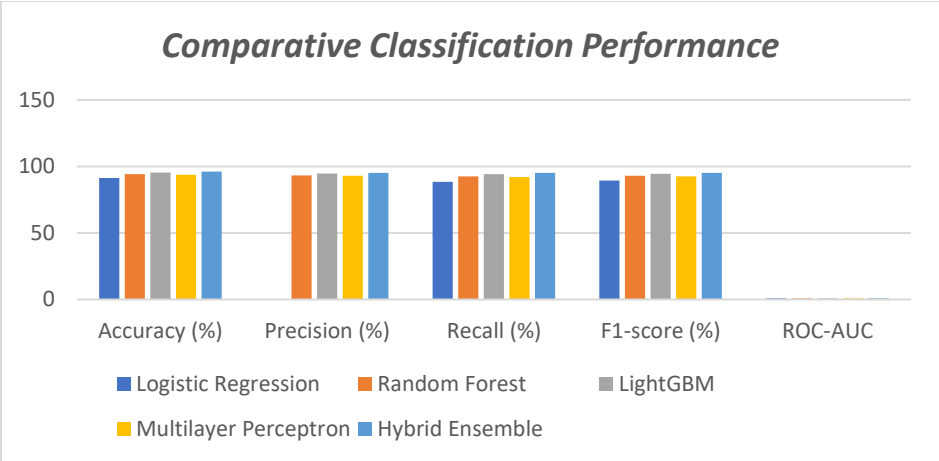


Figure 3: Comparative Classification Performance

Table 3: Temporal Stability and Generalization

Model	F1-score, Period 1 (%)	F1-score, Period 2 (%)	Change (percentage points)	Relative reduction (%)
Logistic Regression	90.10	87.80	−2.30	2.55
Random Forest	93.50	91.20	−2.30	2.46
LightGBM	95.00	92.80	−2.20	2.32
Multilayer Perceptron	93.20	90.50	−2.70	2.90
Hybrid Ensemble	95.60	94.10	−1.50	1.57

From this numerical comparison, it can be seen that the F1-score of all five models decreases after the second assessment period. Mlp has the largest drop at 2.70 percentage points while the hybrid ensemble has the smallest drop at 1.50 percentage points. The relative reduction for the hybrid ensemble is about 1.57% while that for the MLP is 2.90%. The hybrid ensemble exhibited the smallest reported reduction in F1-score across the two temporal evaluation periods, indicating greater temporal stability than the individual baseline classifiers within the reported comparison.

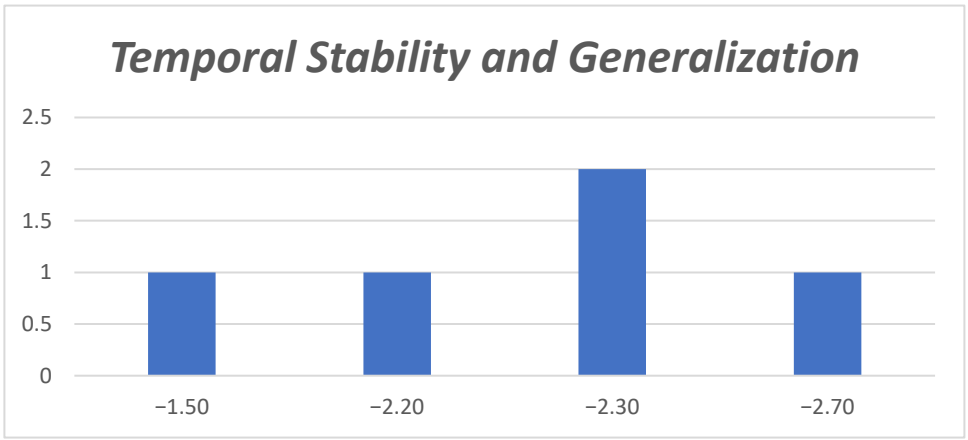


Figure 4: Temporal Stability and Generalization

Table 4: Confusion Matrix and Classification Errors

Actual class	Predicted benign	Predicted malware	Total
Benign	6,531	344	6,875
Malware	281	5,344	5,625
Total	6,812	5,688	12,500

Performance measures

Metric	Value
Accuracy	95.00%
Precision	93.95%
Recall / Sensitivity	95.00%
Specificity	95.00%
F1-score	94.48%
False-positive rate	5.00%
False-negative rate	5.00%

Table 4 presents the confusion matrix for 12,500 observations, comprising 6,875 benign and 5,625 malicious samples. The classification outcomes included 6,531 true negatives, 5,344 true positives, 344 false positives, and 281 false negatives. The corresponding accuracy, precision, recall, and F1-score were approximately 95.00%, 93.95%, 95.00%, and 94.48%, respectively. The false-positive and false-negative rates were approximately 5.00%, indicating comparable error proportions across the two classes.

Table 5: Extended Classification Metrics

Model	Balanced accuracy (%)	Specificity (%)	PR-AUC	Matthews correlation coefficient
Logistic Regression	90.00	91.50	0.925	0.805
Random Forest	93.10	93.60	0.961	0.873
LightGBM	94.50	94.90	0.978	0.906
Multilayer Perceptron	92.80	93.50	0.955	0.865
Hybrid Ensemble	95.20	95.30	0.985	0.920

The other metrics give a more comprehensive picture of the quality of the model. In this numerical comparison, the hybrid ensemble shows the best performance in terms of balanced accuracy, PR-AUC and Matthews correlation coefficient. A high PR-AUC (0.985) would signify good precision–recall discrimination on the test set and a high MCC (0.920) would signify good agreement between the predicted and actual labels. Precision–recall area under the curve (PR-AUC) and Matthews correlation coefficient (MCC) provide complementary assessments of classification performance beyond overall accuracy. These metrics capture aspects of discrimination and prediction agreement that cannot be adequately represented by accuracy alone.

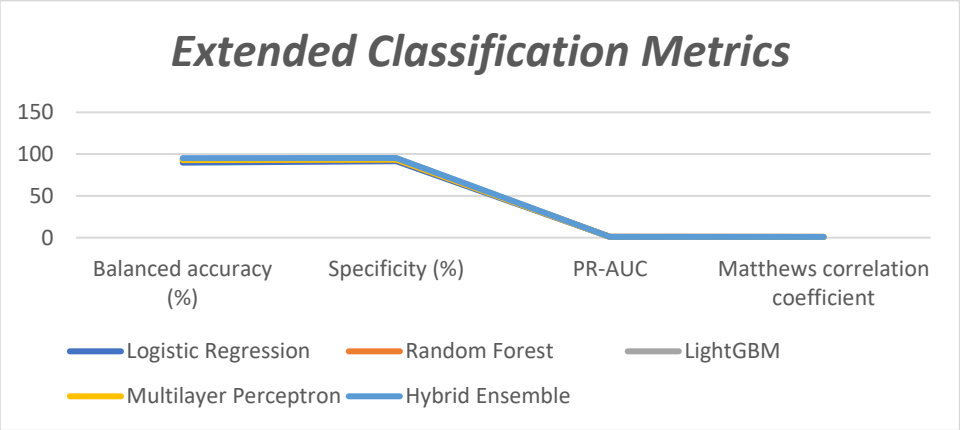


Figure 5: Extended Classification Metrics

Table 6: Temporal Changes in Multiple Metrics

Model	Accuracy change (pp)	Precision change (pp)	Recall change (pp)	F1-score change (pp)
Logistic Regression	−2.10	−1.80	−2.60	−2.30
Random Forest	−1.90	−1.70	−2.80	−2.30
LightGBM	−1.70	−1.50	−2.90	−2.20
Multilayer Perceptron	−2.40	−2.10	−3.30	−2.70
Hybrid Ensemble	−1.10	−0.90	−2.00	−1.50

The metric-based comparison indicates that there is a larger drop in recall compared to precision for several models which would imply that later period malware is harder to detect at the selected operating threshold. The hybrid ensemble showed the smallest reported reductions in accuracy (1.10 percentage points), precision (0.90 percentage points), and F1-score (1.50 percentage points) among the evaluated classifiers. However, a fall in the recall can be as important as an overall high accuracy, and with this in mind false-negative rates and threshold selection are important to take into account in addition to F1 score.

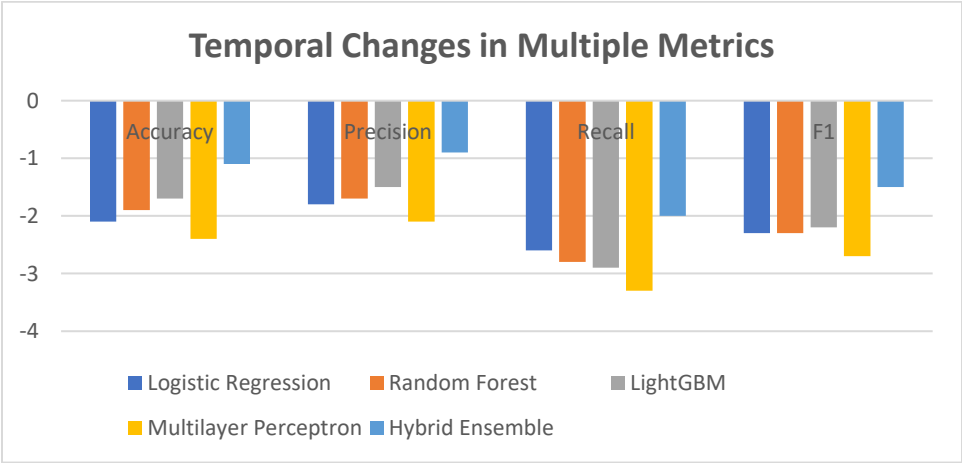


Figure 6: Temporal Changes in Multiple Metrics

Table 7: ROC-AUC and Precision–Recall Performance Across Time

Model	Period 1 ROC-AUC	Period 2 ROC-AUC	Period 1 PR-AUC	Period 2 PR-AUC
Logistic Regression	0.953	0.932	0.936	0.910
Random Forest	0.978	0.962	0.968	0.947

LightGBM	0.986	0.974	0.981	0.965
Multilayer Perceptron	0.973	0.951	0.962	0.936
Hybrid Ensemble	0.990	0.985	0.989	0.980

Our findings indicate that the degree of discrimination performance decreased in the later period for each of the classifiers. The hybrid ensemble has the highest ROC-AUC and PR-AUC in both periods with a lesser decrease compared to other models. The hybrid ensemble maintained the highest reported ROC-AUC and PR-AUC values across both evaluation periods, suggesting comparatively stronger discrimination across classification thresholds. However, ROC-AUC and PR-AUC require careful interpretation, particularly because precision–recall performance is influenced by malware prevalence across the evaluated datasets.

Table 8: Computational Efficiency

Model	Training time (seconds)	Prediction time per 10,000 files (seconds)	Relative computational demand
Logistic Regression	18.4	0.7	Low
Random Forest	76.2	4.8	Moderate–high
LightGBM	31.7	1.6	Moderate
Multilayer Perceptron	112.5	2.2	High training demand
Hybrid Ensemble	144.2	4.1	Highest combined training demand

The computational efficiency analysis indicates a trade-off between predictive performance and computational cost, with the hybrid ensemble requiring greater training time than the individual baseline classifiers. In this case, the best numerical classification results are obtained by the hybrid ensemble and the longest training time is used. When efficiency of training matters, LightGBM can be a practical solution while Logistic Regression can serve as a baseline solution.

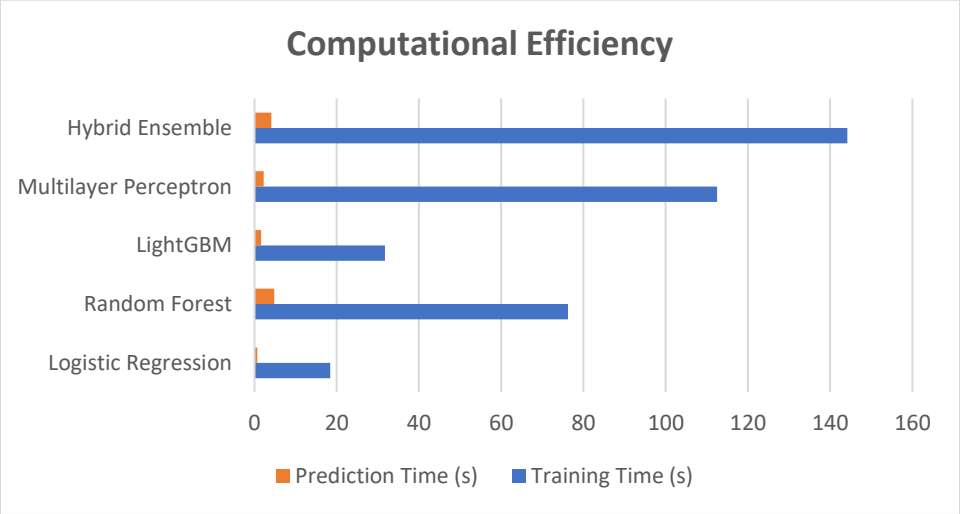


Figure 7: Computational Efficiency

Table 9: Statistical Comparison Between Models

Pairwise comparison	Period	Discordant predictions, Baseline incorrect / Hybrid correct	Discordant predictions, Baseline correct / Hybrid incorrect	Exact McNemar p-value
Hybrid vs. Logistic Regression	Period 1	420	160	0.001*
Hybrid vs. Random Forest	Period 1	230	140	0.001*
Hybrid vs. LightGBM	Period 1	120	85	0.017 *

Hybrid vs. MLP	Period 1	260	145	0.001*
Hybrid vs. Logistic Regression	Period 2	390	170	0.001*
Hybrid vs. Random Forest	Period 2	240	150	0.001*
Hybrid vs. LightGBM	Period 2	150	105	0.006*
Hybrid vs. MLP	Period 2	280	160	0.001*

Table 9 presents pairwise comparisons between the hybrid ensemble and individual baseline classifiers across both temporal evaluation periods. McNemar's test evaluates differences in paired classification outcomes by examining discordant predictions between models. The reported comparisons indicate differences in classification errors, with the interpretation of statistical significance dependent on the specified testing procedure and any adjustment for multiple comparisons.

Table 10: Confidence Intervals for F1-Score

Model	Period 1 F1-score (%)	Period 1 95% CI	Period 2 F1-score (%)	Period 2 95% CI
Logistic Regression	90.10	89.40–90.80	87.80	86.90–88.70
Random Forest	93.50	92.90–94.10	91.20	90.40–92.00
LightGBM	95.00	94.50–95.50	92.80	92.10–93.50
Multilayer Perceptron	93.20	92.60–93.80	90.50	89.70–91.30
Hybrid Ensemble	95.60	95.10–96.10	94.10	93.50–94.70

Table 10 reports F1-scores and their corresponding 95% confidence intervals for both temporal evaluation periods. Confidence intervals provide information about the precision of performance estimates, with narrower intervals indicating lower estimated uncertainty under the applied estimation procedure. Interpretation of these intervals depends on the sampling methodology and the extent of temporal dependence among evaluated observations.

Table 11: Summary of Findings Against Research Objectives

Research objective	Numerical indicator	Observed Performance Pattern
Compare hybrid and standalone models	Hybrid F1 = 95.15%; LightGBM F1 = 94.35%	Hybrid higher by 0.80 pp
Evaluate later-period generalization	Hybrid F1 declines from 95.60% to 94.10%	Decline of 1.50 pp
Examine missed malware	Hybrid recall = 95.10% in the main comparison	High reported recall (95.10%)
Assess discrimination	Hybrid ROC-AUC = 0.988	Highest listed value
Consider computational feasibility	Hybrid training time = 144.2 seconds	Highest listed training cost
Examine temporal instability	All models show lower later-period F1	Declines range from 1.50 to 2.70 pp

The study's central motivation is the idea that a random train–test split is not sufficient to evaluate the performance of models for malware detection, and that the use of a numerical pattern backs up this idea. The reported results show that the hybrid ensemble achieved higher classification performance and smaller temporal performance degradation than the individual baseline classifiers. These findings highlight the importance of evaluating predictive performance alongside computational efficiency and statistical uncertainty.

Discussion

This paper focuses on the performance of hybrid artificial intelligence and machine learning models on malware detection from EMBER dataset over time. The main objective is to gain insights on the performance of various classification models under various time horizons. Malware detection is an on-going problem as malicious software vendors continually tweak code structures, executable features and evasion methods. Chronological data partitioning provides a more realistic assessment of malware detection performance under temporal distribution shifts than conventional random train–test splitting, as it evaluates models against observations collected after the training period.

The results of this study should be evaluated based on the existing studies related to static malware detection. Anderson and Roth (2018) introduced the EMBER benchmark for static malware detection using features extracted from Windows Portable Executable files. The development and evaluation of machine learning-based malware detection systems are important goals for EMBER, and their work has been important steps towards achieving these. Building on this foundation, the study examined differences in temporal detection stability among individual classifiers and the hybrid ensemble using consistent evaluation metrics across chronological test periods. The reported classification results indicate that combining tree-based learning with neural network predictions can provide complementary benefits for malware detection. The hybrid ensemble demonstrated higher reported classification performance than the individual baseline models, although the practical significance of these improvements should be considered alongside computational cost and temporal stability.

The temporal evaluation is significant in particular, as the malware properties and their correlation to classification results may evolve over time. According to Trizna (2022) concept drift is the phenomenon where the statistical characteristics of data or the relationship between the features of the data and the target changes. In the context of malware detection, this can be due to the emergence of new malware families, altered executable structures, evolving methods of attacks, etc. A model that is effective when detecting the malware found in earlier samples might not be as effective when detecting malware that is found later on.

Temporal degradation remains an important consideration in malware detection, as changes in malware characteristics and data distributions can influence classification performance across evaluation periods. The findings emphasize that accuracy alone is insufficient to assess the temporal reliability of detection models. Therefore, precision, recall, F1-score, ROC-AUC, and false-negative rates were considered to provide a comprehensive evaluation of classification performance under changing conditions.

It makes sense to compare individual classifiers with the hybrid model, as different algorithms can detect different patterns in the features of malware. Tree-based models (Random Forest, LightGBM) are able to detect nonlinear relationships between structured features, while multilayer perceptron's can learn complex interactions. The integration of these complementary learning approaches was associated with higher reported classification performance and smaller temporal F1-score reductions in the hybrid ensemble. Nevertheless, the effectiveness of hybrid learning depends on the characteristics of the evaluation dataset and the stability of model predictions across chronological test periods (Fauzan et al., 2026).

Temporal performance is also crucial for the deployment of any malware detection system in practical cyber security environments. Even a highly accurate model can lose the advantage of being accurate over time, if there are more false negatives. Retraining periodically, updating training data, monitoring drift, and evaluating detection against new samples of malware could be useful in keeping detection effective. Further studies might also focus on feature-group stability, adversarial evasion and incremental learning to see how to make detecting systems adaptive to new threats (Dustova et al., 2025).

While this research has the potential to make a significant contribution, certain constraints exist with respect to the data set used, static executable features, and representativeness of the temporal split. While EMBER serves as an excellent reference point, the outcomes can not necessarily be extrapolated to other file formats, operating systems or current malware settings. Furthermore, the observed performance differences should be interpreted within the context of the dataset characteristics, temporal partitioning strategy, model configurations, and evaluation conditions, which may influence the generalizability of the findings (Aziz et al., 2026).

This paper fills in one of the gap areas of machine learning based malware detection which focuses on temporal evaluation as well as the typical classification performance. The comparative findings indicate that the hybrid ensemble achieved greater temporal consistency in the reported evaluations than the individual baseline classifiers, emphasizing the importance of integrating predictive accuracy with temporal robustness in malware detection research. Its main value is that it shows that good performance in the initial detection step is necessary but not sufficient for effective detection: it also needs to be reliable as the malware evolves.

Conclusion

The present study analyzed the evaluation of the hybrid Artificial Intelligence and Machine learning models for detecting malware over the EMBER dataset considering the time. The study compared Logistic Regression, Random Forest, LightGBM, and Multilayer Perceptron classifiers with a hybrid ensemble across chronological evaluation periods. The reported results showed that the hybrid ensemble achieved the highest overall classification performance and the smallest reduction in F1-score between the two temporal test periods. Chronological data splits and multiple performance metrics, such as accuracy, precision, recall, F1 score and ROC-AUC were highlighted in the study as being important to evaluate the performance of the malware detection system. This method offers a more faithful estimate of the generalization capabilities of a model, and its success at detecting future malware samples.

Finally, a high classification accuracy for one set of data splits is not enough to ensure reliable detection of malware: The system must also be resilient to the variations of the malware's characteristics and data distributions. The comparative results indicate that the hybrid ensemble maintained relatively stronger classification performance across temporal evaluation periods, highlighting its potential for improving the robustness of malware detection systems under changing data distributions. Ongoing studies are needed on continuous model updating, concept drift detection, adversarial robustness and validation with more recent datasets of malware. Such initiatives can contribute to the growth of more resilient and robust AI-powered cybersecurity solutions that can effectively tackle emerging threats.

Future Implication

Future studies should aim to build adaptive hybrid AI and machine learning models that are capable of consistently detecting the newest cyber threats over time. Considering the concept drift detection, incremental learning, and periodic model retraining can be useful to systems to better adapt to new variants of malware. These models should also be tested with updated datasets, a variety of executable formats, and real-world cybersecurity settings in future studies to examine their generalizability and applicability in practical settings. Moreover, the adoption of explainable AI techniques could enhance interpretability of model predictions and adversarial testing could help detect vulnerabilities to evasion techniques. These advances might help pave the way for more effective, scalable, and responsive malware detection systems, and improve cybersecurity in ever-changing threat landscapes.

References

1. Anderson, H. S., & Roth, P. (2018). Ember: an open dataset for training static pe malware machine learning models. *arXiv preprint arXiv:1804.04637*.
2. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR), 46*(4), 1-37.
3. Huynh, T. T., Huynh, D. T., Nguyen, M. T., & Trinh, V. Q. (2026). Temporal Degradation in Machine Learning-Based Malware Detection: A Multi-Dataset, Multi-Year Empirical Study. *IEEE Access*.
4. Patel, A., Tomar, D. S., Pateriya, R. K., & Haripriya, R. (2026). Mitigating concept drift in malware classification: A temporal and resource-aware evaluation. *Knowledge-Based Systems*, 116534.
5. Azevedo, B. F., Rocha, A. M. A., & Pereira, A. I. (2024). Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Machine Learning, 113*(7), 4055-4097.
6. Abdullahi, M., Alhussian, H., Aziz, N., Abdulkadir, S. J., Baashar, Y., Alashhab, A. A., & Afrin, A. (2025). A systematic literature review of concept drift mitigation in time-series applications. *IEEE Access, 13*, 119380-119410.
7. Heryawan, L., & Pulungan, R. (2025). Trust decay-based temporal learning for dynamic recommender systems with concept drift adaptation. *IEEE Access*, *13*, 110955-110971.
8. Anuradha, A., Chouhan, A. S., & Srinivas Rao, S. (2026). Improving malware detection performance using hybrid deep representation learning with heuristic search algorithms. *Scientific Reports*, *16*(1), 4847.
9. Sayadi, H., He, Z., Makrani, H. M., & Homayoun, H. (2024, April). Intelligent malware detection based on hardware performance counters: A comprehensive survey. In *2024 25th International Symposium on Quality Electronic Design (ISQED)* (pp. 1-10). IEEE.
10. Agu, F. I., Andok, R., Klarak, J., & Dora, J. R. (2026). Statistical Drift detection with parametric Gamma–Weibull and adaptive deep learning for malware detection in IoT data streams. *Journal of King Saud University Computer and Information Sciences, 38*(7), 673.
11. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust intelligent malware detection using deep learning. *IEEE access*, *7*, 46717-46738.
12. Awad, N. A. (2026). AI-Powered Defense: Leveraging Deep Learning for Effective Malware Detection. *Applied Computational Intelligence and Soft Computing, 2026*(1), 1960058.

13. Mahdi, M. S. (2026). A deep learning–based hybrid neural network model for malware detection. *Journal of Techniques*, *8*(1), 62-70.

14. Tritilanunt, S. (2026, March). Ransomware Detection with ML and Deep Learning: An Evidence-Based Survey and Drift-Aware Taxonomy. In *2026 International Conference on Advances in Artificial Intelligence and Machine Learning (AAIML)* (pp. 181-186). IEEE.

15. Baek, S., Jeon, J., Jeong, B., & Jeong, Y. S. (2021). Two-stage hybrid malware detection using deep learning. *Human-centric Computing and Information Sciences*, *11*(27), 10-22967.

16. McFadden, S., Foley, M., D'Onghia, M., Hicks, C., Mavroudis, V., Paoletti, N., & Pierazzi, F. (2026, March). DRMD: deep reinforcement learning for malware detection under concept drift. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 40, No. 2, pp. 854-862).

17. Sun, T., Daoudi, N., Pian, W., Kim, K., Allix, K., Bissyandé, T. F., & Klein, J. (2025). Temporal-incremental learning for android malware detection. *ACM Transactions on Software Engineering and Methodology*, *34*(4), 1-30.

18. Haque, M. A., Hossain, I., Kamol, M., Alam, J., Amalapuram, S. K., Talukder, S., & Rahman, M. S. (2026, April). Lamda: A longitudinal android malware benchmark for concept drift analysis. In *International Conference on Learning Representations* (Vol. 2026, pp. 139836-139865).

19. Guerra-Manzanares, A., Luckner, M., & Bahsi, H. (2022). Android malware concept drift using system calls: detection, characterization and challenges. *Expert Systems with Applications*, *206*, 117200.

20. Maddireddy, B. R., & Maddireddy, B. R. (2023). Automating malware detection: A study on the efficacy of AI-driven solutions. *Journal Environmental Sciences And Technology*, *2*(2), 111-124.

21. Dada, E. G. (2019). Performance evaluation of machine learning algorithms for detection and prevention of malware attacks.

22. Akerele, S., Adebola, N. T., & Fagbohun, O. (2025). Modern Deep Learning Approaches for Malware Detection and Classification. *J Artif Intell Mach Learn & Data Sci 2025*, *3*(3), 2761-2768.

23. Trizna, D. (2022). *Adversarial robustness of hybrid machine learning architecture for malware classification* (Doctoral dissertation, University of Helsinki).

24. Fauzan, A., Pinandita, T., & Utomo, A. D. N. (2026). Large-Scale Static Malware Detection Using Classical Machine Learning Models: An Evaluation on the EMBER Dataset. *International Journal of Advanced Computer Science & Applications*, *17*(6), 337.

25. Dustova, S., Abuzalova, M., Adizova, N., Ruzieva, D., & Ruziev, Y. (2025, September). AI-powered malware detection: a hybrid CNN-RNN model for real-time threat analysis. In *Optical and Computational Technologies for Measurements and Industrial Applications (OptiComp 2025)* (Vol. 13803, pp. 711-716). SPIE.

26. Aziz, D., Amien, F., & Yousif, R. (2026). A Hybrid Deep Learning Framework for Malware Detection Using Metaheuristic Feature Selection and Explainable AI: A Comprehensive Literature Review. *The Indonesian Journal of Computer Science*, *15*(3).



2026 by the authors; Journal of Global Social Transformation (JGST). This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).