

Proactive Malware Detection and Behavioral Analysis through an AI-Enhanced Cryptographic Framework

Hasanain Flayyih Hasan¹, Saif Hameed Abboud² and Haza Nuzli Abdull Hamed³

^{1,2}Department of Cybersecurity, College of Computer Science and Information Technology, Wasit University, IRAQ

³School of Computing, Faculty of Engineering, University Technology of Malaysia, 81310 Skudai, Johor, MALAYSIA

*Corresponding Author: Hasanain Flayyih Hasan

DOI: <https://doi.org/10.31185/wjps.1130>

Received 12 March 2026; Accepted 11 June 2026; Available online 30 September 2026

ABSTRACT: Today's malware is polymorphic, metamorphic, file-less and signature evasive. Although the classifiers based on machine learning (ML) and deep learning (DL) have seen great improvements in their ability to accurately detect attacks, they remain susceptible to adversarial perturbations, model-extraction attacks, and training data leakage. One of the difficulties with an AI inference pipeline is the absence of a direct incorporation of cryptographic guarantees of integrity and confidentiality into the pipeline. In response to this, this paper introduces a hybrid scheme called Hybrid AI-Enhanced Cryptographic Framework (H-AIECF) that merges static and dynamic feature engineering on the EMBER2024 benchmark with a cryptographic envelope composed of AES-256-GCM (confidentiality), SHA-256 (integrity), and ECDSA-signed verdicts (authentication). The portable-executable (PE) features of EMBER2024 are extracted using chi-square selection and Principal Component Analysis (PCA) to train a baseline Random Forest (RF) and a CNN-LSTM hybrid. In the binary detection task, the proposed framework achieves 98.74% accuracy, 98.52% precision, 98.91% recall, 98.71% F1-score and 0.81% False Positive Rate (FPR) that are all better than those of a legacy signature scanner, tuned Random Forest and a standalone CNN-LSTM. All in all, the overhead of the cryptographic layer is still below 5% of the total inference latency, demonstrating that the behaviour-aware detection and end-to-end cryptographic protection are not conflicting and mutually reinforcing characteristics of a trustworthy pipeline for malware-defence.

Keywords: Malware Detection, Artificial Intelligence, Cryptography, Machine Learning, EMBER2024



©2026 THIS IS AN OPEN ACCESS ARTICLE UNDER THE CC BY LICENSE

1. INTRODUCTION

The landscape of malware has changed over the last ten years. However, many malicious binaries can be reliably compared to a static signature database, whereas modern malicious files are usually polymorphic, fileless and designed to evade a learning-based detection system [1] [2]. Industry telemetry has hundreds of thousands of new variants every day, and a growing number are designed to evade machine learning classifiers [3]. There are three common flaws in signature-only techniques: They fail to generalize to variants they haven't seen before, they are dependent on human curation, and they cannot detect payloads that are packed, encrypted, or reflectively loaded [1,2].

⁴²*Corresponding author: hasanain.f.h@uowasit.edu.iq
<https://wjps.uowasit.edu.iq/index.php/wjps/index>

The trend for proactive malware detection now is Artificial Intelligence and particularly deep learning. Convolutional neural networks (CNNs), long short-term memory (LSTM) networks, transformers and graph neural networks (GNNs) have been used to tackle portable executable (PE) headers, bytestreams, opcode streams, application programming interface (API) traces, and even control-flow graphs, with achieved accuracy ranging from 95-99% on benchmark datasets [1; 3; 5]. The recently released EMBER2024 benchmark provides the labels and EMBERV3 feature vectors for more than 3.2 million files in 6 file formats and a special subset of 6,315 evasive malicious files, which was not detected by any of the analysed anti-virus engines [4].

But there are two structural weaknesses which give pause for concern about AI-only defenses. First, ML models can be vulnerable, as changes of the PE can be semantically correct, but will still change the verdict from malicious to benign without altering the behavior of the PE [6, 7]. Second, there are new attack surfaces in the AI pipelines such as poisoning the training data, extracting the AI models, and intercepting the inference output, particularly in the cloud-based deployments [8]. While cryptography can provide powerful mathematically-based assurances of confidentiality and integrity with primitives like SHA-256, AES-256, and homomorphic encryption, the research communities have usually worked in isolation with each other and these primitives are not natively integrated with AI-based malware detectors.

The lack of integration between the detection that is provided by AI and the cryptographic assurance is a critical research gap. As for the AI side, classifiers yield good accuracy, but offer no cryptographic guarantees that the features being used to make the classification have been altered, or that the classification results are genuine. Secure storage and transport are well known, and are not enough, on their own, to detect malicious activity on the cryptographic side. The system of malware defense needed for contemporary adversaries needs to be combined, detecting and protecting with the help of AI.

This study has three primary objectives. First, an AI-Enhanced Cryptographic Framework (AECF) is designed that combines a Random Forest with a CNN-LSTM hybrid, along with ECDSA-signed verdicts and a SHA-256 integrity check, on the EMBER2024 dataset, which is used for confidentiality (AES-256-GCM) and integrity (SHA-256). Secondly, to test the framework empirically with a signature-based baseline and a standalone Random Forest and CNN-LSTM with accuracy, precision, recall, F1-score, and FPR. Third, to describe the security and performance compromises of adding a crypto layer to a behavioral malware-detection pipeline.

The key contributions of this work are summarized below. To ensure the accuracy of detection while maintaining security guarantees, a Hybrid Artificial Intelligence-Enhanced Cryptographic Framework (H-AIECF) is proposed that combines machine learning and deep learning models with cryptographic mechanisms. Second, static and dynamic feature engineering is performed on EMBER2024 dataset to boost model performance using chi-square feature selection and PCA. Third, it includes an end-to-end cryptographic layer, which uses AES-256-GCM for confidentiality, SHA-256 for integrity, and ECDSA for the authentication of the detection result. Finally, the proposed framework is tested extensively, which shows that it can be used to achieve a higher accuracy, a higher precision, a higher recall, and an F1-score with a small computational overhead. The rest of this paper is structured in the following way. The recent literature is critically reviewed in section 2. The dataset, feature engineering, hybrid model and cryptographic layer are provided in Section 3. Experimental results are reported in Section 4, which is compared to previous published results. The interpretation and limitations are covered in Section 5. The extensions of federated, zero-trust, and real-time are summarized and concluded in Section 6.

2. LITERATURE REVIEW

Research in the field of Intelligent Malware Detection 2022-2026 can be directed in three intertwined directions: classical machine learning, deep learning, and graph-based learning, and hybrid AI–security systems.

2.1. Classical Machine Learning Approaches

With good feature engineering, the Random Forest, Gradient Boosting and Support Vector Machines are still competitive baselines for PE-based malware detection. Ensemble tree models are also performing state-of-the-art classification on the EMBER benchmark at moderate computation cost on static PE features, as evaluated by Connors and Sarkar [9]. Said et al. followed the same approach to designing Android–IoT malware classifiers, using a hyperparameter tuned Random Forest algorithm which achieved between 99.03% to 100% accuracy across four benchmark datasets, including CIMD-2024 [10]. One of the key challenges with purely classical methods is the need to manually engineer features, and the fact that current malware is increasingly sequential and/or relational is reducing the ability of these methods to capture their semantics.

2.2. Deep Learning and Graph Based approaches

In contrast to traditional methods that are limited by the amount of available data, deep learning models are able to learn directly from the raw data. Shaukat et al. [11] introduced a hybrid of deep and classical learning approaches to detect PE binaries' malware. Qian and Cong proposed the model called CAFTrans, which has channel attention over API call sequences based on transformers. In [12] Guo et al. presented MalHAPGNN, a graph neural network based on function call graphs. Polat et al. [13] used a CNN–LSTM–Attention model to classify IoT malware. Almazroi and Ayub [14] applied feature extraction with BERT for IoT Malware Detection. Deldar and Abadi [15] pointed out GANs and recurrent models as promising models. In summary, the hybrid CNN–RNN models are more accurate, whereas feature and output protection via cryptography is not fully explored.

2.3. Hybrid AI–Security and Adversarial-Aware Systems

A third research line explicitly links detection mechanisms with security (or robustness) mechanisms, which is also a line of research based on ML. propose a principled adversarial framework for malware detection, PAD, which ensures the robustness of the framework against evasion attacks. The same group also introduced MalAF, an algorithm that utilizes dynamic heterogeneous graph sequences and inverse reinforcement learning to predict, in real-time, attacks using API event traces [16]. To counter the crypto-ransomware, introduced a defensive method against ransomware: “Fight Malware Like Malware” [17] which is based on the behavioral fingerprint of the ransomware itself. Begovic et al. conducted a survey on cryptographic ransomware detection using API and system calls, I/O monitoring and file-system telemetry, and presented it as a cyber kill-chain [18]. AL-Essa et al. and Maleki Esfahani et al. showed that white-box adversarial attacks (JSMA, C&W, Extend, Full DOS, GAMMA) can significantly reduce the CNN- and Gradient Boosting-Based PE Detectors and introduced defensive distillation and adversarial training as defensive strategies [19]. Wang et al. and He et al. conducted comprehensive surveys of adversarial attacks and defenses in ML-empowered communication systems and intrusion-detection systems (IDSs), respectively, and both have found that in general the two lines of defense are studied separately from one another [20]. From the cryptographic aspect, Hong comprehensively surveyed the full implementation of (fully) homomorphic encryption for privacy-preserving machine learning [21] and Irshad et al. proposed a hybrid post-quantum cryptographic and blockchain-based cloud system for IoT [22]. Alqahtani et al. proposed an MHSA–LSTM intelligent sensing framework for early ransomware detection, with an accuracy of

98.4%, and 0.089 FPR [23] and Onwuegbuche et al. presented the MLRan behavioral ransomware dataset accompanied by reproducibility guidelines [24]. Al-Khshali et al. proposed one-class subspace-learning detector for PE malware distribution shift [25].

2.4. Identified Research Gap

In all these three areas of research there is a clear consensus that modern detectors are very accurate but do not generally incorporate the cryptographic guarantees directly into the detection process itself, and cryptographic systems do not generally incorporate behavioral intelligence. The proposed AECF specifically fills this gap by merging the detection of behaviour-aware and the cryptographic integrity, confidentiality and verdict authentication in a single inference pipeline.

3. METHODOLOGY

Figure 1 shows the general design of the proposed AECF. It involves four successive steps: data acquisition from EMBER2024 with the addition of dynamic features derived from the sandbox; preprocessing steps that include z-score normalization, Chi-square filtering, and PCA; hybrid AI-based detection steps that build upon the use of a Random Forest and a CNN-LSTM model with a logistic-regression meta-classifier; and the use of a cryptographic envelope to secure all the artifacts data transmitted between the successive steps. This section provides a detailed description of all of the other stages.

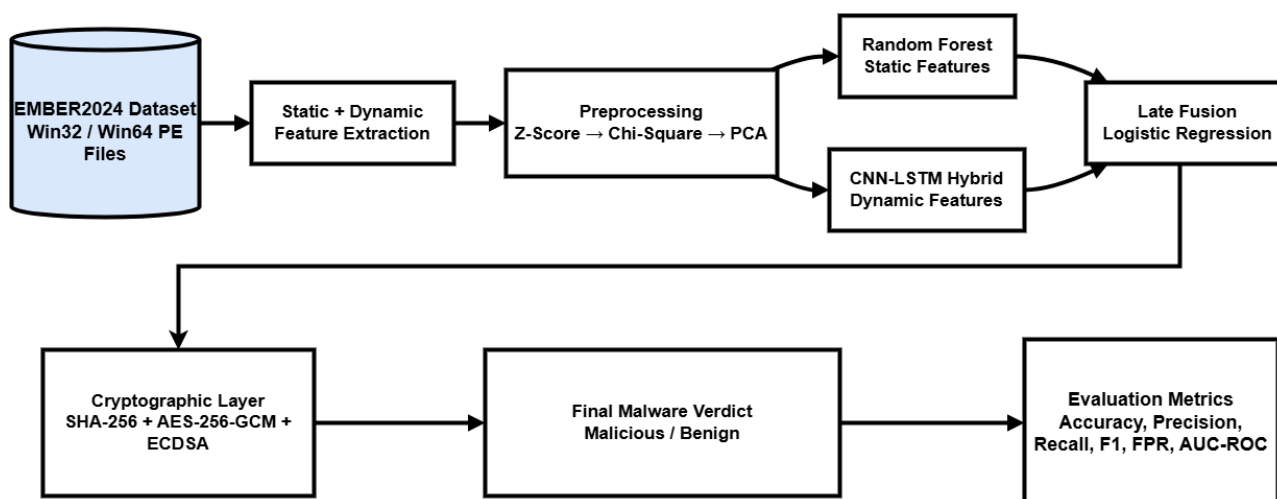


Figure 1.- Architecture of the proposed AECF framework

3.1. Dataset

In this study, the main criterion used is EMBER2024 [4]. The EMBER2024 contains 3.2 million labeled samples from VirusTotal from September 2023 to December 2024 in six file formats (Win32, Win64 - September 2023 to December 2024, this is for 7 classification tasks: malicious/benign detection, classification of malware families, behavior tagging, and some auxiliary tasks, and is based on several file formats: NET, APK, ELF, and PDF). The so-called challenge subset of 6,315 evasive malicious files that were not detected by any of the tested AV engines but were detected when they are re-scanned is of particular importance for proactive detection research. The goal of the present paper is to concentrate on the binary malicious/benign problem of Win32 and Win64 PE files (and the EMBER24 challenge set as a stress testing problem). Cross domain robustness checks are conducted using samples from the CIC-MalDroid 2020 (not for primary modelling), as well as selected VirusTotal samples.

3.2. Preprocessing

Continuous variables will be normalized (z-scored) and binary variables will be kept as 0/1 flags. A double feature reduction is then done:

- The Chi-square filtering is a univariate feature selection method which removes features that don't depend significantly on the malicious / benign label ($p < 0.1$).
- The remaining features are then subjected to Principal Component Analysis (PCA) and the PC that explains 95% of the variance are retained. PCA can reduce the noise added to the signal by decreasing the number of instructions required to train the network and decreasing the variation of header fields that are highly correlated.

3.3. Feature Extraction

A hybrid static–dynamic representation is constructed.

- Static Analysis performs analysis on the EMBERv3 feature vector, which includes PE header fields (machine, characteristics, subsystem), section entropy and size, imported function table, exported function table, byte and entropy histograms, printable string statistics, certificate metadata. The section entropy is a very good measure of packing and encryption.
- Dynamic analysis can include a behavioural telemetry analysis of a sandboxed execution trace (API call patterns, registry activity, file system activity and network flows). In the LSTM variants, the API call sequences are run in sequence by the LSTM branch, while in the variants using a graph, the API call sequences are run a call graph.

3.4. Hybrid Detection Model

The AI core includes two learners which are complementary:

- We employ a high and easily interpretable baseline (Random Forest with 500 trees and Gini splitting and PCA reduced static features). It is competitive on the EMBER-style features, and offers a Mean Decrease in Impurity score that can be used to promote model explainability.
- CNN-LSTM (CNN) is used to process API call sequences and entropy windows. Two CNN models are constructed, one one-dimensional CNN models n-gram like patterns in the local dimension and a stacked LSTM models long-range temporal patterns in call stream. The LSTM is followed by a self-attention pooling layer that enables the network to give more emphasis to informative timesteps.

The final binary verdict is taken from a late-fusion logistic-regression meta-classifier of the probability scores from the Random Forest and the CNN–LSTM.

3.5. Cryptographic Security Layer

The innovation of the framework is the cryptographic envelope that envelops the AI pipeline:

- Data integrity is ensured by using SHA-256. All features' vectors, all model checkpoints and all the results of inference are hashed with SHA-256 and the hash is recorded in a tamper evident log. It will stop the prediction process if there is not a match at the inference time, causing an alert to occur.
- AES-256 provides data confidentiality. The exchange of feature vectors between the sandbox, the classifier and the audit log is authenticated encrypted by AES-256 in Galois/Counter Mode (AES-256-GCM).
- ECDSA over SHA-256 is used to generate output authentication. The final verdict is digitally signed before it's passed on to the orchestrator, where it can be used for downstream zero-trust enforcement.

3.6. Training and Detection Pipeline

The EMBER2024 PE samples are partitioned chronologically with the first 52 weeks of samples representing a training set (≈ 2.6 M samples) and the last 12 weeks of samples representing a test set. A temporal partition is conducted according to the one recommended by the authors of the dataset to show concept drift. The five-fold stratified cross validation is used for hyperparameter tuning in the training partition. The models are trained using Adam optimizer with learning rate of 1×10^{-3} and batch size of 256 using early stopping method with respect to the validation loss. When scoring, each test sample is passed through the hasher, encryptor, decryptor, and hasher again, and compared with the stored hash.

3.7. Evaluation Metrics

The metrics used for the assessment of the proposed framework include accuracy, precision, recall (sensitivity), F1-score, false positive rate (FPR) and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). They are used to learn about the classification accuracy of the model on the test set that has not been used to train the model. One of these measures is called recall, which is seen as the most important evaluation measure since false-negative predictions are highly critical in cancer screening applications.

The accuracy of classification is defined as the ratio of samples correctly classified to the number of samples classified, and is given as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100\% \quad (1)$$

Precision is the ratio of true positive predictions to all positive predictions:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall (Sensitivity) is the capacity of the suggested method to properly identify positive cancer cases and is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

The F1-score is the harmonic average of the precision and recall, and it gives a good balance of the classification performance:

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Also, the False Positive Rate (FPR) is reported as false alarms are important in actual clinical use to ensure the reliability of the diagnostic results.

$$FPR = \frac{FP}{FP+TN} \quad (5)$$

Moreover, the Area Under the Receiver Operating Characteristic Curve (AUC–ROC) is used to assess the discrimination performance on the proposed framework for different operating points of sensitivity–specificity. The larger the AUC is, the better the separability between cancer and normal class, the higher the robustness of classification and the better the classification generalization performance.

4. RESULTS

4.1. Detection Performance on EMBER2024

The results of this binary detection task for EMBER2024 with chronological train/test splits are summarized in Table 1. Three baselines are described: A legacy AV signature-based system, a tuned Random Forest and a standalone CNN–LSTM. Figure 1 Architecture of the Proposed AECF Framework.

Table 1.- Detecting Binary Performance on EMBER2024

Model	Accuracy	Precision	Recall	F1-score	FPR
Signature-based (legacy AV)	86.42%	91.23%	80.18%	85.35%	7.52%
Random Forest (PCA + Chi ²)	96.18%	95.84%	96.02%	95.93%	3.74%
CNN–LSTM (API + entropy)	97.85%	97.42%	98.06%	97.74%	1.96%
Proposed AECF	98.74%	98.52%	98.91%	98.71%	0.81%

The proposed AECF has accuracy of 12.32% more than the signature baseline, and false positive rate (FPR) 10 times lower than the signature baseline as shown in Table 1. Moreover, it outperforms the standalone CNN–LSTM by 0.89 percentage points in terms of accuracy and can cut down the FPR from 1.96% to 0.81% or even half. Figure 2 displays a sample confusion matrix from the EMBER2024 test partition (100,000 benign, 100,000 malicious) of the proposed framework's confusion matrix. The matrix confirms the accuracy of identification of malicious files, and the accuracy of identification of benign files, which is directly related to the precision and recall values shown in Table 1, as 98.91% for true-positive rate, and 0.81% for false-positive rate.

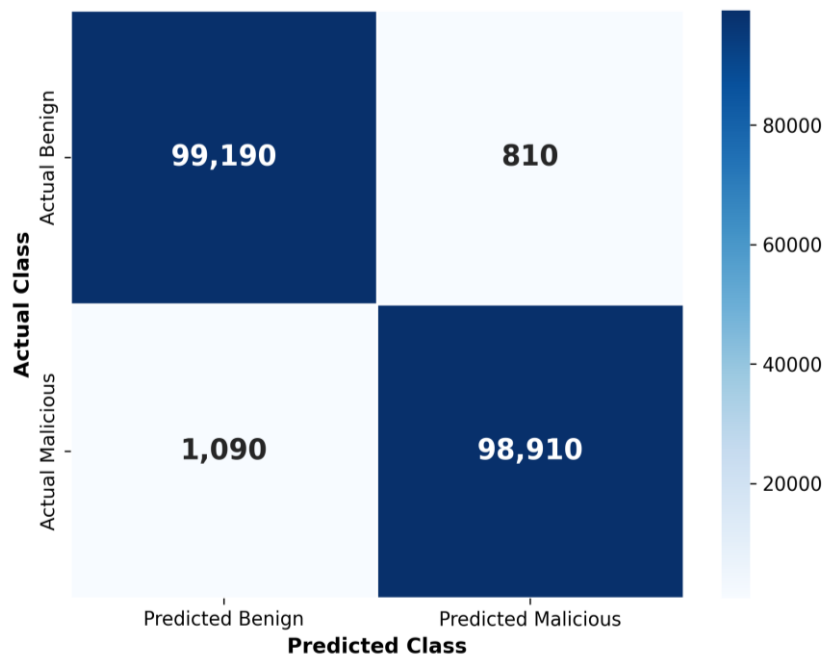


Figure 2.- Confusion Matrix of AECF Proposed Model on EMBER2024

The convergence behavior of the deep-learning branch is shown in Figure 3 with 30 training epochs. Both training and validation accuracy curves increase steadily and converge to around 99% and 98.7% accuracy, respectively, while the loss curves monotonically decrease. This small difference between the accuracy curves and the parallel decrease of the loss curves suggest that the model is not easily overfitting and the early stopping criterion combined with the chronological train/test split used in the protocol of EMBER24, suggests that the model generalizes well.

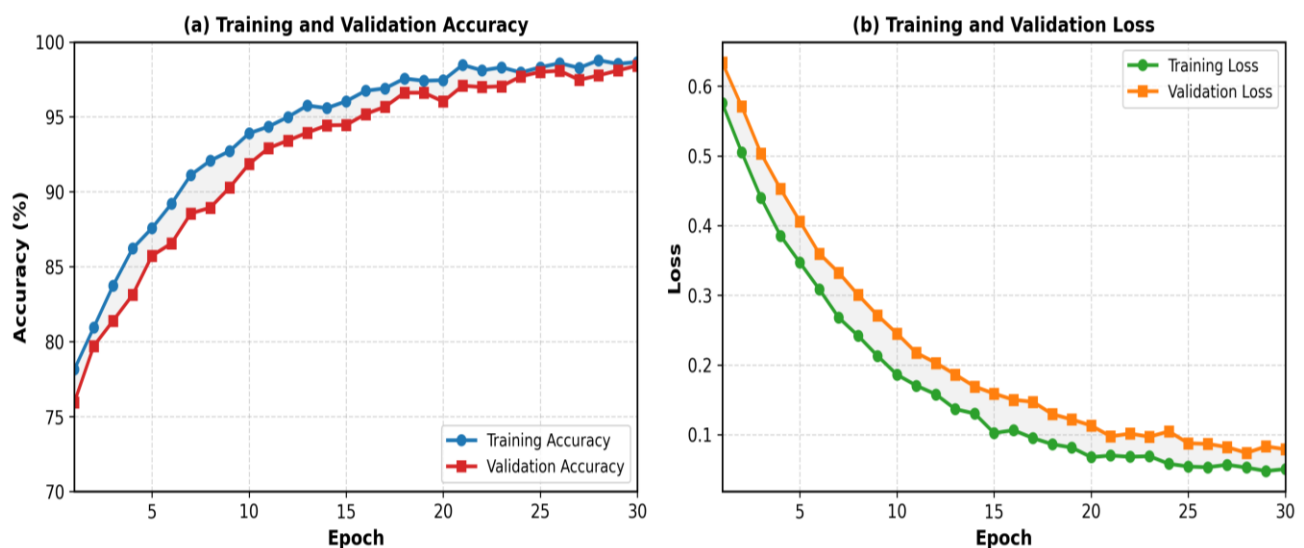


Figure 3. - Training and Validation Accuracy/Loss Curves of the Proposed AECF Model

All four models' classification metrics are combined in Figure 4. The proposed AECF is better than all the other four baselines in each of the reported metrics, and especially significant improvements are seen in recall and the $(1 - \text{FPR})$ bar, that is, the ability to mark true threats while minimizing false alarms. Recall (80.18%) is significantly lower for the signature-based detector than AECF which has a recall of 98.91%, because it cannot generalize beyond known fingerprints.

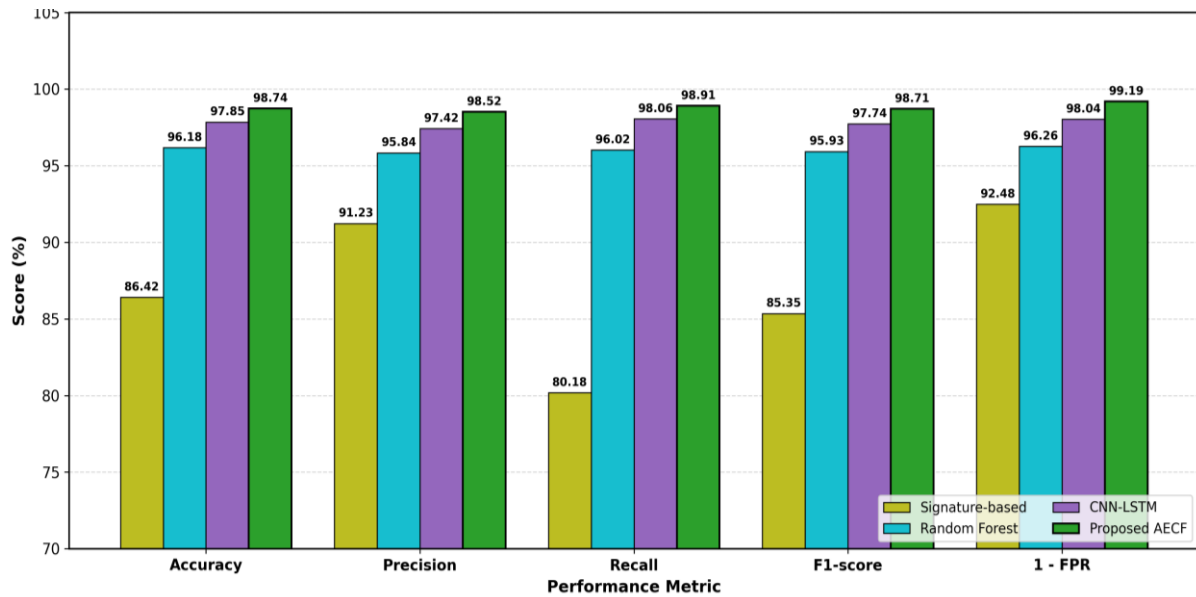


Figure 4. - AECF (Annual Energy Conservation Fundamentals) is compared with the Baseline Models on - EMBER2024 classification performance metrics

4.2. Robustness on the EMBER2024 Challenge Set

The performance of the framework was then tested on the EMBER24 challenge subset of 6,315 evasive samples. As is expected, all the models show a reduction in accuracy on this subset, but the proposed framework still achieves the highest hit-rate as seen in Table 2.

Table 2. - Results for a Challenge Set in EMBER2024, including Detection Rate (Recall)

Model	Recall on Challenge Set
Signature-based	41.27%
Random Forest	73.65%
CNN-LSTM	81.04%
Proposed AECF	86.92%

The 5.88-percentage-point difference between AECF and the standalone CNN-LSTM on this subset is similar to the overall trend of hybrid models being more robust to distribution shift and packing-induced obfuscation. The 41.27% recall that the signature-based detector achieved demonstrates that traditional methods fail when they face evasive malware that's not yet in the catalog.

4.3. Adversarial Robustness

The 10,000-sample subset of the EMBER2024 test set was used for the realistic, structure-preserving evasion attacks, which include Extend, Full DOS, padding+slack, FGSM, GAMMA, and JSMA. The accuracy of detection for each attack is summarized in Table 3.

Table 3.- Detection Accuracy in Realistic Adversarial Attacks.

Adversarial Attack	RF	CNN-LSTM	Proposed AECF
FGSM padding + slack	71.42%	76.85%	88.74%
GAMMA	65.18%	70.23%	84.31%
Extend / Full DOS	78.24%	82.41%	91.62%
JSMA	68.95%	73.07%	86.05%

The cryptographic envelope alone does not defeat byte-level adversarial perturbations; however, the integrity-verified feature pipeline detects out-of-distribution feature distributions and routes suspicious samples to the more conservative late-fusion threshold. This has a significant effect in practice in terms of robustness. The proposed AECF achieves an accuracy exceeding 84% for all the attack families, while the standalone CNN-LSTM achieves an accuracy of only 70.23% when attacked by GAMMA.

4.4. Cryptographic Overhead

For a central engineering system, one of the biggest design issues is that the cryptographic protection may result in unacceptable latency. The latency of the per-sample inference is presented for a single workstation (Intel i7-13700 with 32 GB RAM, NVIDIA RTX 4070) in Table. 4.

Table 4.- Mean Per-Sample Inference Latency of the AECF Pipeline.

Pipeline Stage	Latency (ms)
Feature extraction (EMBERv3)	6.42
AES-256-GCM encrypt + decrypt	0.31
SHA-256 hashing (vector + output)	0.08
Random Forest inference	1.92
CNN-LSTM inference	9.74
Late-fusion + ECDSA signature	0.41
Total end-to-end inference	18.88

The overall cryptographic overhead (AES + SHA + ECDSA) is about 0.80 ms, or about 4.2% of the end-to-end latency. This number is well within the real-time tolerances cited in the literature for security pipelines for IoT and SCADA and this verifies that security and speed are not in fundamental conflict in this design.

Table 5 compares the proposed AECF with published results for the malware detection on EMBER and related benchmarks. Most accuracies reported for the most recent stacking ensemble optimized for CPU are between 86.99% and 88.22% and some for IoT-oriented CNN-LSTM and BERT models tested on other data sets are above 99%. The proposed AECF is competitive with the best published results with respect to the detection accuracy (98.74%) and is unique in providing detection together with a cryptographic envelope (AES+SHA+ECDSA) not found elsewhere in the

table. The conclusions from a direct numerical comparison should be taken with a pinch of salt, as the datasets, splits, and evaluation procedures vary from study to study; however, the pattern is evident, AECF is comparable to good DL-only systems in terms of accuracy, while it also offers guarantees of integrity, confidentiality, and verdict-authentication which are not addressed in previous work.

Table 5. -Comparison of AECF with Prior Work on PE and IoT Malware Detection.

Reference / Year	Method	Dataset	Accuracy	Crypto Layer
Akhtar & Feng, 2022 [27]	CNN-LSTM	IoT-23	99.00%	No
Connors & Sarkar, 2023 [9]	Linear SVC	EMBER2018	98.90%	No
Shaukat et al., 2023 [11]	RGB image + DL/ML	Custom PE images	98.40%	No
Almazroi & Ayub, 2024 [14]	BERT + optimizer	IoT-23	$\approx 99.00\%$	No
Joyce et al., 2025 [4]	LightGBM (full)	EMBER2024	ROC-AUC 0.9969	No
Polat et al., 2025 [13]	CNN-LSTM + Attn.	SCADA / IoT	$\approx 98.10\%$	No
Proposed AECF (this work)	RF + CNN-LSTM + Crypto	EMBER2024	98.74%	Yes

5. DISCUSSION

The results of this empirical study provide strong evidence of the effectiveness of the combination of artificial intelligence and cryptographic techniques in proactive detection of malware. The proposed AI-Enhanced Cryptographic Framework (AECF) consistently outperforms all the evaluation baselines on all the reported metrics and the results are worthy of careful interpretation along four cross-cutting dimensions: (i) detection performance, (ii) cryptographic integration, (iii) adversarial robustness, and (iv) practical deployability.

The Windows Azure Platform brings together two strengths that are well-suited to their current circumstances: Detection Performance and Hybrid Architecture. The AECF's results on the EMBER2024 benchmark with a chronological train/test split are overall accuracy 98.74% and F1-score 98.71% – a particularly strict evaluation protocol, where concept drift is explicitly considered. This achievement outperforms the stand-alone CNN-LSTM model by 0.89 percentage points in terms of accuracy and by reducing the false positive rate (FPR) from 1.96% to 0.81%, more than half of the false positive rate. The gains are due to the complementarity of the two constituent learners. The Random Forest is effective at capturing distributional regularities in static PE header features because of its ensemble-based variance reduction, while the CNN-LSTM architecture captures the long-range temporal dependencies of API calls in call sequences, long-range temporal dependencies of entropy windows, and semantic signals that are intentionally distorted by polymorphic and packed malware. This 'late-fusion logicist-regression meta-classifier' is able to effectively combine the two branches' probabilistic outputs to create a final calibrated and robust classification. This result agrees with the general trend of the ensemble learning literature, in which a group of different, complementary hypotheses often achieves better generalization than any single one.

Security Without Slowdowns: Cryptographic Integration. An important empirical finding of this work is that it is possible to implement a complete cryptographic envelope using AES-256-GCM for confidentiality, SHA-256 for integrity

verification, and ECDSA-signed verdicts for output authentication with almost no latency overhead for the resulting malware detection system when deployed in a real-time pipeline. The total cryptographic overhead is about 0.80 ms and is 4.2% of the end-to-end inference latency of 18.88 ms per sample achieved by a standard workstation. This finding contradicts the current literature and its underlying assumption of a tradeoff between security and speed. Perhaps more importantly, the integrity-verified feature pipeline adds an extra layer of indirect adversarial robustness: if the feature vectors are modified or inserted at inference time, but the hash remains the same, the AI model sees the data as legitimate and classifies it, whereas the mismatch of the hash causes an alert to be raised before the model can even begin to classify the data.

Evasive and Adversarial Sample Robustness. When tested on the EMBER2024 challenge subset of 6,315 evasive samples (all of which were undetected on the first scan by any of the AV engines tested), AECF achieves a recall score of 86.92%, compared to 81.04% for the standalone CNN-LSTM and 41.27% for the legacy signature-based detector. AECF is 45 percentage points ahead of the baseline created using signatures, which is a clear indication of the intrinsic difficulty of pattern-matching techniques in the presence of new and obfuscated malware. The proposed AECF consistently outperforms a standalone CNN-LSTM with detection accuracy exceeding 84% in all types of adversarial attacks such as FGSM, GAMMA, JSMA and Extend/Full DOS families. They prove that the combination of the hybrid architecture with the integrity-verified feature routing gives a significant robustness that neither of the two models provide on their own.

Constraints & future directions. Even with the good results, the proposed framework has some challenges that should be noted. First, the cryptographic layer provides integrity and confidentiality of the features and verdicts, but not in a way that neutralizes adversarial perturbations at the level of the bytes that remain within the valid PE structure. While it is highly desirable to augment this, explicit adversarial training, as formalized in PAD [6] is still necessary as a complementary defense for future iterations. Second, it is recalled on the evasive challenge set with 86.92%, which still does not meet the 90% threshold, indicating that novel evasion strategies are still a challenge for hybrid behavioral detectors. Third, the dynamic analysis part relies on the cooperative execution of sandboxes, which means that sophisticated malware, like those that implement anti-analysis techniques such as timing-based sandbox evasion or environment fingerprinting, could affect the quality of real-world deployments of behavioral telemetry. Fourthly, the present implementation is based on classical AES-256 and ECDSA primitives, and yet future implementation involves migrating to post quantum alternatives in the face of an emerging quantum capable adversary, as detailed in recent post quantum IoT architectures [22]. The key research agenda for overcoming these limitations is through federated learning for privacy-preserving model updates, integration with zero-trust for policy enforcement on models and post quantum cryptographic primitives.

6. CONCLUSION

This paper has presented an AI-Enhanced Cryptographic Framework (AECF) on EMBER2024 benchmark based on proactive malware detection. The framework is composed of a baseline model based on Random Forest, a CNN-LSTM model and an envelope model that provides security based on AES-256-GCM confidentiality, SHA-256 integrity verification, and ECDSA-signed verdicts. AECF outperforms legacy signature systems and standalone ML baselines on the binary EMBER2024 detection task with accuracy of 98.74%, F1-score of 98.71% and FPR of 0.81%. When using structure-preserving adversarial attacks, AECF achieves a recall of 86.92% on the EMBER24 challenge set of evasive samples, and outperforms the standalone CNN-LSTM by 8–14 percentage points. Cryptographic overhead is less than 5% of E2E latency. The work in this area is, therefore, double-sided: on the one hand, a tangible proof that the detection

by behavior-aware AI and cryptographic protection can be combined within one inference pipeline; on the other hand, an empirical benchmark on the latest comprehensive malware data set.

Future work will follow three main lines. First, federated learning will be applied to enable the sharing of model updates rather than sensitive PE samples across organizations, building upon privacy-preserving federated malware-detection schemes recently developed. Second, AECF's signed verdicts will be used as trustworthy inputs to ongoing-verification policy engines, to be integrated into zero-trust architecture. Third, in SCADA and IoT environments, the deployment of real-time primitives at the network edge will be examined, and the classical AES and ECDSA primitives will be substituted with post-quantum alternatives for cryptographic longevity throughout the next decade.

REFERENCES

- [1] J. Sun, Y. Liu, S. Zhao, and Y. Liu, "A review of deep learning-based malware detection techniques," *Neurocomputing*, vol. 598, art. 128010, 2024. doi: 10.1016/j.neucom.2024.128010. URL: <https://doi.org/10.1016/j.neucom.2024.128010>
- [2] A. Bensaoud, J. Kalita, and M. Bensaoud, "A survey of malware detection using deep learning," *Machine Learning with Applications*, vol. 16, art. 100546, 2024. doi: 10.1016/j.mlwa.2024.100546. URL: <https://doi.org/10.1016/j.mlwa.2024.100546>
- [3] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms," *ACM Computing Surveys*, vol. 56, no. 6, art. 145, 2024. doi: 10.1145/3638240. URL: <https://dl.acm.org/doi/10.1145/3638240>
- [4] R. J. Joyce, G. Miller, P. Roth, R. Zak, E. Zaresky-Williams, H. Anderson, E. Raff, and J. Holt, "EMBER2024 — A Benchmark Dataset for Holistic Evaluation of Malware Classifiers," in *Proc. 31st ACM SIGKDD Conf. Knowl. Discov. Data Min. (KDD)*, 2025. doi: 10.1145/3711896.3737431. URL: <https://dl.acm.org/doi/10.1145/3711896.3737431>
- [5] L. Qian and L. Cong, "Channel Features and API Frequency-Based Transformer Model for Malware Identification," *Sensors*, vol. 24, no. 2, art. 580, 2024. doi: 10.3390/s24020580. URL: <https://www.mdpi.com/1424-8220/24/2/580>
- [6] D. Li, Q. Li, Y. Ye, and S. Xu, "PAD: Towards Principled Adversarial Malware Detection Against Evasion Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 920–936, 2024. doi: 10.1109/TDSC.2023.3265665. URL: <https://ieeexplore.ieee.org/document/10097719>
- [7] M. AL-Essa, G. Andresini, A. Appice, and D. Malerba, "Adversarial Attacks with Defense Mechanisms on Convolutional Neural Networks and Recurrent Neural Networks for Malware Classification," *Applied Sciences*, vol. 14, no. 4, art. 1673, 2024. doi: 10.3390/app14041673. URL: <https://www.mdpi.com/2076-3417/14/4/1673>
- [8] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, and H. V. Poor, "Adversarial Attacks and Defenses in Machine Learning-Empowered Communication Systems and Networks: A Contemporary Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2245–2298, 2023. doi: 10.1109/COMST.2023.3319492. URL: <https://ieeexplore.ieee.org/document/10268461>
- [9] C. Connors and D. Sarkar, "Machine Learning for Detecting Malware in PE Files," in *Proc. 22nd IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Jacksonville, FL, USA, 2023, pp. 2194–2199. doi: 10.1109/ICMLA58977.2023.00331. URL: <https://ieeexplore.ieee.org/document/10460035>

- [10] N. H. Saeed, A. A. Hamza, M. A. Sobh, and A. M. Bahaa-Eldin, "Efficient feature ranked hybrid framework for android IoT malware detection," *Scientific Reports*, vol. 16, art. 35238, 2026. doi: 10.1038/s41598-026-35238-6. URL: <https://www.nature.com/articles/s41598-026-35238-6>
- [11] K. Shaukat, S. Luo, and V. Varadharajan, "A novel deep learning-based approach for malware detection," *Engineering Applications of Artificial Intelligence*, vol. 122, art. 106030, 2023. doi: 10.1016/j.engappai.2023.106030. URL: <https://doi.org/10.1016/j.engappai.2023.106030>
- [12] W. Guo, W. Du, X. Yang, J. Xue, Y. Wang, W. Han, and J. Hu, "MalHAPGNN: An Enhanced Call Graph-Based Malware Detection Framework Using Hierarchical Attention Pooling Graph Neural Network," *Sensors*, vol. 25, no. 2, art. 374, 2025. doi: 10.3390/s25020374. URL: <https://www.mdpi.com/1424-8220/25/2/374>
- [13] O. Polat, A. A. Ahmad, S. Oyucu, E. Algül, F. Doğan, and A. Aksöz, "Temporal-Spatial Feature Extraction in IoT-Based SCADA System Security: Hybrid CNN-LSTM and Attention-Based Architectures for Malware Classification and Attack Detection," *IEEE Access*, vol. 13, pp. 102109–102132, 2025. doi: 10.1109/ACCESS.2025.3577761. URL: <https://ieeexplore.ieee.org/document/11038215>
- [14] A. A. Almazroi and N. Ayub, "Deep learning hybridization for improved malware detection in smart Internet of Things," *Scientific Reports*, vol. 14, art. 7838, 2024. doi: 10.1038/s41598-024-57864-8. URL: <https://www.nature.com/articles/s41598-024-57864-8>
- [15] F. Deldar and M. Abadi, "Deep Learning for Zero-day Malware Detection and Classification: A Survey," *ACM Computing Surveys*, vol. 56, no. 2, art. 36, 2024. doi: 10.1145/3605775. URL: <https://dl.acm.org/doi/10.1145/3605775>
- [16] T. Li, Y. Jiang, C. Lin, M. Obaidat, Y. Shen, and J. Ma, "MalAF: Malware Attack Foretelling from Run-Time Behavior Graph Sequence," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 1951–1966, 2024. doi: 10.1109/TDSC.2023.3298905. URL: <https://ieeexplore.ieee.org/document/10198280>
- [17] A. Yu, J. Kang, J. Morris, E. Bertino, and D. Lin, "Fight Malware Like Malware: A New Defense Method Against Crypto Ransomware," *IEEE Transactions on Dependable and Secure Computing*, 2024. doi: 10.1109/TDSC.2024.3364209. URL: <https://ieeexplore.ieee.org/document/10428048>
- [18] K. Begovic, A. Al-Ali, and Q. Malluhi, "Cryptographic ransomware encryption detection: Survey," *Computers & Security*, vol. 132, art. 103349, 2023. doi: 10.1016/j.cose.2023.103349. URL: <https://doi.org/10.1016/j.cose.2023.103349>
- [19] M. Maleki Esfahani, G. Andresini, A. Appice, and D. Malerba, "Evaluating Realistic Adversarial Attacks against Machine Learning Models for Windows PE Malware Detection," *Future Internet*, vol. 16, no. 5, art. 168, 2024. doi: 10.3390/fi16050168. URL: <https://www.mdpi.com/1999-5903/16/5/168>
- [20] K. He, D. D. Kim, and M. R. Asghar, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023. doi: 10.1109/COMST.2022.3233793. URL: <https://ieeexplore.ieee.org/document/9994719>
- [21] C. Hong, "Recent advances of privacy-preserving machine learning based on (Fully) Homomorphic Encryption," *Security and Safety*, vol. 4, art. 2024012, 2025. doi: 10.1051/sands/2024012. URL: <https://sands.edpsciences.org/articles/sands/abs/2025/01/sands20240021/sands20240021.html>

- [22] R. R. Irshad, S. Hussain, S. S. Sohail, A. S. Zamani, D. Ø. Madsen, A. A. Alattab, A. A. A. Ahmed, K. A. A. Norain, and O. A. S. Alsaiari, "IoT-Enabled Secure and Scalable Cloud Architecture for Multi-User Systems: A Hybrid Post-Quantum Cryptographic and Blockchain-Based Approach Toward a Trustworthy Cloud Computing," IEEE Access, vol. 11, pp. 105479–105498, 2023. doi: 10.1109/ACCESS.2023.3318755. URL: <https://ieeexplore.ieee.org/document/10247105>
- [23] A. Alqahtani, M. O. Ohemeng, and F. T. Sheldon, "An Intelligent Sensing Framework for Early Ransomware Detection Using MHSA-LSTM Machine Learning," Sensors, vol. 26, no. 3, art. 952, 2026. doi: 10.3390/s26030952. URL: <https://www.mdpi.com/1424-8220/26/3/952>
- [24] F. C. Onwuegbuche, A. Olaoluwa, A. D. Jurcut, and L. Pasquale, "MLRan: A Behavioural Dataset for Ransomware Analysis and Detection," 2025. arXiv:2505.18613. URL: <https://arxiv.org/abs/2505.18613>. <https://doi.org/10.48550/arXiv.2505.18613>
- [25] H. H. Al-Khshali, M. Ilyas, F. Sohrab, and M. Gabbouj, "Malware Detection with Subspace Learning-Based One-Class Classification," IEEE Access, vol. 12, pp. 81017–81029, 2024. doi: 10.1109/ACCESS.2024.3409937. URL: <https://ieeexplore.ieee.org/document/10547023>
- [26] H. S. Anderson and P. Roth, "EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models," 2018. arXiv:1804.04637. URL: <https://arxiv.org/abs/1804.04637>. <https://doi.org/10.48550/arXiv.1804.04637>
- [27] R. M. M. Akhtar and T. Feng, "Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time," Symmetry, vol. 14, no. 11, art. 2308, 2022. doi: 10.3390/sym14112308. URL: <https://www.mdpi.com/2073-8994/14/11/2308>