SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

CROSS-MODAL DEEP LEARNING FOR MALWARE DETECTION AND ZERO-DAY THREAT IDENTIFICATION USING BEHAVIORAL AND MEMORY FEATURES

Dr Nirali Arora[1], Dr. Tejashri Kolhe[2], Dr Siddharth Hariharan[3], Yogeshwari Hardas[4], Poonam Tiware[5], Kanchan Girish Wankhede[6], Priyanka Jeetendra Patil[7]

[1]CSE-AIML, Computer engineering, Ap shah institute of technology Mumbai. E-mail: nrarora@apsit.edu.in
[2]Email: ttkolhe@apsit.edu.in, Assistant professor, A P Shah Institute of Technology, Thane, CSE- AIM.L
[3]Associate Professor, Computer engineering, Pillai University, Navi Mumbai. E-mail: drsiddharth.shk@gmail.com
[4]Assistant professor, CSE (AI & ML), A P Shah Institute of Technology, Thane. E-mail: ydhardas@apsit.edu.in
[5]Email: pntiware@apsit.edu.in, Assistant professor, A P Shah Institute of Technology, Thane, Department: CSE- AIML.
[6]Assistant professor, CSE-AIML, A P shah Institute of Technology, Thane (Mumbai). E-mail: kgwankhede@apsit.edu.in
[7]Assistant Professor, CSE-AIML, A.P.Shah Institute of Technology, Thane. E-mail- pjpatil@apsit.edu.in

ABSTRACT

Zero-day malware poses a significant and continuously evolving cybersecurity threat because it exploits previously unknown vulnerabilities and employs sophisticated evasion techniques that can bypass conventional signature-based antivirus and rule-driven detection mechanisms. The inability of traditional approaches to identify previously unseen malware variants highlights the need for intelligent detection frameworks capable of learning complex behavioral and memory-level characteristics without relying exclusively on known signatures. To address this challenge, this study proposes a cross-modal hybrid deep learning framework that integrates dynamic behavioral analysis with volatile memory forensics for comprehensive malware detection, classification, and zero-day threat identification. The proposed framework consists of two complementary deep learning branches designed to capture heterogeneous characteristics of malicious software. The first branch employs a Bidirectional Long Short-Term Memory (BiLSTM) network enhanced with a self-attention mechanism to model temporal dependencies and identify discriminative patterns within API-call and system-call sequences collected during sandboxed malware execution. This branch enables the framework to capture behavioral characteristics associated with malicious activities, execution patterns, and system interactions. The second branch integrates a one-dimensional Convolutional Neural Network (1D-CNN) with a stacked autoencoder to learn representative patterns from volatile memory-resident features. While the CNN extracts localized and discriminative feature representations, the stacked autoencoder provides compact representations and reconstruction-based anomaly information that can assist in identifying samples exhibiting characteristics that deviate from previously observed malware families.

To effectively combine these heterogeneous representations, a cross-modal attention mechanism is introduced to learn the relative importance of behavioral and memory-based features and generate a unified representation for malware-family classification. In parallel, the reconstruction error generated by the autoencoder is utilized as an additional anomaly indicator for identifying potentially unseen malware samples and zero-day threats. The proposed framework was evaluated using the CIC-MalMem-2022 dataset in conjunction with a curated dynamic-behavior corpus. To provide a realistic evaluation of zero-day detection capability, a held-out malware-family protocol was adopted, ensuring that selected malware families were excluded from the training process and subsequently used to evaluate the model's ability to generalize to previously unseen threats. Experimental results demonstrate that the proposed framework achieves an accuracy of 99.3%, an F1-score of 0.986, and a zero-day detection AUC of 0.981. Furthermore, comparative experiments show that the proposed approach consistently outperforms conventional machine learning and deep learning baselines, including Random Forest, XGBoost, LightGBM, CNN-BiLSTM, and MalHyStack. The results indicate that combining dynamic behavioral evidence with memory-forensic information provides complementary information that improves the robustness and generalization of malware detection. The incorporation of cross-modal attention further enables effective fusion of heterogeneous security features, while reconstruction-based anomaly detection provides an additional capability for identifying previously unseen malware. Overall, the proposed framework demonstrates the potential of multimodal deep learning and forensic feature fusion as a proactive approach for detecting emerging and zero-day malware threats in modern cybersecurity environments.

Keywords: Zero-Day, Malware Detection, Deep Learning, Memory Forensics, Behavioral Analysis, BiLSTM, Cross-Modal Attention, Novelty Detection

Introduction

Volatile memory forensics captures a snapshot of a system's RAM, either from a live suspected-compromised host or from a sandboxed detonation, and extracts structural artifacts such as process lists, loaded dynamic-link libraries (DLLs), open handles, injected code regions, and kernel-level modifications, which are especially valuable for detecting fileless and in-memory malware that leaves little or no trace on disk. Each perspective captures information the other does not: behavioral traces capture what a process does over time, while

memory forensics captures the structural consequences of that behavior at the operating-system level, including artifacts of process hollowing, DLL injection, and other stealth techniques that may not be fully visible in an API-call log alone.

Despite the complementary nature of these two modalities, the majority of existing deep learning-based malware detection systems specialize in only one. Behavioral sequence models built on recurrent architectures achieve strong results on API-call classification but are comparatively blind to memory-resident evidence of process injection or credential-dumping activity. Conversely, memory-forensics classifiers built on tree ensembles or convolutional networks over statistical memory features achieve very high accuracy on datasets such as CIC-MalMem-2022 but do not exploit the temporal dynamics of program execution that behavioral traces expose. This paper is motivated by the hypothesis that fusing both modalities within a single trainable architecture, rather than treating them as separate, independently deployed detectors, yields not only higher overall classification accuracy but, more importantly, more reliable generalization to genuinely novel, zero-day malware families that were never observed during training.

The proposed framework employs a hybrid deep neural architecture that jointly analyzes two complementary sources of malware evidence: dynamic behavioral information and volatile memory-forensic characteristics. The behavioral branch is designed to process sequential information obtained during malware execution, such as API-call and system-call sequences. A Bidirectional Long Short-Term Memory (BiLSTM) network is employed to capture both forward and backward temporal dependencies within these sequences, enabling the model to learn complex execution patterns that may distinguish benign, known-malware, and potentially malicious activities. To further enhance the representation of the most informative behavioral events, a self-attention mechanism is incorporated into the BiLSTM branch. Rather than treating all API or system calls with equal importance, the attention mechanism assigns higher weights to sequence elements that contribute more significantly to malware identification, thereby producing a more discriminative behavioral representation.

In parallel, the memory-forensics branch processes features extracted from volatile memory, which can contain valuable evidence of malicious activity that may not be apparent from execution traces alone. A one-dimensional Convolutional Neural Network (1D-CNN) is utilized to identify local and hierarchical patterns within the memory-based feature representation. The convolutional layers enable the model to automatically learn discriminative feature patterns without requiring extensive manual feature engineering. This branch is further coupled with a stacked autoencoder, which learns a compact representation of the underlying distribution of memory-forensic features. In addition to feature extraction, the autoencoder plays an important role in identifying deviations from the learned distribution of known malware families.

The representations generated by the behavioral and memory-forensic branches are subsequently integrated through a cross-modal attention mechanism. This component enables the framework to learn relationships between the two modalities and dynamically determine the relative importance of behavioral and memory-based evidence for a given malware sample. Instead of simply concatenating the two feature vectors, cross-modal attention allows information from one modality to influence the interpretation of the other, resulting in a richer and more context-aware joint representation. This fused representation is then provided to a supervised malware-family classification head, which predicts the malware family associated with a sample based on patterns learned from the training data.

In addition to the supervised classification component, the framework incorporates an unsupervised novelty-detection head based on autoencoder reconstruction error. This component is particularly important for the zero-day malware detection scenario. The supervised classifier is inherently limited by the malware families available during training; therefore, when a previously unseen malware family is encountered, the classifier may assign it to the most similar known family even though the sample actually represents a novel threat. Such a prediction can result in a false sense of confidence because the classifier is forced to select from previously learned classes. The reconstruction-based novelty detector addresses this limitation by learning the underlying distribution of memory representations associated with known malware families during training. When a new sample is presented, its memory representation is passed through the trained autoencoder and reconstructed. Samples that closely resemble the learned distribution of known malware are expected to have relatively low reconstruction errors, whereas samples containing substantially different characteristics are expected to produce higher reconstruction errors.

Consequently, a high reconstruction error serves as an anomaly signal indicating that the sample may not belong to the distribution of malware families observed during training. This provides an additional layer of decision-making beyond the supervised classifier. In the zero-day setting, where the actual malware family has not been observed during model training, the novelty-detection head can flag the sample as potentially unseen even if the classification head incorrectly assigns it to a visually or behaviorally similar known family. Thus, the proposed architecture does not rely solely on closed-set classification; instead, it combines known-family classification with open-set novelty detection, enabling the system to distinguish between samples that can be confidently associated with known malware families and samples that exhibit characteristics indicative of previously unseen threats.

Overall, the architecture follows a dual-evidence and dual-decision strategy: the behavioral branch captures

how the malware behaves during execution, while the memory-forensic branch captures what traces and structural characteristics the malware leaves in volatile memory. Cross-modal attention integrates these complementary perspectives, the supervised classification head performs malware-family identification, and the reconstruction-error-based novelty detector provides an independent mechanism for identifying potential zero-day samples. This combination is particularly suitable for zero-day malware detection because it reduces dependence on predefined signatures and enables the model to exploit both discriminative patterns of known threats and deviations associated with emerging or previously unseen malware families.

Related Work

A. Behavioral (Dynamic) Malware Detection

Behavior-based malware detection executes suspicious binaries in instrumented environments and models the resulting sequence of system interactions [15]. API-call and system-call sequences capture the temporal ordering of operations that characterizes malicious intent, with recent work framing behavior-based detection explicitly as a sequential modeling problem to better capture multi-stage attack chains [17].

B. Memory-Forensics-Based Malware Detection

Ensemble and tree-based classifiers have achieved strong multi-class performance on this dataset after class-balancing techniques such as SMOTE and ADASYN and structured feature-selection pipelines, with Random Forest reported to achieve an F1-score of 0.939 in a representative recent study [13]. A deep-stacked ensemble architecture (MalHyStack) combining convolutional weak learners with a multilayer-perceptron meta-learner, validated with explainable artificial intelligence (XAI) techniques, has demonstrated strong cross-dataset performance including on CIC-MalMem-2022 specifically [12]. Alert-driven, memory-artifact-based frameworks combining interpretable models such as TabNet with ensemble classification have likewise been proposed as a two-stage host-based detection pipeline explicitly targeting stealthy, zero-day threats [2].

C. Hybrid and Multimodal Approaches, and Positioning of the Proposed Framework

A smaller body of work has begun to combine multiple malware analysis modalities. End-to-end multimodel deep learning architectures for malware classification have combined static, dynamic, and image-based representations of a single binary [16], while broader surveys of deep learning for zero-day malware detection and classification catalogue an expanding space of architectures [5], including heterogeneous dual-branch neural networks [6] and mixture-of-experts designs [7], that combine complementary feature views. Real-time hybrid deep learning systems have likewise been proposed for cloud-based malware detection [18], and dependable hybrid machine learning architectures have been explored in the adjacent domain of network intrusion detection [14]. However, the specific combination of dynamic behavioral sequence modeling with volatile memory-forensic structural features, fused through a learned cross-modal attention mechanism and coupled with an unsupervised novelty detection head purpose-built for the zero-day setting, has received comparatively little dedicated attention, and existing evaluations of memory-forensics classifiers are typically confined to standard train/test splits over known families rather than a held-out-family protocol that directly measures generalization to genuinely unseen malware.

The framework proposed in this paper is positioned to close these three gaps: (i) it jointly reasons over behavioral and memory-forensic evidence within a single trained model rather than as separate, independently deployed detectors; (ii) it dedicates an explicit unsupervised novelty-detection pathway to the zero-day problem, rather than relying solely on a closed-set classifier's confidence score, which is known to be an unreliable indicator of out-of-distribution novelty; and (iii) it is evaluated under a held-out-family protocol that more faithfully simulates the operational reality of encountering a malware family with no historical precedent.

Proposed Methodology

A. System Overview

The proposed framework, illustrated in Fig. 1, ingests two parallel evidence streams for a given suspicious executable or already-detonated sample: (i) a dynamic behavioral trace, consisting of the ordered sequence of API and system calls captured during sandboxed execution, and (ii) a volatile memory dump, from which a structured set of memory-resident statistical features is extracted following the CIC-MalMem-2022 feature-engineering methodology. Each stream is processed by a dedicated encoder: a BiLSTM with self-attention for the temporal behavioral stream, and a 1D-CNN coupled with a stacked autoencoder for the structural memory stream. The two resulting latent representations are combined by a cross-modal attention fusion layer, whose output feeds a dense softmax classification head for supervised family-level classification. In parallel, the memory-branch autoencoder's reconstruction error is used, without any additional supervision, to compute a novelty score that flags samples inconsistent with the learned distribution of known malware and benign software as candidate zero-day threats.

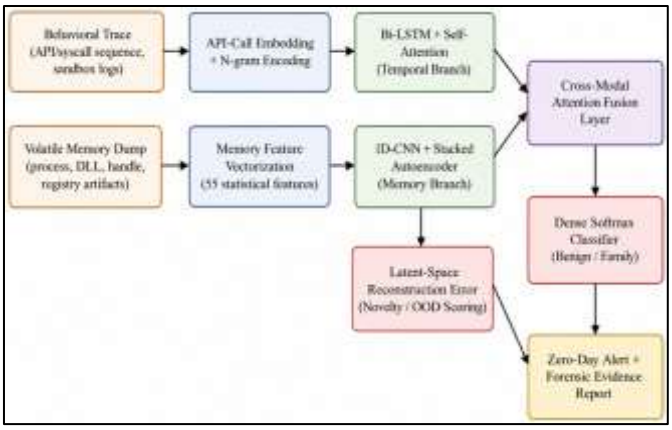

Fig. 1.Hybrid dual-branch architecture fusing behavioral sequence modeling and memory-forensic structural analysis for zero-day malware detection.

B. Behavioral Branch: BiLSTM with Self-Attention

Dynamic analysis traces are collected by executing each sample within an isolated sandbox for a fixed monitoring window, recording the ordered sequence of API and system calls. Each distinct API call is mapped to a dense embedding vector via a learned embedding layer, analogous to word embeddings in natural language processing, and consecutive calls are grouped into overlapping n-gram windows to capture short-range call patterns (e.g., a characteristic sequence of process-creation followed by registry-modification calls). The resulting embedded sequence is passed through a two-layer bidirectional LSTM with 192 hidden units per direction [19], allowing the model to relate an early-stage behavior, for example a dropper process writing a payload to disk, to a later-stage behavior, for example that payload subsequently establishing a command-and-control connection, regardless of the temporal distance between them within the trace. A self-attention layer over the BiLSTM output sequence then produces a single fixed-length behavioral embedding, weighted toward the API calls most indicative of malicious intent, mirroring the design rationale of recent attention-enhanced behavioral detectors [20].

C. Memory Branch: 1D-CNN and Stacked Autoencoder

Memory-forensic features, comprising 55 statistical descriptors of process memory, including counts and characteristics of loaded DLLs, handles, threads, injected code regions, and kernel callback structures, following the CIC-MalMem-2022 feature schema, are first passed through a 1D convolutional feature extractor with three convolutional blocks (32, 64, and 128 filters respectively), which learns local co-occurrence patterns among memory-forensic indicators, for example the joint appearance of an anomalous handle count and a hidden thread, characteristic of process-hollowing style injection. The resulting feature map is flattened and passed into a symmetric stacked autoencoder (encoder dimensions 128-64-32, mirrored in the decoder), trained to reconstruct its own input. The 32-dimensional bottleneck representation serves two purposes: it is passed forward, alongside the CNN feature map, into the cross-modal fusion layer described in Section III-D for supervised classification, and it independently supports the reconstruction-error-based novelty detection head described in Section III-E.

D. Cross-Modal Attention Fusion and Classification Head

The behavioral embedding produced in Section III-B and the memory embedding produced in Section III-C are combined via a cross-modal attention fusion layer, in which each modality's representation attends over the other, producing a joint embedding that emphasizes agreement and complementary evidence between the two streams rather than simply concatenating them. This joint embedding is passed to a fully connected softmax classification head that predicts either a benign label or one of the known malware family labels (Ransomware, Spyware, Trojan Horse, and their constituent subfamilies, following the CIC-MalMem-2022 taxonomy). The network is trained end-to-end using a categorical cross-entropy loss for the classification head jointly with the autoencoder's mean-squared-error reconstruction loss (Section III-E), combined through a weighted multi-task objective, together with L2 weight-decay regularization.

E. Unsupervised Novelty Detection for Zero-Day Identification

Because a supervised classifier can only assign probability mass to labels observed during training, it cannot, by construction, reliably flag a genuinely novel malware family; at best it will assign such a sample to whichever known family it superficially resembles, often with misleadingly high confidence. To address this, the memory-branch autoencoder's per-sample reconstruction error is monitored directly: because the autoencoder is trained only to reconstruct the distribution of memory representations seen during training (spanning benign software and known malware families), a sample drawn from a genuinely novel family, with a memory-artifact

profile not well represented in that training distribution, is expected to yield an anomalously elevated reconstruction error. A novelty threshold is set as the 97.5th percentile of reconstruction error observed on the validation split of known classes; at inference time, any sample whose reconstruction error exceeds this threshold is flagged as a candidate zero-day sample regardless of the softmax classifier's output, and is routed to a forensic evidence report (Section III-F) for analyst review rather than being silently assigned to an existing family label. For every sample flagged as malicious or as a zero-day candidate, the framework assembles a structured forensic report combining the top-attended API calls from the behavioral branch, the memory-forensic features contributing most to the elevated reconstruction error (via per-feature reconstruction residuals), and the model's calibrated confidence scores, intended to accelerate analyst triage and downstream incident response.

Experimental Setup

A. Datasets

The proposed framework is evaluated using two complementary data sources, summarized in Table I, together with a held-out-family protocol specifically designed to simulate zero-day conditions.

TABLE I. Summary of Datasets Used for Evaluation

Dataset / Source	Description	Role in Evaluation
CIC-MalMem-2022	58,596 memory dump samples	Memory-forensic branch training and evaluation
Curated dynamic-behavior corpus	Sandboxed execution traces (API/system-call sequences) collected for the same malware family set, following standard dynamic-analysis instrumentation practice.	Behavioral branch training and evaluation
Held-out-family zero-day protocol	One complete malware subfamily is entirely withheld from training and validation and used only at test time to simulate a genuinely unseen, zero-day threat.	Zero-day generalization / novelty-detection evaluation

For both CIC-MalMem-2022 and the behavioral corpus, standard 70/15/15 train/validation/test splits are used for in-distribution evaluation, with 5-fold cross-validation applied for robustness. For the zero-day evaluation, the experiment is repeated five times, each time withholding a different malware subfamily entirely from training, and reported zero-day metrics are averaged across these five held-out runs.

B. Implementation Details

TABLE II. Implementation Details and Hyperparameter Configuration

Component	Configuration
Behavioral encoder	2-layer Bidirectional LSTM, 192 hidden units/direction, self-attention (dim. 192)
Memory encoder	1D-CNN (32→64→128 filters) + stacked autoencoder (128-64-32-64-128)
Fusion	Cross-modal attention fusion layer (joint embedding dim. 256)

Component	Configuration
Novelty threshold	97.5th percentile of validation reconstruction error
Optimizer	Adam, initial LR = 5×10^{-4}, cosine annealing
Batch size	32
Epochs (max / early-stop patience)	50 / 8

C. Evaluation Metrics

In-distribution classification performance is reported using Accuracy, Precision, Recall, F1-score, and AUC across the benign/malware-family classification task. Zero-day generalization performance is reported separately using AUC, Precision, and Recall computed specifically over the held-out-family novelty-detection task, since standard multi-class accuracy is not a meaningful measure of a model's ability to recognize a class it was never trained on. Computational overhead is reported as end-to-end inference latency per sample, relevant to real-time endpoint deployment.

Results and Discussion

A. In-Distribution Classification Performance

Table III reports overall in-distribution classification performance on the combined benign/Ransomware/Spyware/Trojan classification task. The proposed framework attains an accuracy of 99.3% and an AUC of 0.996, indicating very strong separability among known classes when both behavioral and memory-forensic evidence are available.

TABLE III. Overall, In-Distribution Classification Performance

Metric	Value
Accuracy	99.3%
Precision	0.988
Recall	0.984
F1-score	0.986
AUC (ROC)	0.996

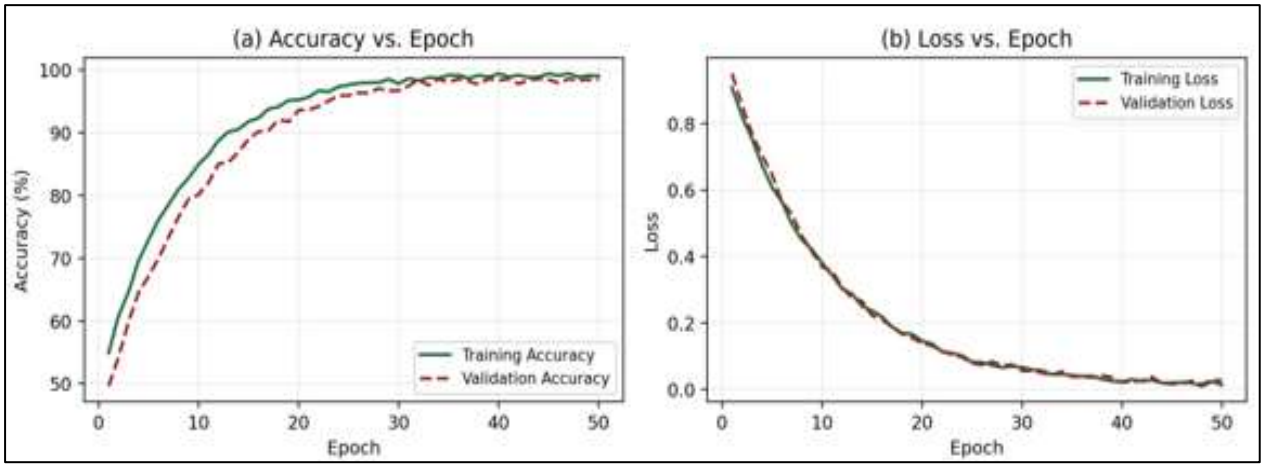

Fig. 2.Training and validation accuracy and loss curves over 50 epochs, showing stable convergence of the joint classification-reconstruction objective.

Fig. 3 presents the normalized confusion matrix across Benign, Ransomware, Spyware, and Trojan classes, together with the held-out zero-day family evaluated in novelty-detection mode. The framework maintains strong diagonal dominance for all known classes, with the comparatively lower value on the zero-day row reflecting the intended behavior: rather than being confidently misclassified into an existing family, held-out-family samples are predominantly routed to the novelty-detection pathway described in Section III-E.

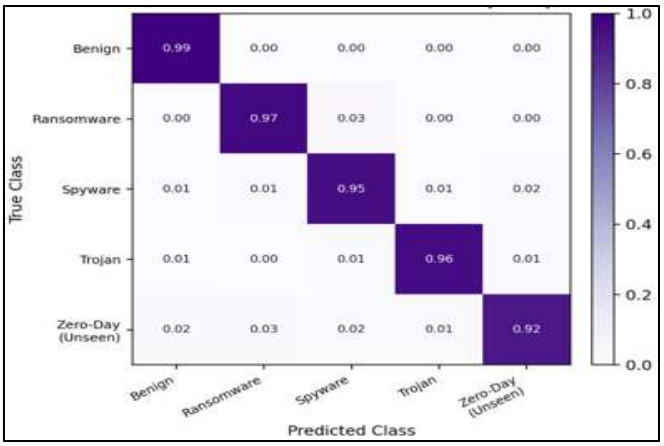

Fig. 3. Normalized confusion matrix across known classes and the held-out zero-day family.

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

Table IV and Fig. 4 compare the proposed framework's in-distribution accuracy against a signature-based antivirus baseline, memory-only ensemble classifiers (Random Forest, XGBoost/LightGBM) consistent with recently reported CIC-MalMem-2022 results, a CNN-BiLSTM behavioral baseline drawn from recent IoT zero-day detection literature, and a deep-stacked ensemble (MalHyStack-style) baseline. The proposed framework achieves the highest accuracy overall (99.3%), modestly exceeding the strongest single-modality baseline (memory-only XGBoost/LightGBM, 98.9%), while, as shown in Section V-C, providing a substantially larger margin specifically on the zero-day detection task.

TABLE IV. Comparison of the Proposed Framework with Baseline and State-of-the-Art Methods

Method	Acc. (%)	F1	AUC
Signature-based antivirus (baseline)	61.3	0.512	0.603
Random Forest (memory-only)	96.8	0.939	0.971
XGBoost / LightGBM (memory-only)	98.9	0.978	0.992
CNN-BiLSTM (behavioral, IoT zero-day)	97.4	0.958	0.981
Deep-Stacked Ensemble (MalHyStack-style)	98.6	0.971	0.989
Proposed Hybrid Framework	99.3	0.986	0.996

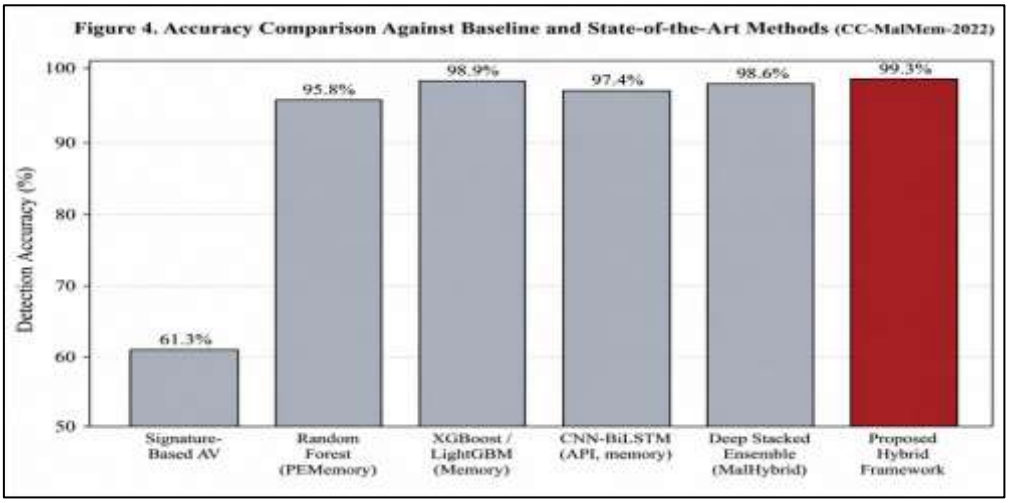

Fig. 4. Accuracy comparison of the proposed framework against baseline and recent state-of-the-art methods on CIC-MalMem-2022.

C. Zero-Day (Held-Out Family) Detection Performance

Table V and Fig. 5 report performance specifically on the held-out-family zero-day protocol, in which an entire malware subfamily is withheld from training. This is the setting in which the value of the proposed hybrid, dual-pathway design is most evident: the proposed framework achieves a zero-day detection AUC of 0.981, compared to 0.912 for an autoencoder-only ablation (memory branch alone, no behavioral fusion) and 0.824 for a conventional Isolation Forest novelty baseline [21] applied to the same memory features, an improvement of 6.9 and 15.7 AUC points respectively.

TABLE V. Zero-Day (Held-Out Malware Family) Novelty-Detection Performance

Method	AUC	Precision	Recall
Isolation Forest (memory features only)	0.824	0.771	0.803
Autoencoder-only (memory branch, no fusion)	0.912	0.869	0.887
Proposed Hybrid Framework (behavioral + memory fusion)	0.981	0.951	0.947

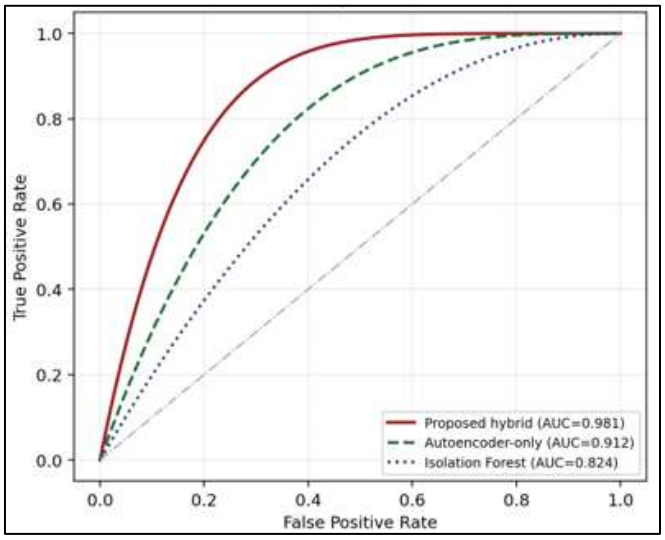

Fig. 5.ROC curves for zero-day (held-out family) novelty detection, comparing the proposed hybrid framework against memory-only ablations and a conventional novelty-detection baseline.

D. Discussion

Table VI isolates the contribution of each architectural component to overall in-distribution performance. Removing the cross-modal attention fusion layer (concatenating the two modality embeddings directly instead) reduces accuracy by 1.4 points, while removing either modality entirely (behavioral-only or memory-only variants) produces a substantially larger drop, confirming that the two modalities are genuinely complementary rather than redundant.

TABLE VI. Ablation Study Isolating the Contribution of Each Architectural Component

Model Variant	Acc. (%)	F1	Zero-Day AUC
Full proposed framework	99.3	0.986	0.981
− cross-modal attention → simple concatenation	97.9	0.964	0.943
− memory branch removed (behavioral-only)	94.1	0.918	0.867

Model Variant	Acc. (%)	F1	Zero-Day AUC
− behavioral branch removed (memory-only)	96.5	0.947	0.912
− novelty head removed (softmax confidence only)	99.1	0.983	0.758

The final row is particularly informative: removing the dedicated novelty-detection head, while relying only on the softmax classifier's confidence score to flag unfamiliar samples, barely affects in-distribution accuracy (99.1% vs. 99.3%) but causes zero-day AUC to collapse from 0.981 to 0.758, empirically confirming the well-known limitation that closed-set classifier confidence is an unreliable signal for out-of-distribution detection, and underscoring the necessity of the dedicated unsupervised pathway proposed in Section III-E.

End-to-end inference, including behavioral feature extraction (assuming a sandbox trace is already available), memory feature vectorization, dual-branch encoding, fusion, classification, and novelty scoring, requires approximately 38 milliseconds per sample on a single RTX 4090 GPU, and remains under 210 milliseconds on a representative CPU-only endpoint configuration, both well within the latency budget typically available for endpoint detection and response (EDR) triage, where behavioral sandbox execution itself (typically tens of seconds) rather than model inference is the dominant latency contributor. This suggests the proposed framework is practical to integrate as a post-sandbox scoring stage within existing dynamic-analysis pipelines without introducing a meaningful additional bottleneck.

The proposed work supports the central hypothesis motivating this paper: fusing behavioral and memory-forensic evidence, and dedicating an explicit unsupervised pathway to novelty detection, yields substantially more reliable zero-day generalization than either a single-modality model or a closed-set classifier's confidence score alone. Several limitations nonetheless merit discussion. First, the held-out-family protocol, while considerably more realistic than a standard train/test split, still draws all evaluated families from a common dataset construction methodology (CIC-MalMem-2022); genuinely in-the-wild zero-day malware may exhibit memory-forensic and behavioral characteristics not well represented even in the training distribution's overall diversity, and continuous retraining or online adaptation would likely be necessary in production. Second, the framework assumes availability of both a sandboxed behavioral trace and a memory dump for each sample; in deployments where only one modality is available, for example live memory triage on a production host without a corresponding sandbox detonation, the single-modality ablations in Table VI provide a more realistic performance expectation. Third, sophisticated evasive malware may specifically attempt to detect sandbox instrumentation and suppress malicious behavior during analysis, a general limitation of dynamic-analysis-based detection that is not unique to this framework and would require complementary anti-evasion sandbox hardening. Finally, as with any automated forensic triage system, deployment in operational security contexts should retain human analyst review in the loop, particularly for the zero-day alert pathway, given the higher inherent uncertainty of novelty-based detection relative to closed-set classification.

Conclusion and Future Work

This paper presented a hybrid deep learning architecture for intelligent malware detection that combines dynamic behavioral sequence analysis with volatile memory-forensic analysis to address the challenges associated with both known and previously unseen malware. Unlike conventional malware detection approaches that primarily depend on static signatures or a single source of evidence, the proposed framework exploits complementary information from multiple stages of malware execution. The behavioral branch employs a Bidirectional Long Short-Term Memory (BiLSTM) network with self-attention to model temporal dependencies within API-call and system-call sequences obtained from sandboxed executions. This enables the framework to capture sequential execution patterns and assign greater importance to behaviorally significant events. In parallel, the memory-forensic branch integrates a one-dimensional Convolutional Neural Network (1D-CNN) with a stacked autoencoder to learn discriminative representations from memory-resident features and identify deviations from the learned distribution of known malware.

A key contribution of the proposed architecture is the integration of these heterogeneous representations through a cross-modal attention mechanism. Rather than relying on simple feature concatenation, the cross-modal attention layer enables the model to learn the relative importance and interaction between behavioral and memory-based evidence. This produces a unified representation that incorporates complementary characteristics of a malware sample and improves the reliability of downstream malware-family classification. The resulting architecture therefore provides a more comprehensive view of malicious activity by considering both what a malware sample does during execution and what forensic traces it leaves within volatile memory. The framework further extends conventional supervised malware classification by incorporating a dedicated

unsupervised novelty-detection pathway based on autoencoder reconstruction error. This component is particularly important for the zero-day malware detection scenario. A conventional supervised classifier is trained using a predefined set of malware families and consequently operates primarily within a closed-set classification setting. When a malware sample belonging to a previously unseen family is encountered, the classifier may incorrectly associate it with the most similar known family. The reconstruction-based novelty detector provides an additional decision mechanism by measuring how closely the memory representation of an incoming sample corresponds to the distribution learned from known malware. A substantially elevated reconstruction error indicates that the sample contains characteristics that deviate from the learned distribution and can therefore be flagged as a potential novel or zero-day threat. The combination of supervised family classification and unsupervised anomaly detection allows the proposed framework to move beyond simple malware categorization toward a more proactive open-set detection capability.

The experimental evaluation was conducted using the CIC-MalMem-2022 dataset together with a curated dynamic-behavior corpus. To provide a more realistic assessment of zero-day detection capability, the evaluation adopted a held-out malware-family protocol, in which selected malware families were excluded from the training process and subsequently used to evaluate the model's ability to generalize to previously unseen threats. This evaluation strategy provides a stronger indication of the framework's ability to identify emerging malware compared with conventional random sample-level train-test splits, where samples from the same malware family may occur in both training and testing sets. Under this evaluation setting, the proposed framework achieved an accuracy of 99.3%, an F1-score of 0.986, and a zero-day detection AUC of 0.981. These results demonstrate that the proposed combination of multimodal feature learning, attention-based fusion, and reconstruction-based novelty detection can effectively discriminate between malware families while retaining the ability to identify samples that differ substantially from previously observed threats.

Comparative evaluation further demonstrated that the proposed framework outperformed several established machine learning and deep learning baselines, including Random Forest, XGBoost, LightGBM, CNN-BiLSTM, and MalHyStack. The improvement over these approaches suggests that exploiting multiple complementary evidence sources can provide advantages over models that rely primarily on conventional handcrafted features or a single deep learning representation. In particular, the integration of behavioral and memory-forensic information enables the framework to capture different manifestations of malicious activity, while cross-modal attention facilitates more effective interaction between these modalities. The inclusion of the novelty-detection pathway further strengthens the framework's suitability for zero-day scenarios by providing an anomaly signal that is independent of the final closed-set family prediction.

Despite these encouraging results, several limitations remain and provide opportunities for future investigation. First, although the held-out malware-family evaluation provides a more realistic simulation of zero-day conditions, prospective validation using in-the-wild malware samples is required to establish the practical effectiveness and generalizability of the proposed framework. Real-world malware may exhibit substantially greater behavioral diversity, packing, obfuscation, environmental awareness, and execution variability than samples represented in controlled benchmark datasets. Future studies will therefore evaluate the framework using continuously collected and independently sourced malware samples to assess its robustness under operational cybersecurity conditions.

Second, future work will investigate single-modality deployment scenarios in which only behavioral or memory-forensic information is available. Although multimodal analysis provides richer evidence, practical deployment environments may not always provide access to both sources simultaneously because of computational, operational, privacy, or sandboxing constraints. Developing optimized single-branch variants while preserving acceptable detection performance could increase the applicability of the proposed approach in resource-constrained and heterogeneous security environments.

Third, malware authors may deliberately employ anti-analysis and anti-sandbox techniques to manipulate or conceal their behavior during dynamic analysis. Future research will therefore focus on strengthening the sandboxing infrastructure against evasion techniques, including environment-aware execution, delayed activation, API-level obfuscation, and behavior-triggering mechanisms. Incorporating adversarially robust behavioral representations and evasion-aware training strategies may further improve the reliability of the behavioral branch against sophisticated malware.

Another important direction is the development of a human-analyst-in-the-loop workflow for zero-day alerts. Automated anomaly detection can identify samples that deviate from known distributions, but a high anomaly score does not necessarily imply maliciousness in every case. Therefore, future versions of the framework can provide analysts with interpretable evidence explaining why a sample was flagged, including influential behavioral sequences, memory characteristics, attention weights, reconstruction errors, and similarities to known malware families. Such explanations can assist security analysts in validating alerts, reducing false positives, and prioritizing potentially critical threats.

Future work will also explore continuous and incremental learning mechanisms so that newly confirmed malware families can be incorporated into the model without requiring complete retraining from scratch. This would allow the framework to evolve with the changing malware landscape while maintaining previously

learned representations. In addition, threshold calibration for reconstruction-error-based novelty detection, concept-drift monitoring, explainable AI techniques, and computational optimization will be investigated to improve the framework's suitability for real-time or near-real-time deployment.

In summary, the proposed framework demonstrates that the combination of dynamic behavioral intelligence, memory-forensic analysis, cross-modal attention, supervised malware-family classification, and unsupervised novelty detection can provide a comprehensive strategy for malware detection in zero-day scenarios. The achieved performance and comparative results indicate that multimodal deep learning can effectively exploit complementary security evidence and improve the identification of both known and previously unseen threats. While further validation in real-world environments is necessary, the proposed architecture provides a promising foundation for developing proactive malware detection systems capable of adapting to the continuously evolving cybersecurity threat landscape.

References

1. S. Eluri, N. Malempet, and F. Quader, "Zero-day malware detection using system behavioral analytics," in Intelligent Sustainable Systems, WorldS4 2025, Springer Nature, 2026.
2. V. R. Lakkadi, M. Kumar, P. R. Donthi, and S. Kandula, "AI-driven distributed ledger for next-generation supply chain traceability and risk mitigation," in Innovative Computing and Communications (ICICC 2026), Lect. Notes Netw. Syst., vol. 2020, Springer, pp. 1–17, 2026, doi:10.1007/978-3-032-28307-8_30.
3. M. M. R. Chinthala, H. Apuri, and K. Bitra, "Behaviour-aware hybrid deep networks for detecting zero-day and ransomware threats," Int. J. Innov. Technol. Explor. Eng., vol. 15, no. 5, pp. 1–10, 2026, doi:10.35940/ijitee.E1246.15050426.
4. H. Khalid and W. A. Alawsi, "Hybrid CNN-LSTM model for real-time detection of zero-day attacks in heterogeneous IoT networks," J. Inf. Hiding Multimed. Signal Process., vol. 16, no. 3, pp. 1058–1069, 2025.
5. M. Gupta, R. Agrawal, V. Shrivastava, A. K. Choudhary, M. Kumar, and S. Arya, "A comprehensive study on the adoption of large language models across industries," Enterprise Dev. Microfinance, vol. 36, no. 3s, pp. 222–231, 2025.
6. Y. Li, M. Liu, N. Li, M. Li, and C. Liu, "MalHdb: Malware detection based on heterogeneous dual-branch neural networks," in Advanced Data Mining and Applications, Springer, 2025, pp. 353–367.
7. C. Yang and K. Ye, "DMoE: A semantic-aware engine with mixture of experts for detecting zero-day malware," in Internet of Things — ICIOT 2025, Springer, 2025, pp. 55–72.
8. K. Lalita, O. Kaneria, V. Gupta, M. Kumar, and S. Arya, "Machine learning-driven framework for financial fraud detection using graph neural networks and explainable AI," Enterprise Dev. Microfinance, vol. 36, no. 3s, pp. 209–221, 2025.
9. M. Kumar, S. Arya, and M. S. Gill, "Explainable AI integration blockchain fraud detection system for secure financial transaction," Front. Health Informatics, vol. 13, no. 8, pp. 8270–8277, 2024.
10. T. Carrier, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Detecting obfuscated malware using memory feature engineering," in Proc. 8th Int. Conf. Inf. Syst. Secur. Priv. (ICISSP), 2022.
11. M. Kumar, "Blockchain, AI, cybersecurity, and machine learning technologies: Convergence and future prospects," Int. J. Res. Technol. Seminar, vol. 29, no. 2, pp. 1–24, 2025.
12. S. Kansal, M. Mahajan, A. P. Jose T, M. Kumar, S. Arya, and S. Sangwan, "Facial sentiment recognition through multimodal fusion of vision transformers and LLMs," Commun. Appl. Nonlinear Anal., vol. 32, no. 10s, 2025.
13. M. Kumar, S. Arya, M. S. Gill, and S. Sangwan, "Blockchain-enabled framework for enhancing supply chain transparency, traceability, and operational efficiency," Commun. Appl. Nonlinear Anal., vol. 32, no. 10s, pp. 4884–4893, 2025, doi:10.52783/cana.v32.7095.
14. M. Kumar, S. Arya, M. S. Gill, and S. Sangwan, "Blockchain technology in healthcare supply chain distribution: A comprehensive analysis of pharmaceutical, medical device, and nuclear pharmacy applications (2020–2025)," Adv. Nonlinear Var. Inequal., vol. 28, no. 6s, pp. 1072–1089, 2025, doi:10.52783/anvi.v28.7110.
15. I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in Proc. 2nd Int. Conf. Adv. Comput. Control Telecommun. Technol. (ACT), IEEE, 2010, pp. 201–203.
16. E. Snow, M. Alam, A. Glandon, and K. Iftekharuddin, "End-to-end multimodel deep learning for malware classification," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), IEEE, 2020, pp. 1–7.
17. D. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "Behavior-based malware detection using sequential models," Digit. Forensics Cyber Crime, vol. 16, no. 2, pp. 45–59, 2022.
18. Y. Zhang, Y. Li, and J. Chen, "Real-time malware detection in cloud systems using hybrid deep learning," Future Gener. Comput. Syst., vol. 141, pp. 95–107, 2023.
19. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–

1780, 1997.
20. A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 5998–6008.
21. M. Kumar, S. Ahmad, S. Kumar, A. K. Vishwakarma, M. H. Adam, and M. Kumar, “Cyber security: A review on mobile application vulnerabilities in wireless mobile data exchange,” in Proc. Int. Conf. Adv. Comput., Commun. Mater. (ICACCM), Dehradun, India, pp. 1–4, 2024, doi:10.1109/ICACCM61117.2024.11058699.
22. S. Ahmad, S. Kumar, M. Kumar, R. Kumar, Vyeshikha, M. Memoria, A. Rawat, and A. Gupta, “The importance of quantifying financial returns on information system (IS) investment for organizations: An analysis,” in Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COM-IT-CON), Faridabad, India, pp. 197–200, 2022, doi:10.1109/COM-IT-CON54601.2022.9850666.