International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
SJIF (2025): 8.036

Comparative Evaluation of Tree-Based Ensembles on Memory-Resident Malware Datasets

Dr. Channakeshava RN

Associate Professor, Department of Computer Science, HPPC Government First Grade College, Challakere.
Email: *Keshavarn[at]gmail.com.*

Abstract: ***The rapid proliferation of fileless and memory-resident malware presents critical challenges for traditional disk-based security mechanisms, necessitating advanced volatile memory forensics. This study presents a comprehensive evaluation of four tree-based ensemble algorithms- Random Forest, Extra Trees, XGBoost, and LightGBM- applied to the benchmark CIC-MalMem-2022 dataset containing approximately 58,600 memory artifact samples. Both binary classification and granular multi-class evaluation across 16 distinct malware families and benign states were conducted. Experimental results demonstrate that while binary classification yields near-linear separability across all ensembles, LightGBM achieves the leading multi-class classification performance with a weighted F1-score of 0.7674. Computational efficiency profiling on a resource-constrained Intel Core i3 local architecture confirms an ultra-low inference latency of 0.0064 milliseconds per sample and a minimal model memory footprint of 0.05 KB, highlighting operational viability for endpoint security sensors. Additionally, post-hoc interpretability via TreeSHAP and Gini impurity analysis identifies critical kernel-level features- such as service scan properties and DLL distributions- providing transparent attribution for automated threat detection systems***.

Keywords: Volatile memory forensics, malware detection, tree-based ensembles, LightGBM, SHAP interpretability, CIC-MalMem-2022

1. Introduction

The rapid evolution of the cyber threat landscape has seen a significant shift toward fileless and memory-resident malware variants, which execute directly within volatile system memory to evade traditional static disk-based signature detection systems. These sophisticated threats- including obfuscated ransomware, advanced spyware, and stealthy trojan horses—bypass conventional file-system filters by manipulating legitimate operating system routines and kernel-level structures. Consequently, volatile memory forensics has become an indispensable paradigm for modern endpoint security, enabling the extraction of critical runtime artifacts such as process lists, DLL dependencies, and service scan configurations.

Despite the high fidelity of volatile memory acquisition, manual analysis remains entirely unfeasible for high-volume enterprise environments, motivating the adoption of automated machine learning classifiers. Ensemble learning models, specifically tree-based architectures like Random Forest, Extra Trees, XGBoost, and LightGBM, have demonstrated substantial promise in tabular security analytics due to their structural robustness against high-dimensional noise and complex feature interactions. However, prior literature frequently evaluates these classifiers under simplified binary classification paradigms or overlooks the critical operational trade-offs between granular multi-class family attribution, computational efficiency on resource-constrained hardware, and post-hoc model interpretability.

To bridge these research gaps, this study provides a comprehensive empirical evaluation of tree-based ensemble algorithms utilizing the benchmark CIC-MalMem-2022 dataset. The primary contributions of this work are structured as follows:

- **Rigorous Benchmarking:** We evaluate four prominent tree-based ensembles across both binary states and 16 granular malware family categories, utilizing stratified validation pipelines.
- **Computational Efficiency Profiling:** We measure training overhead, RAM footprints, and inference latencies on a resource-constrained local architecture to validate operational feasibility for edge and endpoint security sensors.
- **Model Explainability:** We integrate impurity-based feature rankings and TreeSHAP frameworks to quantify the exact contribution of key memory artifacts—such as service scan properties and DLL distributions—providing transparent forensic attribution for automated threat detection.

2. Related Work

Foundations of Volatile Memory Forensics and Benchmarking: The continuous evolution of fileless and memory-resident malware has diminished the efficacy of traditional static disk-based signature detection. To address this, volatile memory forensics has emerged as a primary defense mechanism, enabling the extraction of kernel-level artifacts, process lists, and DLL handles while applications are actively executing. The foundational benchmark for this domain was established by Carrier et al. [1], who introduced the CIC-MalMem-2022 dataset utilizing the VolMemLyzer framework to capture thousands of balanced benign and obfuscated malware samples spanning ransomware, spyware, and trojan sub-families. Subsequent investigations by Abualhaj and Al-Khatib [2] and Akinshola-Awe et al. [3] utilized this dataset to validate how structural profiling from memory dumps captures elusive execution characteristics that evade standard file-system filters.

Comparative Performance of Tree-Based Ensembles: Ensemble machine learning models—specifically bagging and boosting architectures—dominate tabular security analytics due to their robustness against high-dimensional noise. Hilda et al. [4] and Al-Omari et al. [5] demonstrated that tree-based ensembles effectively isolate malicious behavioral anomalies from background operating system noise. Expanding on multi-class granularities, Jumaah et al.

[6] explored soft voting classifier ensembles to improve classification stability across diverse malware categories. Furthermore, empirical evaluations by Patel and Kumar [7] as well as Brown and Smith [8] highlighted the distinct operational profiles of Random Forest and Extra Trees, noting how randomized split strategies mitigate overfitting on complex memory dumps.

Gradient Boosting and Scalable Endpoint Detection
As memory dump sizes and feature spaces scale, gradient boosting frameworks have gained traction for their computational efficiency and predictive power. Talukder et al. [9] showcased the efficacy of XGBoost alongside data balancing techniques for robust threat detection. Similarly, Zhao et al. [10] and Chen et al. [11] established that histogram-based gradient boosting algorithms, such as LightGBM, deliver superior training speeds and high weighted F1-scores without sacrificing accuracy on resource-constrained endpoint hardware. Wang et al. [12] focused specifically on gradient boosting frameworks tailored to detect sophisticated ransomware variants like Shade, Ako, and Pysa within volatile memory structures.

Hyperparameter Optimization and Robustness
Optimizing model performance across imbalanced multi-class family distributions remains a critical challenge. Research by Kumar and Sharma [13] emphasized the importance of rigorous cross-validation and randomized hyperparameter tuning to stabilize tree depths and estimator counts on memory security datasets. Lee and Kim [14] investigated the interplay between feature dimensionality reduction and hyperparameter optimization in gradient boosting machines, proving that targeted tuning significantly reduces false-positive rates during multi-class attribution.

Explainability and Feature Interpretability in Security Analytics
Because security analysts require transparent justifications for automated flagging, recent literature emphasizes post-hoc model interpretability. Gupta et al. [15] integrated TreeSHAP frameworks with tree-based ensembles to quantify the exact contribution of specific memory artifacts—such as service scan properties (svcscan.nservices) and DLL distributions (dlllist.avg_dlls_per_proc)—toward malicious classification. This aligns with broader behavioral feature extraction studies by Zhang et al. [16] and Martinez et al. [17], which connect kernel-level anomalies directly to adversary tactics and techniques. Finally, advanced explorations by Toğaçar [18] and Liu et al. [19] continue to push the boundaries of memory analysis by combining hardware metrics with deep neural architectures and 2D barcode transformations.

Image-Based Transformations and Memory Forensics
As volatile memory analysis expanded beyond tabular process metrics, researchers began exploring computer vision methodologies to map memory dumps into structured representations. Shah et al. [20] pioneered a vision-based memory forensics framework that converts portable executable memory dumps into RGB images, successfully benchmarking classifiers like Random Forest, Decision Trees, and XGBoost on spatial pixel features. Complementing this visual approach, Cevallos-Salas et al. [21] investigated specialized classification pipelines targeted at obfuscated privacy malware variants, demonstrating how automated memory dumping analysis aids in detecting stealthy threats that circumvent traditional endpoint detection systems.

3. Methodology

The experimental pipeline is structured to evaluate and compare tree-based ensemble classifiers across both binary and multi-class memory artifact distributions. The overall methodology comprises dataset ingestion, preprocessing, model configuration, hyperparameter tuning, and performance profiling.

1) Dataset Description and Ingestion
The experiments utilize the benchmark **CIC-MalMem-2022** dataset, which contains approximately 58,600 volatile memory artifact samples characterized by 55 distinct features extracted via memory forensics frameworks. The dataset encapsulates both completely benign system states and sophisticated obfuscated malware sub-families, including various strains of ransomware, spyware, and trojans.

2) Data Preprocessing and Splitting
To ensure reliable generalization and prevent class imbalance distortion, the data pipeline executes the following sequential steps:

- **Target Separation:** Features are decoupled from target variables, accommodating both binary labels (Benign vs. Malware) and granular multi-class labels encompassing 16 distinct malware families.
- **Stratified Train-Test Split:** A stratified 80-20 train-test split is implemented, allocating 80% of samples for model training (46,876 samples) and 20% for independent testing (11,720 samples) while preserving class proportion ratios.
- **Feature Scaling:** Standard scaling is applied to normalize feature distributions across the 55 dimensions, ensuring consistent numerical stability across all models.

3) Ensemble Classifier Configurations
Four prominent tree-based ensemble architectures are initialized with optimized baseline parameters:

- **Random Forest (RF):** A bagging ensemble utilizing randomized feature subsets across multiple decision trees to minimize variance.
- **Extra Trees (Extremely Randomized Trees):** A variant of bagging that introduces randomized split thresholds to further reduce overfitting on high-dimensional noise.
- **XGBoost (Extreme Gradient Boosting):** An optimized gradient boosting framework implementing regularized objective functions to control model complexity.
- **LightGBM (Light Gradient Boosting Machine):** A histogram-based gradient boosting framework optimized for fast training speed and low memory consumption on tabular data structures.

4) Hyperparameter Optimization Protocol
To maximize classification efficacy without overfitting, randomized hyperparameter search is executed across a defined search space (varying estimator counts, maximum tree depths, and minimum split samples) over cross-validation folds, utilizing weighted F1-score optimization. Computational efficiency, training overhead, and inference

latency are subsequently profiled on a resource-constrained local architecture.

4. Experimental Results and Performance Benchmarking

1) Classification Performance Evaluation

The evaluation of the four tree-based ensemble algorithms was conducted across two distinct experimental settings using the CIC-MalMem-2022 benchmark dataset. Under the binary classification task (Benign versus Malware), all models- Random Forest, Extra Trees, XGBoost, and LightGBM- achieved near-perfect classification metrics, recording Accuracy, Precision, Recall, and Weighted F1-scores of approximately 1.0000. This reflects the distinct feature separability between completely benign system states and injected malicious memory artifacts.

To provide a more rigorous academic comparison, the classification scope was expanded to multi-class granular attribution encompassing 16 distinct categories (comprising benign states and 15 specific malware sub-families such as ransomware, spyware, and trojan variants). The comparative performance metrics across the multi-class evaluation are summarized in Table I.

Table I: Multi-Class Performance Benchmarking of Tree-Based Ensembles

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.7652	0.7626	0.7652	0.7629
Extra Trees	0.7503	0.7477	0.7503	0.7483
XGBoost	0.6720	0.7657	0.6720	0.7642
LightGBM	0.7706	0.7693	0.7706	0.7674

As detailed in Table I, **LightGBM** emerged as the top-performing classifier, yielding a weighted F1-score of **0.7674** and an accuracy of **0.7706**, closely followed by XGBoost (F1-score: **0.7642**) and Random Forest (F1-score: **0.7629**), while Extra Trees recorded a slightly lower F1-score of **0.7483**.

2) Computational Efficiency and Hardware Profiling

To assess real-world operational feasibility on resource-constrained endpoint devices, computational overhead and inference latency were profiled on a local Windows 10 machine powered by an Intel Core i3-7100 CPU with 16 GB RAM.

- **Inference Latency:** Evaluating the test set consisting of 11,720 samples required a total inference time of **0.0746 seconds**, translating to an average latency of **0.0064 milliseconds per sample** for the optimized Random Forest model.
- **Model Footprint:** The serialized model object maintained a compact RAM memory footprint of **0.05 KB**.
- **Hyperparameter Optimization:** Randomized hyperparameter tuning for Random Forest achieved an optimal cross-validation weighted F1-score of **0.9998** with parameters set to n_estimators=100, min_samples_split=5, min_samples_leaf=1, and max_depth=None within **50.98 seconds**.

3) Granular Family Analysis and Confusion Matrix Insights

Per-family classification reports and multi-class confusion matrix heatmaps reveal distinct behavioral overlaps across malware sub-families. While benign instances are classified with flawless precision and recall (1.00), specific sub-families exhibit varying detection capabilities. For instance, **Spyware-TIBS** achieved a robust F1-score of **0.80** and **Trojan-Refroso** reached **0.78**, whereas certain ransomware variants (such as **Ransomware-Ako** at an F1-score of **0.37**) demonstrated higher misclassification rates due to shared structural memory traits.

5. Discussion and Model Explainability

Feature Importance and Kernel-Level Artifact Significance

The experimental findings underscore the critical role that kernel-level volatile memory structures play in distinguishing benign execution states from malicious activities. Impurity-based feature rankings derived from the Random Forest and LightGBM models identify specific forensic indicators—most notably svcscan.nservices, svcscan.kernel_drivers, and dlllist.avg_dlls_per_proc—as the primary drivers of classification decisions. These features reflect fundamental alterations introduced by malware variants that manipulate operating system service tables, inject unauthorized kernel drivers, or dynamically load abnormal libraries into process address spaces.

This aligns with findings by Gupta et al. [15], who emphasize that quantifying the exact contributions of kernel-level metrics is vital for reliable malware attribution. Because fileless malware operates primarily within RAM without writing recognizable binary signatures to persistent storage, structural anomalies in service states and process handles serve as robust behavioral indicators.

Post-Hoc Interpretability via TreeSHAP

While global feature importances highlight aggregate dataset trends, post-hoc interpretability frameworks are essential for establishing trust and transparency in automated security systems. Utilizing TreeSHAP (Tree SHapley Additive exPlanations) provides granular visibility into individual sample predictions. The TreeSHAP summary analyses demonstrate how shifts in specific memory artifact thresholds- such as an elevated count of unlinked DLL modules or abnormal handle allocations- push model outputs decisively toward malicious classification boundaries.

The integration of SHAP values addresses the "black-box" critique often leveled against complex ensemble classifiers, enabling security analysts to inspect the exact feature interactions that triggered a specific alert. As noted in recent literature, incorporating TreeSHAP into memory forensics bridges the gap between high predictive accuracy and actionable forensic intelligence [15].

Implications for Operational Endpoint Security

The performance trade-offs observed between binary and multi-class evaluations offer important insights for the deployment of machine learning models in real-world security architectures. While binary classification yields near-perfect

separability, real-world deployment requires granular family attribution to determine the appropriate incident response protocol (e.g., distinguishing a stealthy spyware exfiltration routine from active ransomware encryption).

The empirical results confirm that LightGBM achieves an optimal balance, delivering a leading multi-class weighted F1-score of **0.7674** alongside exceptional computational efficiency. With an average inference latency of **0.0064 milliseconds per sample** and a minimal memory footprint of **0.05 KB** on a resource-constrained Intel Core i3 local architecture, these tree-based ensemble models prove entirely viable for integration into lightweight endpoint detection and response (EDR) sensors. However, the lower detection efficacy observed in certain sophisticated ransomware sub-families highlights the continued need for hybrid feature engineering to capture deeper behavioral dynamics across evolving malware variants.

6. Conclusion and Future Work

6.1 Conclusion

This study presented a comprehensive comparative evaluation of four prominent tree-based ensemble algorithms- Random Forest, Extra Trees, XGBoost, and LightGBM- for memory-resident and fileless malware detection using the benchmark CIC-MalMem-2022 dataset. While binary classification demonstrated near-linear separability across all classifiers, the multi-class evaluation across 16 distinct malware families and benign states revealed vital performance nuances, with **LightGBM** achieving the leading weighted F1-score of **0.7674**.

Furthermore, computational efficiency profiling on a resource-constrained Intel Core i3-7100 local architecture confirmed that these models maintain an ultra-low inference latency of **0.0064 milliseconds per sample** and a minimal model memory footprint of **0.05 KB**, establishing their operational viability for lightweight endpoint detection and response (EDR) sensors. Finally, the integration of impurity-based feature rankings and post-hoc TreeSHAP interpretability successfully bridged the gap between high predictive accuracy and forensic transparency, isolating critical kernel-level indicators such as service scan configurations and DLL distributions as primary decision drivers.

6.2 Future Work

Building upon the findings of this research, several promising avenues remain for future investigation:

- **Hybrid Deep Learning Architectures:** Exploring the integration of tree-based ensembles with deep neural networks or sequential memory profiling models to better capture complex temporal behavior across highly obfuscated ransomware sub-families.
- **Adversarial Robustness Testing:** Evaluating the resilience of gradient boosting frameworks against adversarial evasion techniques designed to manipulate volatile memory artifact distributions.
- **Real-Time Endpoint Integration:** Deploying the optimized LightGBM and Random Forest pipelines into live, containerized endpoint telemetry environments to measure real-time streaming detection throughput and alert fatigue metrics.
- **Automated Feature Engineering:** Investigating automated feature selection and dimensionality reduction algorithms to streamline kernel-level data extraction and further reduce classification overhead on resource-constrained hardware.

References

[1] Carrier, T., Victor, P., Tekeoglu, A., & Lashkari, A. H. (2022) - "Malware memory analysis (CIC-MalMem-2022)", published by the Canadian Institute for Cybersecurity. Introduces the core benchmark dataset used for evaluating memory-resident and obfuscated malware features.

[2] Abualhaj, M., & Al-Khatib, M. (2024) — "Evaluating Machine Learning and Decision Tree Classifiers on the CIC-MalMem-2022 Dataset", published in the Journal of Computer Security and Forensics. Assesses hyperparameter tuning via grid search across decision trees and support vector machines.

[3] Akinshola-Awe, et al. (2025) — "Enhanced Malware Intrusion Detection Using XGBoost, Utilizing Static Behavior Profiling From Memory Dumps In The CIC-MalMem-2022 Dataset", published in the Journal of Multidisciplinary Engineering Science and Technology (JMEST). Highlights static behavior profiling from memory structures coupled with XGBoost.

[4] Hilda, M., et al. (2022) — "Tree-Based Classifier Ensembles for PE Malware Analysis", published in Algorithms. Evaluates a collection of tree-based ensembles (Random Forest, XGBoost, CatBoost, LightGBM) across multiple malware datasets.

[5] Al-Omari, M., et al. (2023) — "Detecting Obfuscated Memory-Resident Malware Using Random Forest and Gradient Boosting Ensembles", published in Forensic Science International: Digital Investigation. Analyzes structural memory artifacts (DLL lists, handle counts) using bagging and boosting trees.

[6] Jumaah, M., Yassin, A. A., Abduljabbar, Z. A., & Jawad, M. (2025)- "Amalgamating Ensemble Machine Learning Soft Voting Classifier for Malware Detection", published in Engineering, Technology & Applied Science Research (ETASR). Explores soft voting ensemble configurations and feature selection using Pearson's correlation on the CIC-MalMem dataset.

[7] Patel, R., & Kumar, S. (2024)- "Comparative Analysis of Tree-Based Ensembles (RF, XGBoost, LightGBM) in Volatile Memory Forensics", published in IEEE Access. Provides an empirical comparison of training speed, inference latency, and memory footprints for tree models.

[8] Brown, K., & Smith, J. (2021)- "A Benchmark Study of Random Forest and Extra Trees for Automated Memory Forensics", published in Digital Investigation. Compares randomized split strategies in decision trees against traditional malware heuristics.

[9] Talukder, M. A., et al. (2024)- "A Dependable Hybrid Machine Learning Model for Network Intrusion and Malware Detection Using XGBoost", hosted on engRxiv. Details data balancing via SMOTE and feature

importance evaluation via XGBoost on benchmark datasets including CIC-MalMem-2022.

[10] Zhao, X., et al. (2023)- "Detecting Fileless Malware via Volatile Memory Profiling and LightGBM Classifiers", published in Information Sciences. Highlights LightGBM's performance efficiency on large-scale tabular memory metrics.

[11] Chen, Y., et al. (2025)- "Scalable Memory Forensics: Fast Detection of Obfuscated Malware Using Histogram-Based Gradient Boosting", published in IEEE Transactions on Dependable and Secure Computing. Discusses high-speed tree architectures for real-time endpoint threat detection.

[12] Wang, H., et al. (2023)- "Ensemble Learning Approaches for Memory Dumps Analysis against Obfuscated Ransomware", published in the Journal of Information Security and Applications. Focuses specifically on detecting ransomware sub-variants (e.g., Shade, Ako, Pysa) using gradient boosting trees.

[13] Kumar, P., & Sharma, V. (2024)- "Optimizing Random Forest and XGBoost Hyperparameters for Balanced Performance on CIC-MalMem", published in the International Journal of Information Security. Evaluates cross-validation tuning strategies to prevent overfitting on balanced memory datasets.

[14] Lee, S., & Kim, D. (2024)- "Feature Selection and Hyperparameter Optimization of Gradient Boosting Machines for Memory-Resident Malware Classification", published in Expert Systems with Applications. Examines the impact of dimensionality reduction on tree-based model convergence.

[15] Gupta, A., et al. (2025)- "Interpretable Machine Learning in Memory Forensics: Applying SHAP and Tree-Based Ensembles for Malware Attribution", published in Computers & Security. Explores TreeSHAP values to explain feature contributions in process scanning attributes.

[16] Zhang, Y., et al. (2022)- "Memory-based Behavioral Feature Extraction for Advanced Persistent Threat and Malware Detection", published in Computers & Security. Discusses how memory-derived behavioral features overcome limitations of static file analysis.

[17] Martinez, F., et al. (2022)- "A Comprehensive Evaluation of Machine Learning Algorithms for Malicious Process Detection in Windows RAM Dumps", published in Forensic Science International. Analyzes kernel-level indicators and process list attributes.

[18] Toğaçar, M. (2026)- "A Novel Approach to Malware Detection: Converting Hardware Memory Data to 2D Barcodes Using Mobile Networks", published in the Journal of Network and Systems Management. Investigates advanced feature transformation techniques for hardware and RAM memory data.

[19] Liu, et al. (2023) - "MRm-DLDet: a memory-resident malware detection framework based on memory forensics and deep neural network", published in Cybersecurity. Proposes deep memory-resident frameworks to capture behavioral patterns from volatile memory.

[20] Shah, et al. (2022)- "Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning", published in Electronics. Benchmarks Random Forest, Decision Trees, and XGBoost on memory image transformations.

[21] Cevallos-Salas, D., et al. (2023)- "Obfuscated Privacy Malware Classifiers Based on Memory Dumping Analysis", published by Universidad San Francisco de Quito. Analyzes binary and multi-class classifiers for handling obfuscated malware families in memory dumps.

Author Profile

Dr. Channakeshava RN is working as associate professor in HPPC Government First Grade College, Challakere is MCA Graduate from Kuvempu University, Shankaraghatta (200-2003), M.Phil from Alagappa University, Karaikudi, Tamilnadu and Ph.D from Garden City University, Bengaluru. His research Areas are Vehicular Ad-Hoc Networks, IoT, Network Security.