Hybrid Ransomware Detection: A PSO-Optimized Combination of CNN-LSTM and XGBoost

Kebbas Mohammed Houssameddine[1], M. Bendella Mohammed Salih[2], and M. Bensenane Hamdane[3]

[1]Main author, Master's student. Email: mh.kebbas@esi-sba.dz, Higher School of Computer Science, 8 Mai 1945 - Sidi Bel Abbes
[2]Thesis Supervisor. Higher School of Computer Science, 8 Mai 1945 - Sidi Bel Abbes
[3]Co-supervisor. Higher School of Computer Science, 8 Mai 1945 - Sidi Bel Abbes

Abstract

The increasing sophistication of ransomware renders traditional detection methods, based on signatures or simple heuristics, progressively obsolete. Artificial Intelligence-based approaches offer a promising alternative but often struggle to find an optimal balance between comprehensive threat detection (recall) and the minimization of false alarms (precision), especially on noisy and imbalanced datasets. This work introduces a novel hybrid architecture for ransomware detection based on the behavioral analysis of Input/Output (I/O) traces. Our two-stage approach first employs a Deep Learning model combining Convolutional (CNN) and Long Short-Term Memory (LSTM) neural networks with an attention mechanism to extract complex temporal features. These semantic features are then fused with raw statistical features from the sequence to feed a second-level classifier based on XGBoost. To maximize performance, the latter's hyperparameters are finely tuned using a Particle Swarm Optimization (PSO) algorithm guided by a custom fitness function that weights both the F2-score and precision. Evaluated on two heterogeneous datasets (one imbalanced, one balanced), our final solution achieves an accuracy of 84.83%, a recall of 93.49%, and an F2-score of 90.50%, demonstrating superior generalization capability and a better performance/reliability trade-off compared to monolithic approaches.

Keywords: Ransomware Detection, Deep Learning, Hybrid Model, CNN, LSTM, XGBoost, Particle Swarm Optimization (PSO), Behavioral Analysis, I/O Traces.

1 Introduction

The proliferation and increasing sophistication of ransomware represent one of the most critical threats to modern information systems security. Capable of paralyzing entire organizations by encrypting their vital data, these attacks lead to considerable financial and operational losses [4]. The emergence of the *Ransomware-as-a-Service* (RaaS) model has democratized cybercrime, accelerating the evolution of new variants that easily evade traditional defenses.

Indeed, conventional detection approaches show fundamental limitations. **Static** methods, based on signature analysis, are inherently ineffective against unknown or polymorphic threats [2]. **Dynamic** methods, which analyze behavior in a sandbox, although more robust, often suffer from delayed detection and are vulnerable to sophisticated evasion techniques [8]. More recently, Machine Learning (ML) and Deep Learning (DL) based approaches have shown great potential. However, many models struggle to find an optimal balance between a high **recall** (the ability to detect all attacks) and a high **precision** (the ability to minimize false alarms), a crucial trade-off for any viable security solution. Furthermore, their performance often varies significantly depending on the nature and distribution of the data (e.g., imbalanced cloud environments versus standard workstations).

To address these challenges, this paper proposes a novel hybrid and optimized ransomware detection architecture based on the behavioral analysis of Input/Output (I/O) traces. Our approach combines the power of deep neural networks for complex temporal feature extraction with the robustness of an ensemble classifier for decision-making. The main contribution of this work is to demonstrate that by fusing semantic features (from a CNN-LSTM model) with raw statistical features, and by rigorously optimizing the final classifier's (XGBoost) hyperparameters via a metaheuristic (PSO), it is possible to obtain a model that not only achieves state-of-the-art performance but is also robust and capable of generalizing across heterogeneous data environments.

The contributions of this paper are as follows:

- The design of a hybrid detection architecture that separates feature extraction via Deep Learning from classification via XGBoost.
- A feature fusion method that combines the semantic outputs of the deep model with raw statistical features for improved discrimination.
- The application of Particle Swarm Optimization (PSO) to find a quasi-optimal hyperparameter configuration by maximizing a custom fitness function that balances recall and precision.
- A rigorous experimental validation on two distinct datasets, demonstrating the robustness and adaptability of our approach.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 details our proposed approach. Section 4 describes the experimental protocol and presents the results. Section 5 discusses these results and their implications. Finally, Section 6 concludes this paper.

2 Related Work

Ransomware detection has garnered considerable attention from the research community, leading to the development of numerous approaches. These can be broadly classified into static, dynamic, and hybrid methods, with a growing trend towards the integration of artificial intelligence.

2.1 Static Analysis-Based Approaches

Static analysis examines executable files without running them, offering speed and safety. Early approaches focused on signatures, strings, or PE header information. A recent and notable evolution of this method is **binary visualization**, where the executable file is converted into an image to be analyzed by Convolutional Neural Networks (CNNs). Moreira et al. [3] demonstrated the effectiveness of this technique using an Xception model, achieving 98.20% accuracy on a balanced dataset. However, although resilient to some forms of code obfuscation, these static methods are fundamentally unable to capture the malware's **runtime behavior**, which is often the most reliable indicator of its maliciousness.

2.2 Dynamic Behavioral Analysis-Based Approaches

To overcome the limitations of static analysis, the majority of research has shifted towards dynamic analysis. This approach involves executing the malware in a controlled environment (sandbox) and monitoring its interactions with the system. The foundational work of Kharaz et al. with **UNVEIL** [2] showed that by analyzing Input/Output (I/O) operations and graphical user interface modifications, it was possible to detect ransomware with very high performance (96.3% recall and zero false positives).

Nevertheless, classic dynamic approaches face two major challenges: the **slowness of detection**, which can occur after encryption has already begun, and **sandbox evasion techniques**, whereby malware detects the analysis environment and conceals its malicious behavior.

2.3 Advanced and Adaptive Approaches

Recent research has focused on overcoming these limitations by proposing more intelligent and contextual solutions.

- **Continuous Learning:** To combat the obsolescence of models against new threats, Ispahany et al. proposed **iCNN-LSTM+** [7], a system based on incremental learning that continuously updates itself from Sysmon log streams. This approach ensures that the model remains relevant over time.
- **Context Specificity:** Other works have shown that performance can be maximized by specializing in a specific environment. **DeftPunk** [5], for example, achieves a recall of nearly 100% by operating at the block storage level in cloud infrastructures, an approach that is highly effective but not generalizable to a standard workstation.
- **Semantic Understanding:** The most cutting-edge approach, **SRDC** [9], uses Large Language Models (LLMs) pre-trained on a cybersecurity corpus to under-

stand the semantic "intent" behind API call sequences. This method has shown unprecedented generalization capability for detecting *zero-day* threats.

These state-of-the-art approaches, while highly performant, often involve significant trade-offs, whether in terms of environmental specificity, implementation complexity, or extremely high computational resource costs.

2.4 Positioning Our Contribution

This review of the state-of-the-art reveals a research gap: there is a need for a solution that is both **robust** (based on dynamic behavioral analysis), **generalizable** to different environments (unlike overly specific solutions), and that achieves an **optimal balance between detection performance and reliability** (high recall and high precision).

Our work aims to fill this gap by proposing a novel hybrid architecture. We combine a Deep Learning model (CNN-LSTM) not as an end-to-end classifier, but as a powerful temporal feature extractor. These features are then fused with raw statistical information to feed an XGBoost classifier, renowned for its excellence on structured data. The key contribution of our approach lies in the integration of a metaheuristic optimization step (PSO) to rigorously tune the hyperparameters of this final classifier, in order to systematically find the best possible trade-off between different performance metrics.

3 Proposed Approach: An Optimized Hybrid Model

Facing the challenges identified in the state of the art, notably the trade-off between detection performance and reliability, as well as the need for generalization across different environments, we propose a hybrid, optimized, and end-to-end ransomware detection architecture. Our approach is distinguished by a clear separation of responsibilities between deep feature extraction and classification, along with a metaheuristic optimization phase to refine the decision-making process.

3.1 Overall Architecture and Data Pipeline

Our system is designed as a sequential processing pipeline that transforms raw I/O traces into a binary prediction ("Benign" or "Ransomware"). Figure 1 illustrates this workflow, which highlights our feature fusion approach.

The design philosophy is based on an iterative process. Our initial experiments with a monolithic Deep Learning model revealed suboptimal performance. This finding led us to develop this hybrid architecture, where a second-level classifier benefits from both the semantic representations learned by the deep model and raw statistical information from the data, thereby maximizing the richness of information available for decision-making.

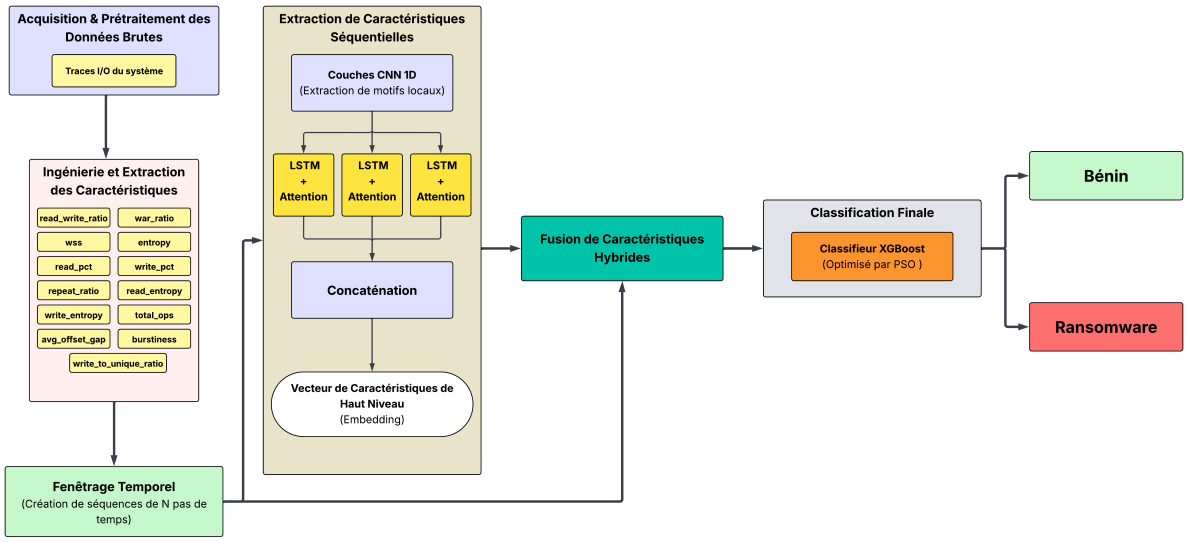


Figure 1: Overall architecture of the proposed ransomware detection pipeline, showing the feature fusion for the hybrid classifier.

The pipeline is broken down as follows:

1. **Data Preprocessing:** Raw I/O traces are first cleaned, then segmented into 5-second time windows. For each window, our set of 13 features is calculated.

2. **Deep Feature Extraction:** The sequences of windows are passed through our CNN-LSTM-Attention module, which acts as a feature extractor.

3. **Hybrid Fusion:** The output of the deep module is combined with statistics calculated on the raw sequence to form a hybrid feature vector.

4. **Optimized Classification:** This hybrid vector is finally classified by our PSO-optimized XGBoost model.

3.2 Deep Feature Extraction (CNN-LSTM-Attention)

This first module is the core of our behavioral analysis. Its purpose is not to produce a final classification, but to transform a complex I/O sequence into a semantic representation vector.

- **Input:** The module takes as input a matrix of shape ‘(50, 13)‘, representing 50 time steps (5s windows) described by 13 normalized features.

- **CNN Block:** A series of 1D convolutional layers acts as a local pattern detector, identifying suspicious "micro-behaviors" over short durations.

- **Parallel LSTM Blocks:** A key innovation in our architecture is the use of three independent LSTM blocks that process the output of the CNN layers in parallel. This allows the model to simultaneously capture different facets of the temporal dependencies.

- **Attention Mechanism:** Each LSTM output is weighted by an attention mechanism, which allows the model to focus on the most relevant moments in the sequence.

- **Output:** The outputs from the attention mechanisms are concatenated and passed through an MLP to produce an initial prediction probability. This probability will be used as a semantic feature for the next stage.

3.3 Classification via Feature Fusion and XGBoost

Rather than using the deep model's output as the final decision, we use it as one of the inputs to a more robust classifier for structured data. The final classifier, an **XGBoost** model, is trained on a **hybrid feature vector** that merges two types of information:

- **The primary model's prediction:** The malware probability from the CNN-LSTM-Attention module. This feature represents the deep network's "understanding" of the sequence.
- **Raw sequence statistics:** A set of descriptive statistics (mean, standard deviation, min, max, etc.) calculated directly on the 50x13 input sequence.

This fusion allows XGBoost to make a more informed final decision by correlating the deep model's semantic analysis with the fundamental statistical properties of the data, which significantly improves precision.

3.4 Hyperparameter Optimization via PSO

To ensure the XGBoost classifier operates at its full potential, its numerous hyperparameters must be optimized. We automated this complex process using the **Particle Swarm Optimization (PSO)** algorithm.

PSO explores the hyperparameter search space to find the combination that maximizes a custom **performance score**. This score was specifically designed to meet the demands of ransomware detection, where the trade-off between recall and precision is critical. We define this performance score, $\mathcal{S}$, as follows:

$$\mathcal{S} = (0.7 \times \text{F2-score}) + (0.3 \times \text{Precision}) \tag{1}$$

By assigning a majority weight (70%) to the **F2-score**, we prioritize threat detection (recall), while allocating a 30% weight to **Precision** to ensure the model maintains a low false alert rate.

Since optimization algorithms like PSO are typically designed to minimize a function, we transformed our problem of maximizing the score $\mathcal{S}$ into a problem of minimizing a **fitness function**, $\mathcal{F}$. This function is simply:

$$\mathcal{F} = 1 - \mathcal{S} \tag{2}$$

The objective of the PSO is therefore to find the hyperparameter combination that **minimizes $\mathcal{F}$**, which is equivalent to **maximizing our performance score $\mathcal{S}$**. This process, although computationally expensive, is performed once offline and ensures that the final classifier is tuned for the best possible performance/reliability trade-off.

4 Experiments and Results

This section details the experimental framework established to validate our approach. We present the datasets and evaluation metrics used, describe the protocol of our iterative experiments, and analyze the results obtained at each stage, from our baseline model to the final optimized hybrid architecture.

4.1 Datasets and Evaluation Metrics

4.1.1 Datasets

To evaluate the robustness and generalization capability of our model, we used two datasets with distinct characteristics:

- **DeftPunk Dataset [5]:** A large-scale dataset from a cloud computing environment, highly imbalanced with only 7% ransomware samples. It is ideal for testing performance under realistic and noisy conditions.
- **RansMap2024 Dataset [6]:** A public dataset of memory and storage access patterns, simulating a classic workstation environment. In contrast to the first, this dataset is balanced, with a ransomware proportion of approximately 52.4%, allowing us to assess the intrinsic performance of the classifier without the bias induced by class imbalance.

4.1.2 Evaluation Metrics

Given the imbalanced nature of the problem, we used a set of complementary metrics:

- **Accuracy:** For an overall measure of performance.
- **Precision:** To measure the reliability of positive alerts ($TP/(TP + FP)$).
- **Recall:** To measure the ability to detect all real attacks ($TP/(TP + FN)$). This is a critical metric in our context.
- **F2-Score:** A harmonic mean that weights recall twice as much as precision ($F_2 = 5 \cdot \frac{\text{Precision} \cdot \text{Recall}}{4 \cdot \text{Precision} + \text{Recall}}$). This is our main performance indicator for optimization.
- **AUC (Area Under the ROC Curve):** To evaluate the overall discrimination capability of the model, regardless of the classification threshold.

4.2 Experimental Protocol

Our protocol followed an iterative process in several steps, focusing on the 5-second time window, which proved to be the most effective in preliminary tests. For each step, the data was split into training (60%), validation (20%), and test (20%) sets.

1. **Baseline:** Evaluation of the base CNN-LSTM model on a restricted set of 4 features.
2. **Enrichment:** Evaluation of the same model on the full set of 13 features.

3. **Hybrid Model:** Evaluation of the hybrid CNN-LSTM + XGBoost model (with default hyperparameters).

4. **Final Optimization:** Evaluation of the hybrid model after optimizing XGBoost's hyperparameters via PSO.

5. **Generalization Validation:** Repetition of the full training and evaluation pipeline on the second dataset (RansMap2024).

4.3 Performance Analysis and Comparisons

4.3.1 From Baseline to Hybrid Model

Table 1 shows the evolution of performance on the DeftPunk dataset through the different design stages.

Table 1: Performance evolution on the DeftPunk dataset (5s window).

Model / Step	Acc.	Prec.	Recall	F2	AUC
1. Baseline (4 feat.)	70.95	28.56	81.50	59.46	84.39
2. Enriched (13 feat.)	75.00	32.00	80.00	61.00	86.00
3. Hybrid (XGBoost)	90.43	63.84	61.79	62.45	92.45
4. Hybrid + PSO	**92.87**	**70.75**	**77.41**	**75.97**	**95.95**

A clear and logical progression is observed. The transition from 4 to 13 features (Step 2) brought a modest improvement, confirming that the base model struggled to fully exploit this information. The switch to the **hybrid model (Step 3) caused a dramatic jump in precision** (from 32% to 63.8%), but at the expense of recall. Finally, **PSO optimization (Step 4) resolved this trade-off**, increasing both precision (+6.91 points) and recall (+15.62 points), leading to a massive improvement in the F2-score (+13.52 points).

4.3.2 Validation on the RansMap2024 Dataset

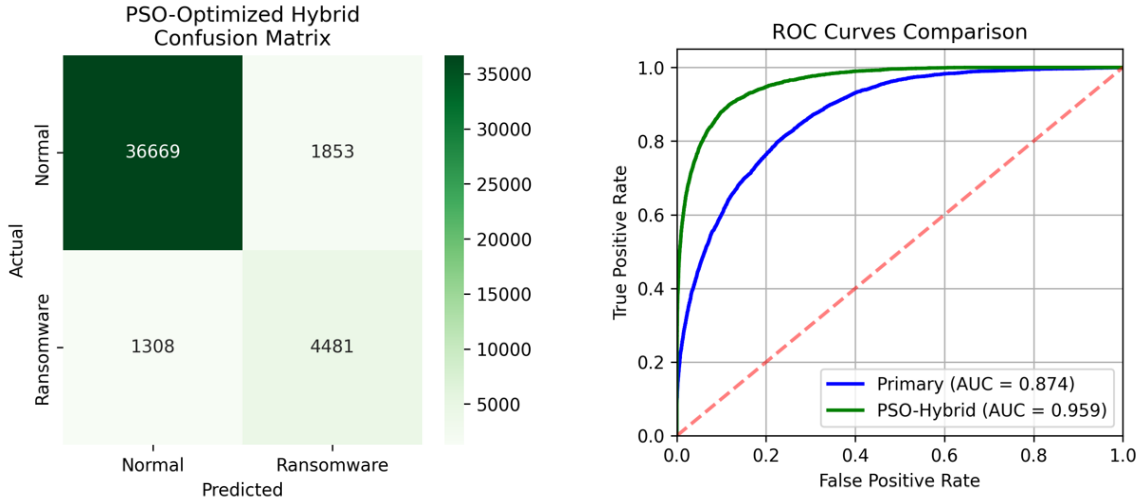
To confirm the robustness of our final approach, we applied it to the balanced RansMap2024 dataset. The results are presented in Table 2.

Table 2: Performance of the final model on the balanced RansMap2024 dataset (5s window).

Model	Acc.	Prec.	Recall	F2	AUC
Base (CNN-LSTM)	81.49	76.46	92.38	88.69	92.95
Hybrid + PSO	**84.83**	**80.22**	**93.49**	**90.50**	**94.60**

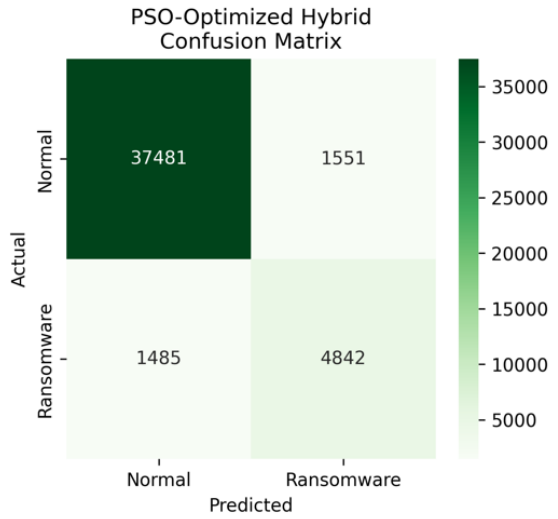
The results confirm the superiority of the optimized hybrid architecture. On this cleaner and balanced dataset, the model achieves excellent performance, with an **F2-score of 90.50%** and a **recall of 93.49%**. The fact that the same pipeline outperforms the base model in two such different contexts is strong validation of its generalization capability.

The figures 2 and 3 allow for the visualization and comparison of the final model's performance on both datasets. On the more challenging DeftPunk dataset, the ROC curve (AUC of 95.95%) shows excellent discrimination capability. On the RansMap2024 dataset, the performance is even better, as evidenced by the confusion matrix which shows a very high number of true positives and few errors. This dual visual validation reinforces the credibility and robustness of the proposed approach.

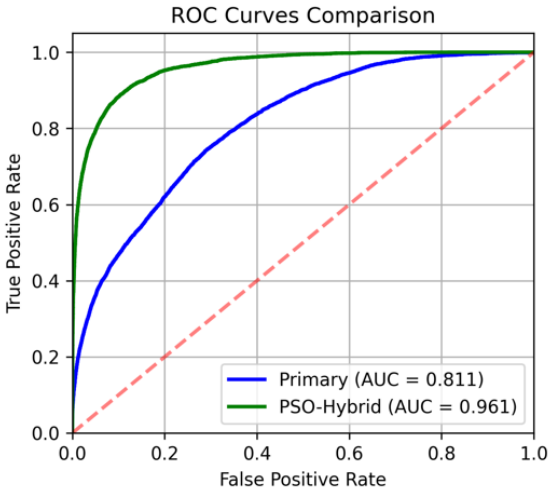


(a) Confusion Matrix (DeftPunk). (b) ROC Curves (DeftPunk).

Figure 2: Visualization of the final model's performance on the **DeftPunk** dataset, showing excellent class separation even in an imbalanced context.



(a) Confusion Matrix (RansMap).



(b) ROC Curves (RansMap).

Figure 3: Visualization of the final model's performance on the **RansMap2024** dataset, confirming its high performance in a balanced environment.

5 Discussion

The experimental results presented in the previous section not only demonstrate the performance of our hybrid architecture but also raise important discussion points regarding its design, its position relative to the state of the art, and its limitations. This section aims to interpret these results in depth.

5.1 Interpretation of Performance and Contributions

The success of our final model can be attributed to a synergy between three key design choices: the hybrid architecture, feature fusion, and metaheuristic optimization.

Superiority of the Hybrid Architecture. Our results show a spectacular performance gap across the different stages of our design. Introducing the hybrid architecture marked a significant first improvement: while the base model (CNN-LSTM on 13 features) achieved an F2-score of 61.00% on the DeftPunk dataset, simply adding an unoptimized XGBoost classifier raised this score to 62.45%. However, the real breakthrough came from optimization, with the final hybrid model reaching **75.97%**. This overall improvement confirms a fundamental hypothesis: although deep neural networks are unparalleled temporal feature extractors, they are not necessarily the most effective classifiers for fine-grained discrimination tasks. By delegating the final decision to a specialized model like XGBoost and rigorously optimizing it, we were able to leverage its superior ability to define complex decision boundaries in the feature space, a strength recognized in many applications on tabular data [1].

Effectiveness of Feature Fusion. The key to the performance of our XGBoost stage lies in the nature of the data it receives. The classifier's input vector is not just the semantic output of the CNN-LSTM; it is enriched with raw statistics (mean, std, etc.)

calculated directly on the I/O sequence. This **feature fusion** is a major methodological contribution. It allows the classifier to make its final decision based on two complementary views of the behavior: an "interpreted," high-level view from the deep network, and a "raw," statistical view. This duality provides increased robustness, allowing the model to capture signals that either approach alone might have missed.

Impact of Targeted PSO Optimization. The PSO optimization was not a simple fine-tuning step; it was a core design stage. By defining a fitness function that explicitly weights the F2-score and Precision, we were able to guide the search towards solutions that resolve the central **Precision-Recall trade-off**. The result is a model that not only detects more threats (recall improves or is maintained) but does so with much greater reliability (precision increases massively).

5.2 Comparison with the State of the Art

To better situate our contribution, it is useful to compare our approach with the reference models from the state of the art.

- **Compared to Static Approaches (e.g., Moreira et al. [3]):** The binary visualization approach achieves very high precision (98.20%) on balanced data. Our model, with a precision of 80.22% on a similar dataset, is less performant on this single metric. However, our dynamic approach is conceptually more robust against evasion techniques that modify runtime behavior without fundamentally altering the binary's overall structure (such as the use of packers or code encryption). Our model analyzes what the malware *does*, not just what it *looks like*.

- **Compared to Classic Behavioral Approaches (e.g., UNVEIL [2]):** UNVEIL achieved an impressive score of zero false positives. However, it relies on specific heuristics (entropy, image similarity) that can be bypassed by informed attackers (e.g., low-entropy ransomware). Our approach, by learning more complex patterns from 13 features, is potentially more resilient to such targeted evasion techniques. Moreover, our validation on a modern cloud dataset (DeftPunk) demonstrates its relevance for current environments.

- **Compared to Adaptive Approaches (e.g., iCNN-LSTM+ [7]):** The main strength of iCNN-LSTM+ is its incremental learning, which allows it to adapt to new threats. This is a limitation of our current model. However, our optimized hybrid architecture has shown superior performance in a static context. Feature fusion and PSO optimization could be integrated into a future incremental learning system to potentially surpass both approaches.

- **Compared to Semantic Approaches (e.g., SRDC [9]):** SRDC excels at *zero-day* detection thanks to the injection of external knowledge via an LLM. Our model, lacking this external knowledge, is likely less performant on this specific point. However, our approach is much more lightweight and pragmatic. It does not require the massive computational resources needed for the pre-training and inference of an LLM, making it much easier to deploy in practice.

In summary, our contribution lies in a **pragmatic, robust, and high-performing** solution. It may not be the best on every single axis (it is not incremental, nor semantic in

the LLM sense), but it represents an **excellent balance** between detection performance, generalization to different environments, and feasibility of deployment.

5.3 Limitations of the Study

Despite its performance, our work has several limitations that are important to acknowledge.

- **Static Nature of the Model:** Our main limitation is that the model is trained offline and is not designed to continuously adapt to *concept drift*.
- **Dependence on Feature Engineering:** The system's performance is strongly conditioned by the relevance of our set of 13 features. It remains vulnerable to evasion techniques that would not leave a footprint on these specific metrics.
- **"Zero-Day" Detection Validation:** Although generalization between two datasets is a good indicator, we did not conduct a strict "zero-day" testing protocol, where an entire ransomware family is completely excluded from training.
- **Real-World Deployment Constraints:** Our study was conducted in a laboratory setting. A production deployment would introduce additional challenges not measured here, such as end-to-end inference latency and the resource consumption of the monitoring agent.

These limitations do not invalidate our approach but clearly outline its scope and highlight exciting avenues for future work.

6 Conclusion

The detection of modern ransomware requires solutions capable of balancing a high recall to miss no threats, and a high precision to avoid alert fatigue. Facing this challenge, this paper has presented a hybrid architecture designed to meet this dual imperative. Our approach has demonstrated that by using a deep CNN-LSTM model as a temporal feature extractor and entrusting the final classification to an XGBoost model, whose hyperparameters have been rigorously optimized by PSO, it is possible to significantly improve detection performance.

Our experiments, conducted on heterogeneous datasets, have validated the superiority and robustness of this approach. By achieving an F2-score of 90.50% and an AUC of 94.60% on a balanced dataset, our model positions itself as a high-performing and reliable solution. The main contribution of this work is therefore the demonstration that a feature fusion approach, combining the semantic power of Deep Learning with the discriminative robustness of optimized ensemble models, represents a very effective and pragmatic strategy. This work opens several promising avenues, notably the integration of incremental learning mechanisms to adapt to *concept drift* and the extension of our architecture for multi-class classification, thus constituting a solid foundation for future research in intelligent cyber defense.

Acknowledgments

The authors wish to express their deep gratitude to their thesis supervisor,

***Mr. Bendella Mohammed Salih**,*

as well as to their co-supervisor,

***Mr. Bensenane Hamdane**,*

for their invaluable guidance, wise advice, and constant support throughout this research project. Their expertise and feedback were essential to the design and completion of this work.

I would also like to dedicate this work to my dear family, for their unconditional love, patience, and encouragement, which have been my greatest source of motivation throughout this journey.

References

[1] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[2] Amin Kharaz et al. "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware". In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 757–772. ISBN: 978-1-931971-32-4.

[3] Caio C Moreira, Davi C Moreira, and Claudomiro S de Sales Jr. "Improving ransomware detection based on portable executable header using xception convolutional neural network". In: *Computers & Security* 130 (2023), p. 103265.

[4] Jannatul Ferdous et al. "AI-based Ransomware Detection A Comprehensive Review". In: *IEEE Access* (2024). DOI: 10.1109/ACCESS.2024.3461965.

[5] Zhongyu Wang et al. "Ransom Access Memories: Achieving Practical Ransomware Protection in Cloud with DeftPunk". In: *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*. 2024.

[6] Manabu Hirano and Ryo Kobayashi. "RanSMAP: Open dataset of Ransomware Storage and Memory Access Patterns for creating deep learning based ransomware detectors". In: *Computers & Security* 150 (2025), p. 104202.

[7] Jamil Ispahany et al. "iCNN-LSTM+: A Batch-Based Incremental Ransomware Detection System Using Sysmon". In: *IEEE Access* 13 (2025), pp. 87978–87998. DOI: 10.1109/ACCESS.2025.3569635.

[8] Lingbo Zhao et al. "ERW-Radar: An Adaptive Detection System against Evasive Ransomware by Contextual Behavior Detection and Fine-grained Content Analysis". In: *Network and Distributed System Security (NDSS) Symposium 2025*. Internet Society, 2025. ISBN: 979-8-9894372-8-3. DOI: 10.14722/ndss.2025.230349.

[9] Ce Zhou et al. "SRDC: Semantics-based Ransomware Detection and Classification with LLM-assisted Pre-training". In: *Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)*. 2025.