A Real-Time Cache Side-Channel Attack Detection and Mitigation Framework Based on Machine Learning 2024

Youssra Ghabbara[1], Zied Trifa[1]

[1] Multimedia, Information Systems and Advanced Computing Laboratory (MIRACL)

University of Sfax, University of Gabes, Tunisia

Contact emails: ghabbarayoussra@gmail.com, trifa.zied@gmail.com

Author

Youssra Ghabbara received the master's degree in computer science from the University of Gabes, Tunisia in 2024.

Her research interest lies in the intersection of artificial intelligence (machine learning , deep learning), cloud computing and cloud security.

1. Aims and contributions of our paper

In our paper, we aimed at:

1. Develop a real-time detection framework for cache side-channel attacks (CSCAs) in cloud environments.
2. Minimize system overhead while maintaining high detection accuracy.

Contributions of our study are fourfold:

1. Unified ML model detecting four CSCAs (Flush+Reload, Flush+Flush, Prime+Probe, Spectre).
2. Feature engineering using Greedy Forward Selection and Pearson Correlation.
3. Hybrid ensemble model (Random Forest + XGBoost) with 96% accuracy and 11% false alarm rate.
4. Intelligent Noise Addition mitigation strategy to reduce data leakage.

2. Motivation

This table highlights the key limitations of existing security approaches, including high false alarms, inefficiency under load, and no real-time detection.

Approach	Limitations
HPC-based models [1][2]	High false alarms, limited to specific attacks.
AES encryption [3]	Ineffective under load conditions.
Unsupervised Deep Learning [4]	No real-time detection.

These challenges emphasize the need for a more robust, efficient, and real-time detection solution leading to our proposed model :

→ A unified real time detection and mitigation model for multiple CSCAs with minimal HPCs .

3.1 Methodology : Feature Engineering

1. Data Collection:

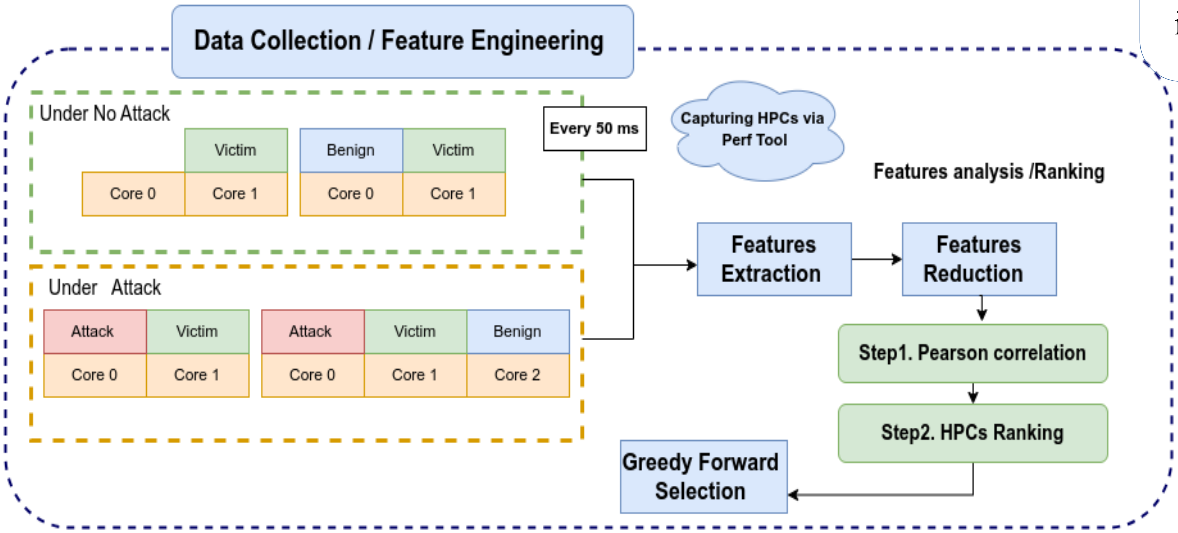
- Hardware Performance Counters (HPCs) sampled at 50µs intervals.
- Benchmarks: Mastik [5], Xlate [6] (attacks), MiBench [7] (benign apps).

2. Feature Selection:

- Greedy Forward Selection [8]: Incremental addition of features for optimal accuracy.
- Pearson Correlation [9] $\rho(i) = \frac{\text{cov}(X_i, Y)}{\sqrt{\text{var}(X_i) \cdot \text{var}(Y)}}$

Retain features with $|\rho| > 0.4$.

ρ: is the Pearson correlation coefficient.
X_i: input dataset of event i (where i = 1, ... , 14).
Y: is the output dataset containing labels, i.e., “attack” or “no attack” in this case.



3.1 Methodology : Feature Engineering

3. Customized Features

- Reduced 14 HPCs to 4 critical metrics for real-time efficiency.

Original HPCs		Customized HPCs
1 Cache-references, 2 Cache misses, 3 Cpu_cycles, 4 Instructions, 5 Branches, 6 Branch-misses, 7 L1-dcache-load-misses,	8 L1-icache-load-misses, 9 LLC-misses, 10 ITLB-load-misses, 11 LLC-store-misses, 12 LLC-loads, 13 DTLB-load-misses, 14 Branch-instructions	1 LLC-misses, 2 Instructions, 3 Branches, 4 Branch-instructions

To derive the final dataset containing four key metrics, we followed a structured approach consisting of three essential stages:

Each step was designed to refine and enhance the dataset, ensuring the most relevant and insightful metrics were retained.

3.2 Methodology : Attack Detection

The algorithms used are :

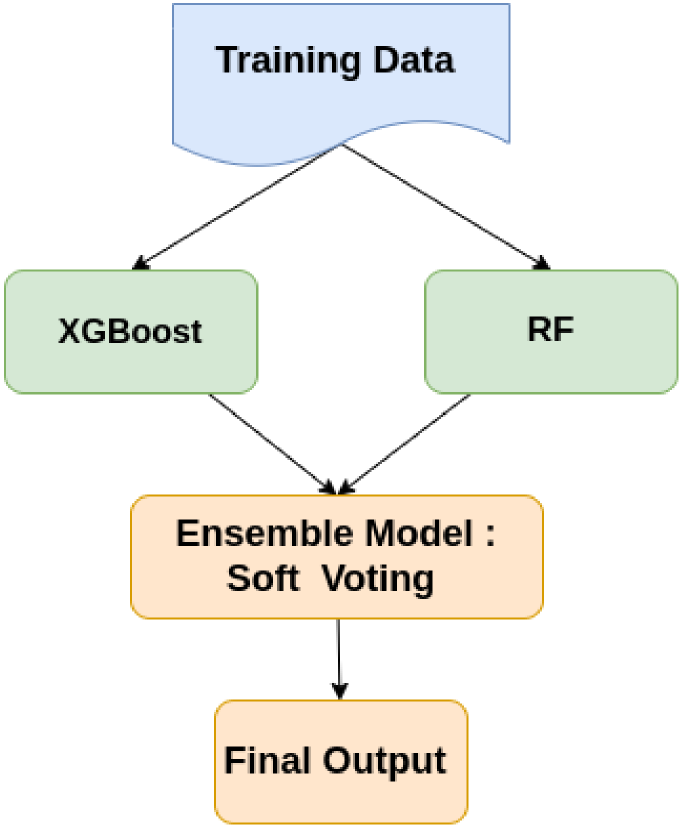
1. Random Forest (RF) [10]: 92% accuracy, 24% FAR.

2. XGBoost [11]: 93% accuracy, 24% FAR.

3. Hybrid Model (RF + XGBoost):

   - Soft voting mechanism.

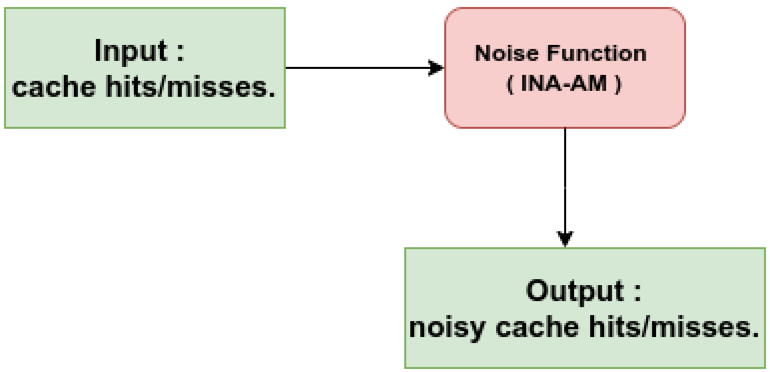
   - 96% accuracy, 11% FAR.



3.3 Methodology : Attack Mitigation

For mitigation we opted :

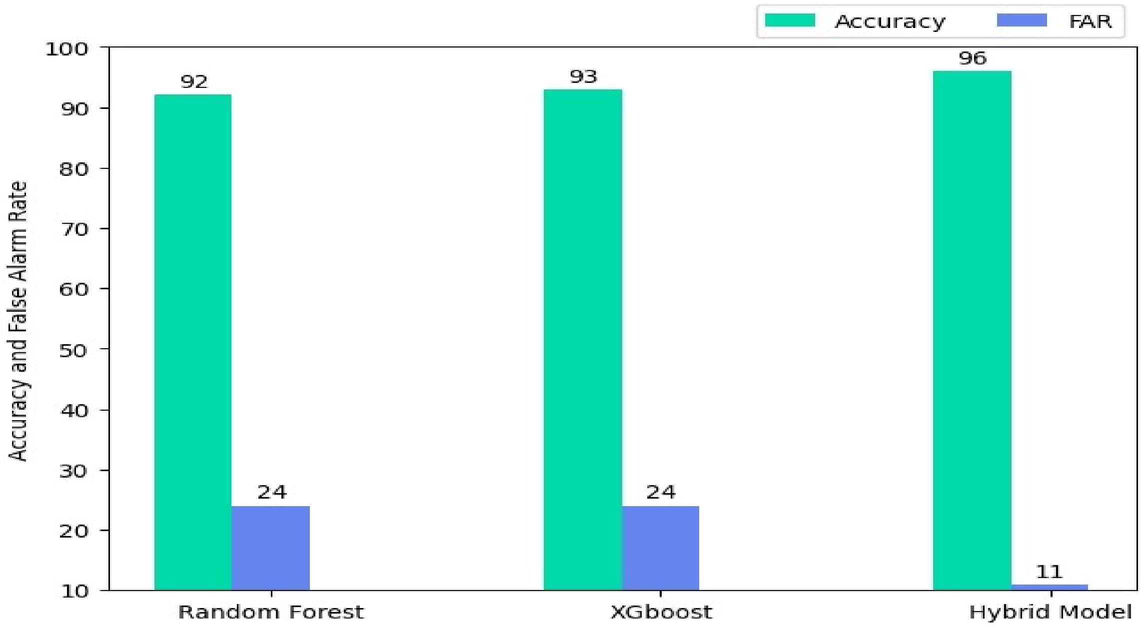
- Intelligent Noise Addition (INA-AM) [12]: confuses attackers by injecting noise into cache hits/misses.



Algorithm: Intelligent Noise Addition for Attack Mitigation (INA-AM)
Input: Cache Hits H, Cache Misses M
Output: Noise Cache Hits H', Noise Cache Misses M'

```
Start
Initialize noisy cache hits vector H'
Initialize noisy cache misses vector M'
hcount ← CountCacheHits(H)
mcount ← CountCacheMisses(M)
noise function ← ComputeNoiseFunction(H, M)
For each cache hit h in H
IF the noise function recommends noise to h Then
Add noise to h
Add h to H'
End If
End For
For each cache miss m in M
IF the noise function recommends noise to m Then
Add noise to m
Add h to M'
End If
End For
Output H'
Output M'
End
```

4.1 Results : Model Evaluation

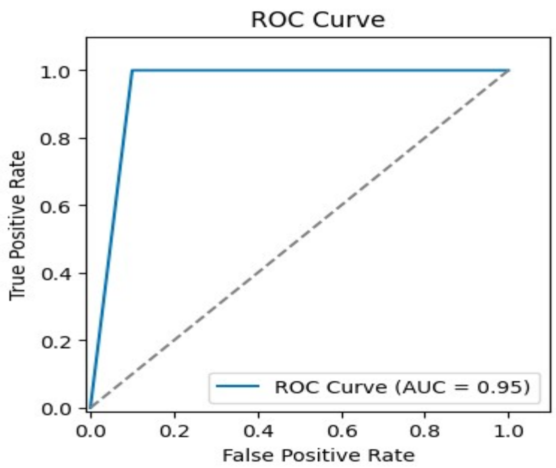


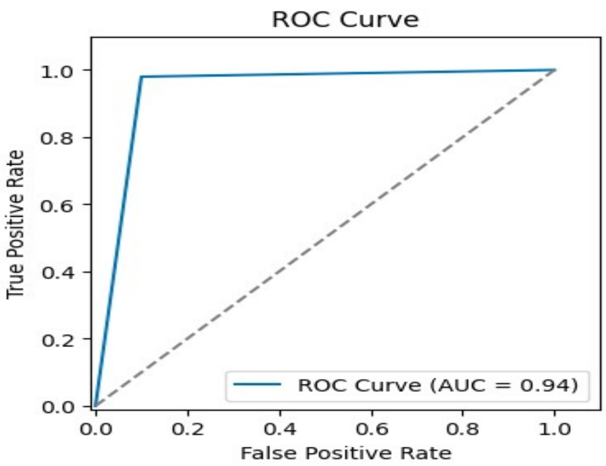
Accuracy & FAR Comparison:

- Hybrid model outperforms individual classifiers (96% accuracy vs. 92-93%).
- FAR reduced from 24% to 11%.

4.2 **Results : Model Evaluation**

ROC Curves:





- AUC = 0.95 (No Load), 0.94 (Full Load).
- These two curves shows the model robust performance under varying system loads.

4. 3 Results : State-of-the-Art Comparison

- **Enhanced Detection:** The proposed model detects threats using 4 CSCAs + stealth, outperforming existing methods.
- **Optimized Efficiency:** It requires just 4 HPCs, compared to 11-12 HPCs in existing approaches.
- **Superior Security:** Achieves 0 bits of information leakage, whereas existing methods leak 189-345 bits.

Metric	Proposed Model	Existing Methods
Detection Range	4 CSCAs + stealth	1-2 CSCAs
HPCs Used	4	11-12
Information Leakage	0 bits	189-345 bits

Overall Improvement: The model offers better detection, lower resource usage, and higher security, making it a more efficient and robust solution.

5. Conclusion and Future Work

Conclusion:

- We developed a unified machine learning model to detect four CSCAs: Flush+Reload, Flush+Flush, Prime+Probe, and Spectre.
- We optimized feature selection using Greedy Forward Selection and Pearson Correlation to enhance detection efficiency.
- We achieved 96% accuracy with an 11% false alarm rate using a hybrid ensemble model (Random Forest + XGBoost).
- We implemented Intelligent Noise Addition to mitigate data leakage and improve security.

Future Work:

1. Integrate deep learning for adaptive threat detection.
2. Scalability testing across diverse cloud architectures.
3. Long-term system maintenance strategies.

References

[1] M. Mushtaq et al., “Nights-watch: A cache-based side-channel intrusion detector using hardware performance counters”, in Proc. 7th International Workshop on Hardware and Architectural Support for Security and Privacy, 2018.

[2]T. Zhang, Y. Zhang, and R. B. Lee, “Cloudradar: A real-time side-channel attack detection system in clouds”, in Proc.International Symposium on Recent Advances in Intrusion Detection, 2016.

[3]N. H. Ali, M. E. Abdulmunem, and A. E. Ali, “Learning evolution: A survey”, Iraqi Journal of Science, 2021.

[4]M.-M. Bazm, T. Sautereau, M. Lacoste, M. Südholt, and J.-M.Menaud, “Cache-based side-channel attacks detection through intel cache monitoring technology and hardware performance counters”, in Proc. 2018 Third International Conference on Fog and Mobile Edge Computing(FMEC), pp. 7–12.

[5]Y. Yarom, Mastik: A micro-architectural side-channel toolkit, Available: http://cs.adelaide.edu.au/~yval/Mastik,2016.

[6] M. Chiappetta, E. Savas, and C. Yilmaz, Xlate, Available: https://www.vusec.net/projects/xlate/, 2016.

[7]M. R. Guthaus et al., “Mibench: A free, commercially representative embedded benchmark suite”, in Proc. Fourth Annual IEEE International Workshop on Workload Characterization (WWC-4), 2001, pp. 3–14.

[8] T. Pahikkala, A. Airola, and T. Salakoski, “Speeding up greedy forward selection for regularized least-squares”, in Proc. 2010 Ninth International Conference on Machine Learning and Applications, 2010, pp. 325–330.

[9] A. R. M. S, N. C. R, C. B. B, M. Rafi, and S. B. R, “Online feature selection using pearson correlation technique”, in Proc. 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), vol. 7,2022, pp. 172–177.

[10]L. Breiman, “Random forests”, Machine Learning, vol. 45,pp. 5–32, 2001.

References

[11]D. Van, Ensemble methods: Foundations and algorithms, 2012.
[12]S. Mahipal, V. Ceronmani, and V. C. Sharmila, “A security framework for improving qos by detecting and mitigating cache side-channel attacks in virtualized environments”, 2023.
[13] M. S. Inci, B. Gülmezoglu, G. I. Apecechea, T. Eisenbarth, and B. Sunar, “Seriously, get off my cloud! cross-VM RSA key recovery in a public cloud”, IACR Cryptol. ePrint Arch.,vol. 2015, p. 898, 2015.
[14]Y. Yarom and K. E. Falkner, “FLUSH+RELOAD: A high resolution, low noise, L3 cache side-channel attack”, IACR Cryptol. ePrint Arch., vol. 2013, p. 448, 2014.
[15] D. Gruss, C. Maurice, K. Wagner, and S. Mangard,“Flush+flush: A fast and stealthy cache attack”, in Proc.International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 2015.
[16] G. I. Apecechea, T. Eisenbarth, and B. Sunar, “S$a: A shared cache attack that works across cores and defies vm sandboxing and its application to aes”, in Proc. 2015 IEEE Symposium on Security and Privacy, 2015, pp. 591–604.
[17] P. C. Kocher, D. Genkin, D. Gruss, and et al., “Spectre attacks: Exploiting speculative execution”, in 2019 IEEE Symposium on Security and Privacy (SP), 2018, pp. 1–19.
[18] C. Li and J.-L. Gaudiot, “Online detection of spectre attacks using microarchitectural traces from performance counters”, in Proc. 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), 2018, pp. 25–28.
[19] Z. Allaf, M. Adda, and A. E. Gegov, “A comparison study on flush+reload and prime+probe attacks on aes using machine learning approaches”, in Proc. UK Workshop on Computational Intelligence, 2017.
[20] D. Page, “Theoretical use of cache memory as a cryptanalytic side-channel”, IACR Cryptol. ePrint Arch., vol. 2002, p. 169, 2002.

References

[21] X.-T. Yuan and S. Yan, “Forward basis selection for sparse approximation over dictionary”, in Proc. International Conference on Artificial Intelligence and Statistics, 2012.
[22] H. Sayadi, N. Patel, A. Sasan, and H. Homayoun, “Machine learning-based approaches for energy-efficiency prediction and scheduling in composite cores architectures”, in Proc. 2017 IEEE International Conference on Computer Design (ICCD), 2017, pp. 129–136.
[23] A. R. M. S, N. C. R, C. B. B, M. Rafi, and S. B. R, “Online feature selection using pearson correlation technique”, in Proc.2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), vol. 7, 2022, pp. 172–177.
[24] D. Van, Ensemble methods: Foundations and algorithms, 2012.
[25] M. Mushtaq et al., “Run-time detection of prime + probe side-channel attack on aes encryption algorithm”, in Proc. 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1–5.