

Machine Learning Classifiers for Hardware Trojan Detection: A Comparative Study of Random Forest and One-Class SVM

Alexandria Nikirk, Jingze Fu, Sean Weller
Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL, USA
anikirk@ufl.edu, jingze.fu@ufl.edu, sweller1@ufl.edu

Abstract—Hardware Trojans are difficult to detect due to process variations and normal switching activity, making traditional threshold methods unreliable. This work compares Random Forest and One-Class SVM classifiers using ring oscillator frequency data, showing their complementary strengths across supervised and anomaly detection scenarios.

Keywords—Hardware Trojan detection, Random Forest, One-Class SVM, supervised learning, anomaly detection

I. INTRODUCTION

Hardware Trojans represent a growing threat in modern integrated circuits, as malicious modifications can be inserted during fabrication without the designer’s knowledge. Detecting such Trojans is inherently difficult because their size, type, location, trigger, and payload are unknown, and the complexity of chip logic makes exhaustive verification impractical. Even small modifications can compromise system security, reliability, or performance, underscoring the need for effective detection mechanisms. Consequently, researchers have increasingly turned to machine learning techniques as scalable tools for distinguishing Trojan-infected chips from Trojan-free ones.

One promising class of detection techniques leverages variations in power consumption. Since all chip activity consumes power, unexpected fluctuations may reveal Trojan activity. In [1], Kelly et al. proposed using ring oscillators (ROs) distributed across a chip to monitor the power supply network and detect Trojan activity. RO frequencies are influenced by runtime activity, manufacturing process variations, and environmental factors such as temperature.

When a Trojan switches, it draws localized power, causing temporary voltage dips that propagate across the network. These dips manifest as measurable frequency shifts in nearby ROs, providing a potential signal for detection. However, practical deployment is complicated by natural process variations and normal switching activity, which can produce frequency changes as large as or larger than those caused by Trojans. As a result, simple threshold-based detection is unreliable, motivating the use of machine learning classifiers to distinguish Trojan-infected chips from Trojan-free ones.

The objective of this project is to design and evaluate machine learning-based classifiers for hardware Trojan detection using RO frequency data. We consider multiple detection scenarios that reflect both ideal laboratory conditions and real-world constraints. In Case 1, labeled data from both Golden (Trojan-free) and Trojan-inserted chips are available for training, enabling supervised learning. In Case 2, only Golden samples are available, requiring anomaly detection methods. By implementing and comparing classifiers across these two cases, we aim to assess their accuracy, sensitivity, and practicality under varying assumptions about data availability. This study highlights the trade-offs between supervised and unsupervised approaches and provides insights into how engineers can design robust detection frameworks that remain effective across diverse deployment settings. Ultimately, the findings underscore the importance of aligning classifier choice with data availability to ensure reliable and scalable Trojan detection in practice.

II. CLASSIFIER SELECTION AND DESIGN

In this project we focused on two detection scenarios: Case 1, where both Golden (Trojan-free) and Trojan-inserted samples are available for training, and Case 2, where only Golden samples can be used. These two cases reflect very different realities. Case 1 is the “ideal lab” situation, where we have labeled examples of both classes. Case 2 is closer to what engineers face in practice, since it is often impossible to collect Trojan-inserted data ahead of time. To cover both, we used two classifiers: Random Forest (RF) for supervised learning, and One-Class Support Vector Machine (OCSVM) for anomaly detection. This pairing allowed us to directly compare the strengths of a supervised model against an unsupervised anomaly detector under the same experimental conditions. By designing the study this way, we ensured that our evaluation framework captured both theoretical best-case performance and practical real-world constraints.

The Random Forest was chosen because it is a reliable workhorse for supervised classification. It builds an ensemble of decision trees, each trained on random subsets of the data and features, then combines their votes to make a final

prediction. This randomness helps prevent overfitting and makes the model robust even when the dataset is relatively small, as in our case. We used 100 trees, which is a common balance between accuracy and runtime. RF’s strengths are clear: it handles nonlinear relationships well, does not require heavy parameter tuning, and gives interpretable outputs like feature importance. Importantly, RF can highlight which ring oscillator features contribute most to classification, offering engineers insight into the physical signatures of Trojan activity. This interpretability is valuable because it not only provides detection but also helps guide hardware designers toward understanding which parts of the chip are most vulnerable. However, RF requires labeled Trojan data to learn what “malicious” looks like, which limits its applicability in scenarios like Case 2 where only Golden data are available. In practice, this means RF is best viewed as a precision tool for controlled environments rather than a universal solution.

The One-Class SVM was selected for Case 2 because it is designed for situations where only one class is known. Instead of trying to separate Golden and Trojan samples directly, OCSVM learns the boundary around Golden data and flags anything outside that boundary as suspicious. We used the RBF kernel with automatic gamma scaling, which adapts to the scale of the dataset without manual tuning. OCSVM’s big advantage is that it does not need Trojan labels, making it practical for real-world deployment. On the flip side, it can be sensitive to parameter choices and tends to produce high false positive rates, especially when Golden data vary a lot between chips. In our experiments, OCSVM achieved very high detection rates (TPR close to 100%), but also misclassified many Golden samples as Trojans, which shows the trade-off between sensitivity and specificity.

In contrast to RF, which thrives when both classes are labeled, OCSVM fills the gap when only benign data are available, though at the cost of precision. This makes OCSVM a strong candidate for early warning systems, but one that requires careful calibration to avoid overwhelming engineers with false alarms. Another important consideration is scalability: OCSVM’s efficiency with runtimes under 0.01 seconds per trial suggests it could be integrated into lightweight monitoring systems where speed is critical, such as runtime chip validation or continuous security auditing.

To evaluate both classifiers fairly, we designed experiments that sampled training data in varying sizes (6, 12, or 24 depending on the trial), repeated the process 20 times, and averaged the outcomes across all 23 Trojan types. This repeated-trial methodology ensured that results were not biased by a single random split of the data, but instead reflected consistent performance across diverse Trojan scenarios.

The classifiers were chosen to reflect the strengths of both supervised and anomaly detection approaches. RF is the right choice when Trojan data are available, offering robust supervised learning. OCSVM is the practical option when only Golden data can be trusted, though it requires careful tuning to avoid excessive false positives. Together, they highlight the trade-offs engineers face in Trojan detection:

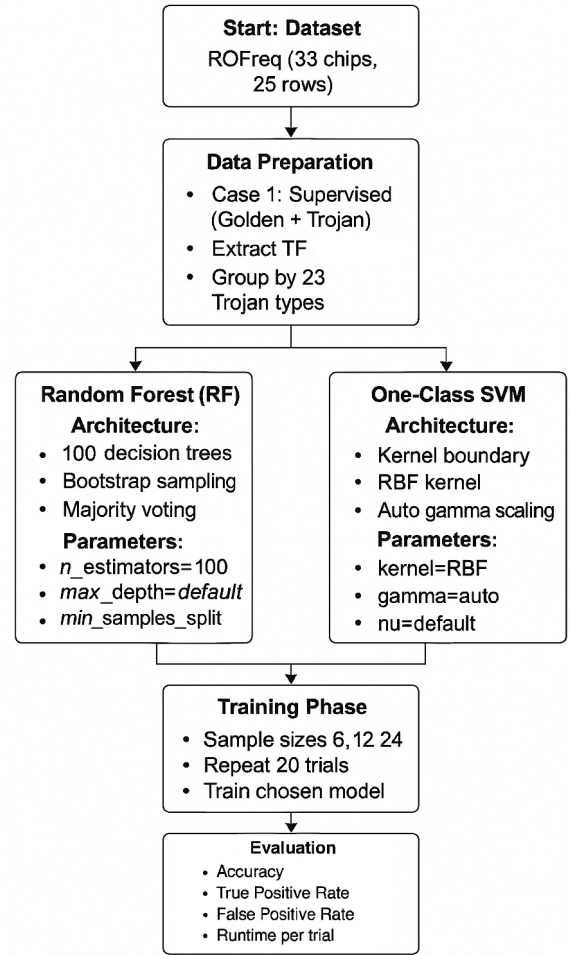


Fig. 1. Trojan Detection Workflow with Model Architecture and Parameters

balancing accuracy, sensitivity, and practicality depending on what data are available. A hybrid approach using OCSVM for initial screening and RF when Trojan samples are collected could combine the best of both worlds. This comparative framework underscores that no single model is universally optimal, but pairing them allows detection strategies to adapt to the realities of data availability. This dual-model perspective provides a roadmap for engineers to design detection systems that remain effective across both controlled laboratory settings and unpredictable real-world deployments.

Furthermore, the study demonstrates how methodological rigor including repeated trials, averaging across diverse Trojan types, and careful runtime measurement is essential for producing results that are both reproducible and meaningful for future research. By documenting both the strengths and weaknesses of RF and OCSVM, this work contributes not only to immediate detection strategies but also to the broader conversation about how machine learning can be responsibly applied to hardware security challenges.

TABLE I
CASE 1: CLASSIFIER PERFORMANCE COMPARISON (RF vs OCSVM)

Sample Size	Accuracy		TPR		FPR		Time (s)	
	RF	OCSVM	RF	OCSVM	RF	OCSVM	RF	OCSVM
6	90.71%	88.49%	97.92%	96.63%	84.24%	96.14%	0.08	0.005
12	90.48%	81.07%	98.44%	88.76%	83.56%	90.53%	0.10	0.008
24	89.28%	70.12%	98.38%	77.42%	75.76%	82.12%	0.12	0.010

TABLE II
CASE 2: CLASSIFIER PERFORMANCE COMPARISON (RF vs OCSVM)

Sample Size	Accuracy		TPR		FPR		Time (s)	
	RF	OCSVM	RF	OCSVM	RF	OCSVM	RF	OCSVM
6	8.00%	92.35%	0.00%	99.93%	0.00%	94.85%	0.075	0.0012
12	8.00%	92.48%	0.00%	99.68%	0.00%	90.30%	0.086	0.0009
24	8.00%	92.88%	0.00%	99.12%	0.00%	78.79%	0.092	0.0014

III. CLASSIFIER EVALUATION RESULT

In this section, we evaluate the performance of the machine learning classifiers on both Case 1 (supervised learning) and Case 2 (one-class learning). All experiments were performed using 20 randomized trials for each training size of 6, 12, and 24 samples. The reported performance metrics include accuracy, true positive rate (TPR), false positive rate (FPR), and average execution time per trial.

A. Case 1: Supervised Learning

Two classifiers were evaluated under Case 1: Random Forest (RF) and One-Class SVM used as a binary classifier (OCSVM). Both models were trained using a balanced subset of Golden and Trojan samples, and tested using all remaining samples.

1) *Random Forest (Case 1)*: Across all three training sizes, Random Forest consistently achieved high true positive rates near 98%, indicating strong capability to identify Trojan-inserted samples (as show in Table I.) With 6 training samples, RF obtained an accuracy of 90.71%, TPR of 97.92%, and a high FPR of 84.24%, showing that although most Trojans were detected, many Golden samples were incorrectly flagged as Trojan. With 12 training samples, the accuracy remained similar at 90.48%, with TPR rising slightly to 98.44% and FPR remaining high (83.56%). When using 24 training samples, RF achieved 89.28% accuracy, 98.38% TPR, and a somewhat reduced FPR of 75.76%, showing that increasing training size helps slightly lower the false alarm rate. Training time gradually increased from about 0.08 seconds at size 6 to 0.12 seconds at size 24.

Overall, RF is highly reliable in detecting Trojans (near-perfect TPR), but its FPR remains high across all training sizes due to the similarity between Golden and Trojan RO-frequency patterns.

2) *OCSVM as a Supervised Classifier (Case 1)*: Although OCSVM is not inherently designed for binary classification, it was evaluated for comparison under Case 1. With 6 training samples, OCSVM achieved 88.49% accuracy, 96.63% TPR,

but an extremely high FPR of 96.14%, indicating that nearly all Golden samples were misclassified as Trojan. As training size increased, performance degraded further in terms of accuracy: 81.07% at size 12 and 70.12% at size 24. TPR decreased to 88.76% and 77.42% at sizes 12 and 24, respectively, while FPR remained very high (above 82% in all cases).

Execution time for OCSVM was extremely low, between 0.005 and 0.01 seconds, but its classification behavior demonstrates that it is unsuitable for supervised Trojan detection on this dataset.

B. Case 2: One-Class Learning

Case 2 restricts training data to only Golden samples, requiring the classifier to identify Trojans as anomalies during testing. Two models were evaluated: RF and OCSVM.

1) *Random Forest (Case 2)*: RF completely failed under Case 2 because one-class learning is incompatible with supervised algorithms (see Table II for detailed metrics.) Since RF only observed Golden samples during training, it learned to classify every sample as Golden. As a result, accuracy across all training sizes remained extremely low at 8%, with TPR of 0% (no Trojans detected) and FPR of 0% (no Golden samples misclassified). Although execution times remained around 0.08–0.09 seconds, RF clearly cannot serve as a one-class classifier.

2) *OCSVM (Case 2)*: in contrast, OCSVM performed far more effectively for one-class detection. With 6 Golden training samples, OCSVM reached 92.35% accuracy, an extremely high TPR of 99.93%, but also a very high FPR of 94.85%. This indicates that the model is extremely sensitive to deviations and labels nearly all Trojans correctly, but also mislabels many Golden samples as Trojan.

As training size increased, OCSVM's performance remained consistent in accuracy while the false positive rate improved. With 12 training samples, the model achieved 92.48% accuracy, 99.68% TPR, and 90.30% FPR. With 24 training samples, OCSVM reached 92.88% accuracy, 99.12%

TABLE III
BEST PERFORMANCE PER CLASSIFIER IN CASE 1 AND CASE 2

Case	Classifier	Best Accuracy	Best TPR	Lowest FPR	Time (s)
Case 1	RF	90.71% (6)	98.44% (12)	75.76% (24)	0.12
	OCSVM	88.49% (6)	96.63% (6)	82.12% (24)	0.005
Case 2	RF	8.00% (all)	0.00% (all)	0.00% (all)	0.092
	OCSVM	92.88% (24)	>99% (6)	78.79% (24)	0.001

TPR, and a significantly improved 78.79% FPR, showing that larger Golden training sets help the model better understand normal variability. Execution time remained extremely low across all training sizes, between 0.0009–0.0014 seconds.

C. Comparative Discussion: Case 1 vs Case 2 Performance

Supervised learning (Case 1) consistently outperformed one-class learning when both Golden and Trojan samples were available during training. Random Forest in Case 1 provided highly reliable Trojan detection, with TPR values consistently near 98% (see Table I). Case 2, while capable of achieving high TPR through OCSVM, suffered from dramatically higher FPR, making it less practical without additional threshold tuning (see Table II).

Classifier Comparison: Random Forest is the strongest supervised classifier, achieving the best balance of accuracy and TPR, though FPR remains relatively high. OCSVM in Case 1 performs poorly as a binary classifier and should not be used in supervised settings. OCSVM in Case 2 is capable of detecting nearly all Trojans (TPR > 99%), but its FPR is high due to the sensitivity of one-class models to normal variation. Random Forest in Case 2 cannot operate as a one-class classifier and always fails. The best overall results are shown in Table III.

Effect of Training Size: Training size affects FPR more than accuracy or TPR. In Case 1, larger training sets slightly reduced FPR but did not significantly improve accuracy or TPR. In Case 2, larger Golden-only training sets helped OCSVM substantially reduce FPR (from ~95% down to ~79%). Execution time scaled modestly with training size and remained low for all classifiers.

IV. CONCLUSION AND PERSONAL COMMENTS

This study compared the performance of Random Forest (RF) and One-Class Support Vector Machine (OCSVM) classifiers for hardware Trojan detection using ring oscillator frequency data. The results demonstrate that supervised learning with RF achieves consistently high true positive rates (near 98%) when both Golden and Trojan samples are available, though false positive rates remain elevated due to overlapping frequency patterns. In contrast, OCSVM excels in one-class scenarios where only Golden data are available, achieving true positive rates above 99% but at the cost of high false positive rates. These findings highlight the trade-offs between supervised and anomaly detection approaches and suggest that hybrid frameworks such as using OCSVM for initial screening and RF for refined classification may

offer a practical balance between sensitivity and specificity. Ultimately, classifier choice must align with data availability to ensure reliable Trojan detection in both laboratory and real-world deployment settings.

Beyond the direct comparison of RF and OCSVM, this work emphasizes the broader lesson that no single classifier can universally solve the hardware Trojan detection problem. Instead, engineers must carefully evaluate the assumptions behind each method and adapt their detection strategies to the realities of deployment. Future research should explore hybrid models, threshold tuning, and ensemble approaches that combine the strengths of supervised and anomaly detection techniques. Additionally, expanding datasets to include more diverse Trojan types and process variations will be critical for improving generalization. By documenting both the successes and limitations of RF and OCSVM, this study contributes to the ongoing conversation about how machine learning can be responsibly applied to hardware security challenges.

Personal Comments: Preparing this work underscored the importance of methodological rigor and careful formatting in technical research. Iterating through classifier design and evaluation was not only a technical exercise but also a lesson in balancing clarity, reproducibility, and presentation standards. The process of troubleshooting L^AT_EX layouts, refining tables, and aligning author blocks reminded me that communication is as critical as computation in engineering research. While the classifiers provided valuable insights into detection trade-offs, the act of presenting these results in IEEE style reinforced the discipline required to make research accessible and professional. This project has strengthened both my technical understanding of machine learning in hardware security and my appreciation for the craft of academic writing.

Equally important was the collaborative nature of this project. As a three-person team, we divided the workload across research, coding, and paper preparation, which allowed each of us to contribute our strengths while learning from one another. This division of tasks fostered efficiency, but more importantly, it created an environment of shared growth where discussions about methodology, experimental design, and formatting led to deeper insights. Working together not only improved the quality of the final paper but also highlighted the value of teamwork in academic research, where collaboration and mutual support are essential for tackling complex technical challenges.

REFERENCES

- [1] Shane Kelly, Xuehui Zhang, Mohammed Tehranipoor, and Andrew Ferraiuolo, Detecting Hardware Trojans using On-chip Sensors in an ASIC Design. *Journal of Electronic Testing* 31, no. 1 (2015): 11-26.