

Performance Analysis for an Energy-Aware Adaptive Smart Contract Framework for Jamming-Resilient Cooperative Spectrum Sensing

Delores Baker^{1,*}, Dinh Nguyen¹, Chance Glenn², Laurie Joiner¹ and Nicholas Jones¹

¹ Department of Electrical and Computer Engineering, University of Alabama in Huntsville, Huntsville, AL 35899, USA; dbp0014@uah.edu

² Department of Electrical and Computer Engineering, Wilberforce University

Abstract - This paper presents a mathematically grounded framework for enhancing cooperative spectrum sensing in cognitive radio networks (CRNs) under jamming conditions using adaptive blockchain smart contracts. A binary hypothesis testing model with Gaussian energy detection is developed to characterize the impact of jamming-induced noise variance inflation on detector and fusion performance. We show that fixed decision logic leads to amplified false alarm rates under cooperative OR fusion during active interference. To address this limitation, we introduce an energy-aware adaptive smart contract that creates a secondary statistical decision boundary at the fusion layer. Analytical expressions are derived for the detection probability, false alarm probability, and override rate, demonstrating how the adaptive policy dynamically reshapes the contract-level ROC curve under jamming while naturally reverting to the baseline fusion behavior after interference subsides. Comparative simulations across pre-jamming, jamming, and post-jamming phases confirm that the proposed approach reduces contract-level false alarms without permanently degrading detection performance. By leveraging a prototype Solidity implementation for immutable sensing-event recording and smart-contract execution, this work demonstrates the practical feasibility of blockchain-assisted sensing enforcement while analytically isolating the sensing-theoretic benefits of adaptive contract logic. The results establish energy-aware smart contracts as an effective mechanism for improving the resilience of CRNs operating in adversarial environments.

Keywords: blockchain; adaptive smart contract; cognitive radio; cooperative spectrum sensing; jamming attack

I. INTRODUCTION

Current communication systems follow a policy of static spectrum allocation. Specific users are assigned spectrum bands and have exclusive license to use their allocated bands. This policy facilitates spectrum management and regulation, and it ensures that licensed users have consistent and reliable service. However, because licensed users often do not fully utilize their assigned bands, and because there is increasing spectrum demand, the policy of static spectrum allocation impedes efficient spectrum use.

Cognitive Radio Networks (CRNs) are an effective solution to policy-induced spectrum scarcity. CRNs enable dynamic spectrum access by allowing unlicensed secondary users (SUs) to opportunistically use allocated spectrum bands when licensed primary users (PUs) are inactive [1]. CRNs also improve spectrum utilization efficiency, and support increasingly congested wireless environments, by

dynamically adapting transmission behavior according to observed spectrum conditions [2]. Standard CRN operation involves four primary functions: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility [8]. Spectrum sensing is particularly critical. Reliable spectrum sensing is a fundamental requirement of CRN operation. When PUs are inactive, there is a spectrum hole. Spectrum sensing is the process by which SUs detect these holes before transmission occurs. If spectrum sensing is not reliable, inaccurate detections may cause harmful interference to PUs or unnecessarily restrict spectrum utilization by SUs.

Cooperative spectrum sensing (CSS), driven by energy detection, is a widely adopted strategy for spectrum sensing. CSS improves sensing reliability through collaborative decision-making among multiple SUs [3]. With CSS, multiple SUs transmit their local sensing decisions to a fusion center, and the fusion center combines these decisions into a global decision about whether a PU is active. When CSS is driven by energy detection, signals received by SUs are passed through a bandpass filter, squared, integrated over a fixed observation period, and compared to a detection threshold to yield local sensing decision. With CSS, fusion of these local decisions is rule-based. OR-rule fusion, which decides that a PU is active if any SU reports PU activity, is particularly attractive. The attraction of OR-rule fusion is its strong cooperative detection sensitivity. But this feature also makes OR-rule fusion highly sensitive to false alarm [4]. When adversarial interference distorts local sensing decisions, a single false signal detection by a SU triggers a positive fusion decision, thereby yielding unnecessary restriction of spectrum utilization by SUs. As wireless environments become increasingly dynamic and congested, maintaining reliable cooperative sensing performance under interference and adversarial behavior remains a major challenge [5].

Here we focus on the challenge of jamming attacks. Jamming attacks are one of the most disruptive threats to reliable cooperative sensing performance by CRNs [6]. Because energy detection is the most widely used method for local spectrum sensing, and because jamming attacks inflate the effective noise variance observed by SUs, jamming attacks degrade local sensing accuracy by increasing the likelihood that energy detectors falsely exceed decision thresholds. Under OR-rule fusion, these local false alarms propagate directly to the cooperative decision layer, resulting

in amplified false alarm rates and degraded spectrum access efficiency.

In this work, we develop a smart contract framework for jamming-resilient cooperative spectrum sensing. Our focus on smart contracts derives from the increasing interest in blockchain-enabled CRN architectures [9]. Blockchain frameworks provide tamper-resistant and decentralized mechanisms for storing cooperative sensing outcomes in distributed CRN environments [11-12]. Within CSS architectures, blockchain can serve as a consensus and verification layer that records sensing decisions, validates participant behavior, and preserves sensing transparency among participating nodes. Smart contracts extend this functionality by enabling automated enforcement of cooperative sensing decisions. In CRNs, smart contracts also may be used to implement fusion logic, evaluate sensing consistency, validate sensing reports, and apply conditional over-ride behavior [13].

Prior blockchain-assisted CRN approaches have primarily addressed trust, sensing integrity, malicious-user detection, and secure spectrum sharing. However, existing smart-contract approaches generally rely on fixed contract-level decision policies that do not explicitly adapt fusion behavior to changing jamming conditions [10, 14-16].

To mitigate jamming-induced false alarms while preserving post-jamming recovery, we propose an energy-aware adaptive smart contract framework. Local sensing uses adaptive noise estimates to update the detection threshold, while the fusion layer introduces a secondary energy-dependent decision boundary that conditionally modifies OR-rule outputs based on jamming, sensing disagreement, and observed energy. Using Gaussian energy detection and binary hypothesis testing, we derive closed-form expressions for detection, false alarm, cooperative fusion, and adaptive override behavior across pre-jamming, active-jamming, and post-jamming conditions. Our primary contributions are summarized as follows:

- Development of an energy-aware adaptive smart contract framework for cooperative spectrum sensing under jamming conditions.
- Derivation of closed-form analytical expressions for detector performance, cooperative OR fusion behavior, and adaptive override probability using Gaussian energy detection theory.
- Analysis of phase-dependent sensing behavior across pre-jamming, active jamming, and post-jamming recovery conditions.
- Demonstration that adaptive contract logic suppresses contract-level false alarm amplification during jamming while reverting toward baseline cooperative sensing behavior post interference.
- Demonstration of prototype smart contract deployment using Solidity and Remix IDE, establishing implementation feasibility for blockchain-assisted sensing enforcement while analytically characterizing sensing-theoretic performance gains.

II. RELATED WORK

Recent blockchain-assisted CRN research has focused primarily on trust, sensing integrity, decentralized spectrum management, malicious-user detection, and secure spectrum access [2]-[14]. Although these approaches improve sensing transparency and decision reliability, many employ fixed contract-level decision rules that do not dynamically adapt to changing interference conditions [15, 16].

Blockchain-assisted CSS security research has largely addressed spectrum sensing data falsification (SSDF), in which malicious users submit forged local sensing decisions to manipulate fusion outcomes [8]. Proposed defenses combine smart contracts with static or adaptive local detection thresholds, user validation, and related security mechanisms [7, 14, 17-20]. A common feature of these approaches is the use of smart contracts to identify or exclude unreliable sensing inputs before fusion.

In contrast, this work uses smart contracts to conditionally override fusion outputs rather than filter sensing inputs. To the best of our knowledge, this fusion-output intervention has not been previously investigated in blockchain-assisted CRNs. We consider barrage jamming, in which wideband noise is transmitted across the spectrum of interest [21], and incorporate CFAR-based jamming detection [22]. The proposed framework extends this approach with an energy-aware adaptive override condition, integrating cooperative spectrum sensing, jamming detection, and blockchain-assisted decision enforcement.

III. SYSTEM MODEL AND MATHEMATICAL FRAMEWORK FOR SIMULATION

A. System Model

The proposed framework models a blockchain-assisted cooperative spectrum sensing architecture operating under dynamically changing interference conditions. Figure 1 illustrates the key components of our model.

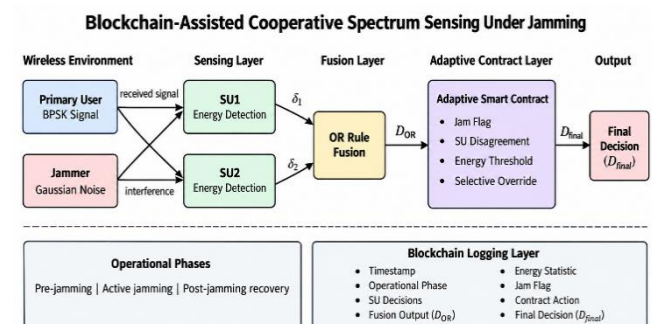


Figure 1. Framework Model for Blockchain-assisted cooperative spectrum sensing under jamming

Wireless Environment. For convenience, we consider a wireless environment with a single primary user (PU) and a Jammer. The PU generates a BPSK-modulated waveform using an equivalent baseband simulation model across

varying signal-to-noise (SNR) conditions. The Jammer injects additive Gaussian noise into the PU signal with adjustable amplitude.

Sensing Layer. Secondary users independently observe the received spectrum environment and perform energy detection-based sensing using equivalent baseband signal representations. For convenience, we consider only two secondary users, SU_1 and SU_2 .

Fusion Layer. The secondary users forward their local sensing decisions to a fusion center, where a baseline sensing decision is generated. The fusion of local sensing decisions is rule-based. The general rule is to decide that a PU is active if m of K SUs report that the PU is active. Common instantiations of this general rule include OR ($m = 1$), AND ($m = K$), and Majority ($m \geq K/2$). Our model adopts OR-rule fusion, which is often preferred for its strong cooperative detection sensitivity.

Adaptive Contract Layer. A blockchain logging layer records the baseline sensing decision, timestamps, and local sensing decisions. It also contains an adaptive smart contract. The contract has three conditions. We consider two of the conditions as static, namely, whether there is disagreement among secondary users, and whether a jamming indicator is flagged. The third condition, involving an energy threshold, is the central innovation for our proposal. The contract uses these conditions to either agree with or override the baseline sensing decision. The determination of the contract is the final decision (D_{final}) about whether the PU is active. Table 1 summarizes the key simulation parameters used throughout the study.

Table 1. Simulation Parameters

Parameter	Value
Modulation Scheme	BPSK
Signal Representation	Equivalent Baseband
Number of Secondary Users (SUs)	2
Noise Model	Additive White Gaussian Noise (AWGN)
Jamming Model	Additive Gaussian Interference
Detection Method	Energy Detection
Fusion Rule	OR Rule
Thresholding Strategy	Adaptive Energy-Aware Thresholding
Samples per Detection Interval (N)	1000
Performance Evaluation Method	Monte Carlo ROC Analysis
ROC Sweep Variable	Detection Threshold (λ)

Operational Phases	Pre-jamming, jamming, post-jamming
--------------------	------------------------------------

B. Signal and Jamming

We consider each secondary user's local sensing decision as an equivalent baseband binary hypothesis test. For signal $x[n]$ received by the i -th SU at sampling instant n , the test is between $\mathcal{H}_0$ (PU inactive) and $\mathcal{H}_1$ (PU active).

$$x[n] = \begin{cases} w[n]: \mathcal{H}_0 \\ s[n] + w[n]: \mathcal{H}_1 \end{cases} \quad (1)$$

$w[n] \sim \mathcal{N}(0, \sigma_w^2)$ is additive white Gaussian noise (AWGN) at SU_i with variance σ_w^2 , and $s[n]$ is the signal transmitted by the PU. With signal variance $\sigma_s^2 = \mathbb{E}[s^2[n]]$, the signal-to-noise ratio (SNR) in linear scale is defined as

$$\gamma = \frac{\sigma_s^2}{\sigma_w^2} \quad (2)$$

Jamming injects additive Gaussian noise, modeled as $j[n] \sim \mathcal{N}(0, \sigma_j^2)$. Hence, during active jamming, the received signal becomes

$$x[n] = s[n] + w[n] + j[n] \quad (3)$$

The effective noise variance under jamming is

$$\sigma_{\text{eff}}^2 = \sigma_w^2 + \sigma_j^2 \quad (4)$$

C. Energy Detection

Because the true noise variance σ_w^2 is unobservable, SUs rely upon empirical estimates to make their local sensing decisions. Because of its simplicity and independence from prior knowledge of the PU signal, energy detection with finite sampling is one of the most widely used methods for this purpose. During operational spectrum sensing, each SU continuously monitors the average received energy E over a sliding window of N samples. The test statistic is

$$E = \frac{1}{N} \sum_{n=1}^N |x[n]|^2 \quad (5)$$

For sufficiently large N , by the Central Limit Theorem (CLT), the statistic is approximately Gaussian:

$$E | \mathcal{H}_0 \sim \mathcal{N}\left(\sigma_w^2, \frac{2\sigma_w^4}{N}\right) \quad (6)$$

$$E | \mathcal{H}_1 \sim \mathcal{N}\left(\sigma_w^2 + \sigma_s^2, \frac{2(\sigma_w^2 + \sigma_s^2)^2}{N}\right) \quad (7)$$

After computing the test statistic E , the SU applies a decision threshold λ :

$$E \underset{\mathcal{H}_0}{\geq} \lambda \quad (8)$$

Performance is evaluable in terms of the probabilities of false alarm (P_{fa}) and detection (P_d). Given the threshold λ , and with $Q(\cdot)$ denoting the Gaussian Q-function, the probabilities are, respectively,

$$P_{fa} = Q\left(\frac{\lambda - \sigma_w^2}{\sqrt{\frac{2\sigma_w^4}{N}}}\right) \quad (9)$$

$$P_d = Q\left(\frac{\lambda - (\sigma_w^2 + \sigma_s^2)}{\sqrt{\frac{2(\sigma_w^2 + \sigma_s^2)^2}{N}}}\right) \quad (10)$$

Solving (9) for λ yields

$$\lambda = \sigma_w^2 + \sqrt{\frac{2\sigma_w^4}{N}} Q^{-1}(P_{fa}) \quad (11)$$

The decision threshold in (11) is fixed according to the noise-only distribution and a target probability of false alarm. This entails that the probability of false alarm increases when there is jamming interference. Under jamming, substitute $\sigma_w^2 \rightarrow \sigma_w^2 + \sigma_j^2$ in (10). Because the total noise variance increases under jamming, the numerator in (9) decreases and the denominator increases. Because $Q(\cdot)$ is monotonically decreasing, and because the decision threshold in (11) is fixed relative to the noise-only distribution (without jamming), the probability of false alarm increases under jamming.

D. Noise Variance Estimation for Adaptive Thresholding

When the noise variance is not known *a priori*, practical implementation requires adaptive thresholding with noise variance estimation. Adaptive thresholding uses adaptive noise estimates to dynamically update the detection threshold. Rather than using the true noise variance σ_w^2 , adaptive thresholding uses a sample-based estimate $\hat{\sigma}_w^2$. During a known idle calibration window of N_0 samples, each SU estimates the baseline noise variance as

$$\hat{\sigma}_w^2 = \frac{1}{N_0} \sum_{n=1}^{N_0} |x_0[n]|^2 \quad (12)$$

The adaptive detection threshold λ_{adapt} is then computed as

$$\lambda_{\text{adapt}} = \hat{\sigma}_w^2 + \sqrt{\frac{2\hat{\sigma}_w^4}{N}} Q^{-1}(P_{fa}) \quad (13)$$

Because adaptive thresholding allows the detector operating point to dynamically track changes in the effective noise floor, updating the decision threshold using (13) can partially restore the operating point for the targeted probability of false alarm. Although adaptive thresholding is adopted throughout the subsequent derivations to illustrate the proposed framework, the contract-level decision logic is independent of the local sensing algorithm and can be integrated with alternative sensing strategies.

E. Cooperative Spectrum Sensing (OR Fusion)

Supposing that each SU uses adaptive thresholding with noise variance estimation, let $D_i \in \{0,1\}$ denote the local sensing decision by the i -th SU.

$$D_i = \begin{cases} 0, & \text{if } E < \lambda_{\text{adapt}}: \mathcal{H}_0 \\ 1, & \text{if } E \geq \lambda_{\text{adapt}}: \mathcal{H}_1 \end{cases} \quad (14)$$

Let the baseline sensing decision $F \in \{0,1\}$ be the output from fusing these decisions with the OR-rule.

$$F = \begin{cases} 0, & \text{if } D_i = 0 \text{ for all } SU_i: \mathcal{H}_0 \\ 1, & \text{if } D_i = 1 \text{ for any } SU_i: \mathcal{H}_1 \end{cases} \quad (15)$$

We consider the case in which there are only two SUs. In this case, $F = D_1 \vee D_2$. Substituting $\lambda \rightarrow \lambda_{\text{adapt}}$ in (9) and (10), the global false alarm and detection probabilities are, respectively,

$$P_{fa}^{(OR)} = 1 - (1 - P_{fa1})(1 - P_{fa2}) \quad (16)$$

$$P_d^{(OR)} = 1 - (1 - P_{d1})(1 - P_{d2}) \quad (17)$$

F. Jamming Detection Gate

For the sake of improving performance during jamming, our model uses a global average energy metric E_{avg} to detect abnormal interference. For only two SUs, the metric is

$$E_{\text{avg}} = \frac{E_1 + E_2}{2} \quad (18)$$

A global jamming indicator, Jam , evaluates this aggregate energy against an adaptive jamming threshold τ_j .

$$\tau_j = \kappa \hat{\sigma}_w^2, \quad \kappa \in [2,4] \quad (19)$$

The comparison between E_{avg} and τ_j defines the jamming indicator.

$$\text{Jam} = \mathbb{I}\{E_{\text{avg}} > \tau_j\} \quad (20)$$

When $E_{\text{avg}} > \tau_j$, Jam = 1; otherwise, Jam = 0.

G. Blockchain Smart Contract Policies

Smart contracts are not independent detectors. They are, instead, conditional veto mechanisms for selectively overriding baseline sensing decisions. If a smart contract overrides fusion decisions only when statistical indicators are consistent with jamming-induced distortion, it preserves baseline sensing behavior whenever jamming interference is absent. For the sake of further improving performance under jamming, we consider two such contracts.

The first contract, $SC_{\text{non-adaptive}}$ ingests local sensing decisions from SUs along with baseline fusion decisions.

$$SC_{\text{non-adaptive}} = \begin{cases} 0, & \text{if Jam} = 1 \text{ and } D_1 \neq D_2: \mathcal{H}_0 \\ F, & \text{otherwise: } \mathcal{H}_1 \end{cases} \quad (21)$$

$SC_{\text{non-adaptive}}$ overrides fusion decisions when the jamming indicator flags active interference and local sensing decisions disagree with each other. Otherwise, it defers to the baseline sensing decision.

The second contract, SC_{adapt} , is our proposed energy-aware adaptive contract. Like $SC_{\text{non-adaptive}}$ this contract ingests local sensing decisions from SUs and baseline fusion decisions. But the adaptive contract differs from $SC_{\text{non-adaptive}}$ by introducing an energy-dependent override threshold e_ℓ .

$$SC_{\text{adapt}} = \begin{cases} 0, & \text{if Jam} = 1, D_1 \neq D_2, E_{\text{avg}} > e_\ell: \mathcal{H}_0 \\ F, & \text{otherwise: } \mathcal{H}_1 \end{cases} \quad (22)$$

SC_{adapt} also defers to the baseline sensing decision when either the jamming indicator does not flag active interference or the local sensing decisions agree with each other. But the adaptive contract is more selective. When the global average energy does not exceed the override threshold, it defers to the baseline sensing decision even if the jamming indicator flags active interference and local sensing decisions disagree with each other.

H. ROC Construction

For fixed (N, σ_w^2, γ) and varying λ_{adapt} or e_ℓ , empirical performance metrics are computed as

$$\hat{P}_{fa} = \frac{\text{count}(SC = 1 | \mathcal{H}_0)}{\text{count}(\mathcal{H}_0)} \quad (23)$$

$$\hat{P}_d = \frac{\text{count}(SC = 1 | \mathcal{H}_1)}{\text{count}(\mathcal{H}_1)} \quad (24)$$

For all receiver operating characteristic (ROC) calculations, the true primary-user occupancy state is used as the ground-truth label when computing detection and false alarm probabilities. Fusion outputs and smart-contract decisions are evaluated relative to the underlying hypothesis rather than serving as ground-truth references. The resulting $(\hat{P}_{fa}, \hat{P}_d)$ pairs form the ROC curve for each policy.

I. Prototype Smart Contract Implementation and Modeling Assumptions

To investigate implementation feasibility, an end-to-end blockchain-assisted sensing prototype was developed using ADALM-Pluto SDR hardware, GNU Radio, Python middleware, Hardhat, Remix IDE, and a Solidity smart contract. The prototype acquires IQ samples, converts them to power and RSSI measurements, performs signal and jamming detection, and submits structured sensing records through a Web3 interface. The smart contract validates sensing inputs and records signal frequency, RSSI, detection status, timestamps, and cryptographic metadata as immutable blockchain events.

Successful contract deployment, transaction execution, event logging, timestamp generation, and record retrieval were verified using Remix IDE and a local Hardhat blockchain environment. The prototype therefore establishes the implementation feasibility of integrating SDR-based spectrum sensing with blockchain-assisted sensing enforcement. The adaptive analytical framework extends this implementation by incorporating jamming indicators, cooperative sensing disagreement, and energy-dependent thresholds to selectively override baseline fusion decisions.

The prototype serves as a proof of implementation feasibility rather than a large-scale blockchain performance evaluation. Accordingly, consensus protocols, transaction propagation, network latency, gas consumption, and block generation are excluded from the analytical model to isolate the sensing-theoretic effects of adaptive contract logic. Figure 2 illustrates the prototype architecture, and Table 2 summarizes its implementation characteristics.

J. Simulation Reproducibility

All simulation results were generated using 1000 Monte Carlo trials per operating point. A fixed random seed was used to ensure reproducibility. Detection and false alarm probabilities were computed using empirical counts over all trials. The selected trial count was sufficient to produce stable ROC estimates and consistent performance trends across repeated simulation runs.

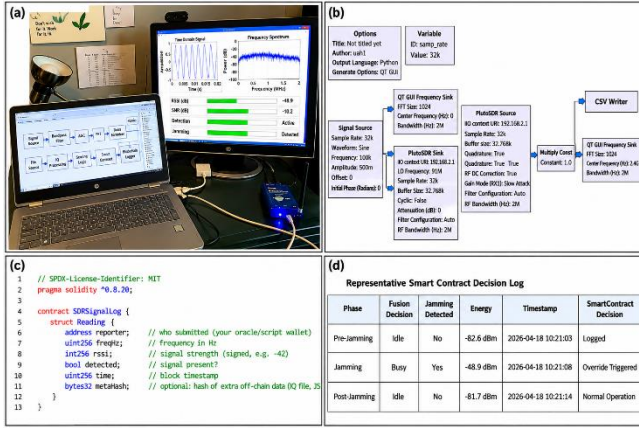


Figure 2. Prototype Implementation of the Blockchain-Assisted Spectrum Sensing Framework. (a) SDR-based prototype platform used for representative spectrum signal acquisition. (b) GNU Radio signal processing pipeline for sample acquisition and preprocessing prior to blockchain interaction. (c) Solidity smart contract implementing deterministic sensing-event validation and immutable blockchain logging. (d) Representative blockchain decision log illustrating recorded sensing outcomes during the pre-jamming, jamming, and post-jamming operational phases.

Table 2. Prototype Implementation Characteristics

Attribute	Prototype Implementation
Smart contract language	Solidity 0.8x
Development environment	Remix IDE
Contract operations	Boolean logic, threshold comparison
Floating point arithmetic	Not used
Event logging	Supported
Timestamp generation	Supported
Record retrieval	Supported
Public Ethereum deployment	Not evaluated
Gas consumption analysis	Outside the scope of this study
Permissioned blockchain suitability	Recommended for deployment
Blockchain Framework	Hardhat (local Ethereum network)

IV. RESULTS

This section evaluates detector-level performance, cooperative fusion behavior, and smart contract decision policies under three operational phases: pre-jamming, active jamming, and post-jamming recovery. All probabilities are computed with respect to the true hypotheses H_0 and H_1 defined in Section 3.2.1.

A. Energy Distribution under Jamming

Figure 3 illustrates the empirical distribution of the energy statistic E defined in (5). Under H_0 (no PU, no jammer), the distribution of E is tightly concentrated around σ_w^2 , consistent with the Gaussian approximation in (6):

$$E | \mathcal{H}_0 \sim \mathcal{N}\left(\sigma_w^2, \frac{2\sigma_w^4}{N}\right)$$

When jamming is introduced, the effective variance increases to $\sigma_w^2 + \sigma_f^2$, resulting in a rightward shift of the energy distribution:

$$E | \text{Jam} \sim \mathcal{N}\left(\sigma_w^2 + \sigma_f^2, \frac{2(\sigma_w^2 + \sigma_f^2)^2}{N}\right)$$

This variance inflation increases the overlap between H_0 and H_1 , thereby degrading the detector separability when the decision threshold λ remains fixed.

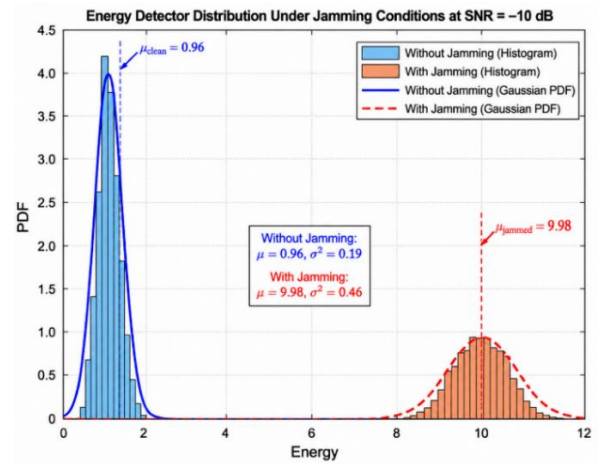


Figure 3. Clean vs. Jammed Energy Distribution

B. Detector-Level ROC Performance

The ROC curve for energy detection is generated by sweeping λ in (8). False alarm and detection probabilities are computed using (9) and (10).

During the pre-jamming phase, the ROC follows the theoretical curve derived from (10):

$$P_d = Q\left(\frac{\lambda - (\sigma_w^2 + \sigma_s^2)}{\sqrt{\frac{2(\sigma_w^2 + \sigma_s^2)^2}{N}}}\right)$$

Under active jamming, substituting $\sigma_w^2 \rightarrow \sigma_w^2 + \sigma_f^2$ shifts the ROC downward, reflecting reduced P_d for a fixed P_{fa} . This degradation is consistent with the increase in effective noise variance described in Section 3.2.1.

Adaptive thresholding using (13) partially restores the detector operating point by compensating for changes in the effective noise floor. Figure 4 confirms ROC degradation

during active jamming and recovery after interference subsides.

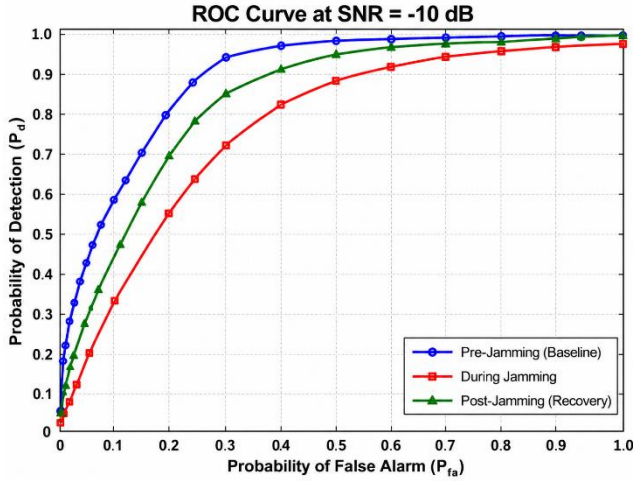


Figure 4. Detector-level ROC performance under active jamming conditions.

C. Override Rate and Decision Dynamics

To quantify smart contract behavior, we define the override probability

$$P_{\text{override}} = \Pr\{SC \neq F\} \quad (25)$$

We consider override probabilities for both smart contracts from Section 3.2.6. For the non-adaptive contract,

$$P_{\text{override}}^{\text{non-adaptive}} = \Pr(\text{Jam} = 1, D_1 \neq D_2) \quad (26)$$

Because the probability of disagreement is

$$\Pr(D_1 \neq D_2) = P_{d1}(1 - P_{d2}) + P_{d2}(1 - P_{d1}) \quad (27)$$

under H_1 , and similarly under H_0 , the override rate remains bounded by the probability of disagreement. For the adaptive contract,

$$P_{\text{override}}^{\text{adapt}} = \Pr(\text{Jam} = 1, D_1 \neq D_2, E_{\text{avg}} > e_\ell) \quad (28)$$

Because E_{avg} increases under jamming, the probability term

$$\Pr(E_{\text{avg}} > e_\ell | \text{Jam} = 1)$$

approaches unity when e_ℓ is below the inflated mean energy level. The adaptive contract introduces a more selective override mechanism by requiring both disagreement and an energy-dependent condition. Consequently, override events are restricted to statistically significant interference conditions rather than being applied uniformly during jamming:

$$P_{fa}^{\text{adapt}} < P_{fa}^{\text{non-adaptive}} \quad (\text{under jamming})$$

In the post-jamming phase, because

$$E_{\text{avg}} \rightarrow \sigma_w^2$$

the probability

$$\Pr(E_{\text{avg}} > e_\ell) \rightarrow 0$$

for appropriately chosen e_ℓ , causing the adaptive override probability to decay rapidly. This ensures recovery to baseline OR fusion performance with the adaptive contract.

D. Cooperative OR Fusion Performance

The cooperative sensing ROC using the OR rule follows equations (16) and (17):

$$P_{fa}^{(OR)} = 1 - (1 - P_{fa1})(1 - P_{fa2})$$

$$P_d^{(OR)} = 1 - (1 - P_{d1})(1 - P_{d2})$$

Under jamming, increased individual SU false-alarm probabilities are amplified by the multiplicative structure of OR fusion in (16), motivating contract-level intervention.

E. Phase-Based ROC Comparison

Figure 5 compares contract-level ROC behavior across the three operational phases. During active jamming, both contract policies suppress false alarms relative to OR fusion, with the adaptive contract providing more selective intervention. Both return toward baseline behavior after jammer removal.

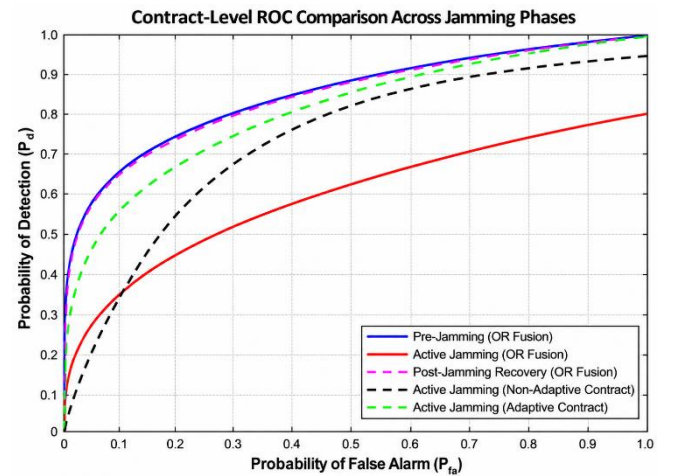


Figure 5. Contract-level ROC comparison across pre-jamming, active-jamming, and post-jamming recovery conditions.

Table 3 provides a quantitative summary of the phase-dependent behavior observed in the ROC analysis. Confidence intervals were computed using the normal approximation to the binomial distribution at the 95% confidence level.

Table 3. Quantitative comparison of OR fusion, non-adaptive smart contract, and adaptive smart contract policies across pre-jamming, active-jamming, and post-jamming recovery phases using 1000 Monte Carlo trials per operating point.

Operational Phase	Policy	P_d (95% CI)	P_{fa} (95% CI)	Override rate	False-alarm reduction vs. OR (%)
Pre-Jamming	OR fusion				
	Non-adaptive contract	0.796 [0.771, 0.821]	0.192 [0.168, 0.216]	0.000	0.0
	Adaptive contract				
Active Jamming	OR fusion	0.996 [0.992, 1.000]	0.910 [0.892, 0.928]	0.000	0.0
	Non-adaptive contract	0.905 [0.887, 0.923]	0.460 [0.429, 0.491]	0.271	49.5
	Adaptive contract	0.910 [0.892, 0.928]	0.650 [0.620, 0.680]	0.173	28.6
Post-Jamming	OR fusion				
	Non-adaptive contract	0.805 [0.780, 0.830]	0.178 [0.154, 0.202]	0.000	0.0
	Adaptive contract				

Table 3 confirms greater false-alarm suppression by the non-adaptive contract, while the adaptive contract achieves a lower override rate and slightly higher detection probability, demonstrating more selective intervention

F. Theoretical Influence of Energy Threshold e_ℓ

The threshold parameter e_ℓ directly controls override sensitivity within the adaptive contract framework. Smaller values of e_ℓ increase override activity and produce more aggressive suppression of statistically inconsistent fusion decisions during jamming. By contrast, larger values of e_ℓ result in more conservative override behavior and preserve baseline cooperative sensing decisions. Consequently, e_ℓ establishes a tunable trade-off between resilience against jamming-induced false alarms and preservation of cooperative detection performance. By including a condition with this parameter in its policy, the adaptive contract thereby effectively introduces a secondary decision boundary in energy space, enabling dynamic reshaping of the contract-level ROC behavior.

V. DISCUSSION

A. Implications

The results demonstrate that the proposed energy-aware adaptive smart contract extends blockchain-assisted cooperative spectrum sensing beyond static decision enforcement by introducing a secondary, energy-dependent decision boundary at the fusion layer. This mechanism selectively suppresses statistically inconsistent OR-fusion

decisions during interference while preserving baseline behavior after jamming subsides. Unlike prior blockchain-enabled sensing approaches focused primarily on trust, security, and data integrity, the proposed framework incorporates interference-induced changes in sensing statistics directly into contract-level decision logic.

B. Limitations

Several limitations should be acknowledged. The analysis assumes Gaussian noise and jamming models, which may not fully represent more complex or structured interference scenarios. The current framework also considers a limited number of cooperative users. Extending the analysis to larger, heterogeneous networks with varying sensing reliability remains an important direction for continued research.

The blockchain layer is treated as an idealized execution environment, excluding transaction latency, propagation delay, gas consumption, and scalability constraints. These factors may affect continuous sensing deployments, particularly on public blockchain networks where frequent sensing transactions can incur recurring costs. Future work will therefore evaluate gas consumption, transaction throughput, latency, and permissioned blockchain architectures under continuous sensing workloads.

VI. CONCLUSION AND FUTURE WORK

This work developed an energy-aware adaptive smart contract framework for jamming-resilient cooperative spectrum sensing. Analytical and simulation results show that the proposed contract selectively mitigates jamming-induced false-alarm amplification while returning toward baseline OR-fusion behavior after interference subsides. Future work will investigate larger heterogeneous sensing networks, structured jamming, blockchain latency and transaction costs, and expanded SDR-based experimental validation.

REFERENCES

- [1] J. Mitola, "Cognitive radio: Making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–19, 1999.
- [2] Z. Chen, L. Wang, and Y. Zhang, "Blockchain Structure Electromagnetic Spectrum Database in Distributed Cognitive Radio Monitoring System," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 4, pp. 1647–1664, Dec. 2022.
- [3] Y. Arjouni, N. Kaabouch, "A Comprehensive Survey on Spectrum Sensing in Cognitive Radio Networks: Recent Advances, 832 New Challenges, and Future Research Directions," *Sensors* 2019, 19, 126. <https://doi.org/10.3390/s19010126>
- [4] G. Ding, Q. Wu, Y.-D. Yao, J. Wang, and Y. Chen, "Kernel-based learning for statistical signal processing in cognitive radio 834 networks: Theoretical foundations, example applications, and future directions," *IEEE Signal Processing Magazine*, vol. 30, no. 4, 835 pp. 126–136, Jul. 2013.
- [5] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, First Quarter 2009.
- [6] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, Jan. 2008.
- [7] E. Ezhilarasi I and J. C. Clement, "Robust cooperative spectrum sensing in cognitive radio blockchain network using SHA-3 algorithm," *Blockchain: Research and Applications*, vol. 5, no. 4, 2024, Art. no. 100224, ISSN 2096-7209.

- [8] N.Y. Al-Matari, A. T. Zahary, and A. Al-Shargabi, "A survey on advancements in blockchain-enabled spectrum access security for 6G cognitive radio IoT networks," *Scientific Reports*, vol. 14, no. 30990, 2024.
- [9] H. Luo et al., "Wireless Blockchain Meets 6G: The Future Trustworthy and Ubiquitous Connectivity," *IEEE Communications Surveys & Tutorials*, 2025, doi: 10.1109/COMST.2025.3593918
- [10] S. Badruddoja, R. Dantu, Y. He, K. Upadhayay, and M. Thompson, "Making Smart Contracts Smarter," in *Proc. IEEE Int. Conf. Blockchain and Cryptocurrency (ICBC)*, Sydney, Australia, 2021, pp. 1–3, doi: 10.1109/ICBC51069.2021.9461148
- [11] T. R. Gadekallu et al., "Blockchain for Edge of Things: Applications, Opportunities, and Challenges," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 964–988, Jan. 15, 2022, doi: 10.1109/JIOT.2021.3119639
- [12] D. C. Nguyen et al., "Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12806–12825, Aug. 15, 2021, doi: 10.1109/JIOT.2021.3072611.
- [13] D. Baker, T. Syed, S. Chakravarty, and L. Joiner, "Enhancing Energy Detection-Based Spectrum Sensing Through Blockchain Technology," in *Proc. IEEE SoutheastCon*, Atlanta, GA, USA, 2024, pp. 904–906, doi: 10.1109/SoutheastCon52093.2024.10500118
- [14] V. Roopa and H. S. Pradhan, "Blockchain Based Spectrum Sensing for Secured Cognitive Radio Wireless Networks," in *Proc. IEEE 11th Int. Conf. Communication Systems and Network Technologies (CSNT)*, Indore, India, 2022, pp. 553–559, doi: 10.1109/CSNT54456.2022.9787585
- [15] Z. Xiandong, P. Zhiwen, and L. Nan, "Optimization of Two-Stage Consensus and Storage of Blockchain-Based Spectrum Sensing in Cognitive Radio Networks," *IEEE Sensors Journal*, vol. 25, no. 13, pp. 24431–24440, Jul. 1, 2025, doi: 10.1109/JSEN.2025.3567951
- [16] Z. Wang, X. Kang, Z. Zhou, Q. Tang, and Y.-C. Liang, "A Trust-Centric Blockchain-Enabled Fair Cooperative Spectrum Sensing System for IoT Networks," *IEEE Internet of Things Journal*, 2025, doi: 10.1109/JIOT.2025.3595939
- [17] C. Ji and Q. Zhu, "Smart contract-based secure cooperative spectrum sensing algorithm," *International Journal of Distributed Sensor Networks*, vol. 17, no. 12, pp. 1–12, 2021, doi: 10.1177/15501477211058673
- [18] A. Khanna et al., "Blockchain-Based Security Enhancement and Spectrum Sensing in Cognitive Radio Network," *Wireless Pers. Commun.*, vol. 127, pp. 1899–1921, 2022, doi: 10.1007/s11277-021-08729-0
- [19] D. Balakumar and N. Sendrayan, "Enhance the Probability of Detection of Cooperative Spectrum Sensing in Cognitive Radio Networks Using Blockchain Technology," *Journal of Electrical and Computer Engineering*, vol. 2023, article 8920243, 2023, doi: 10.1155/2023/8920243
- [20] N.K. Marriwala et al., "An Analytical Model for Dynamic Spectrum Sensing in Cognitive Radio Networks Using Blockchain Management," *Eng. Proc.*, vol. 59, no. 1, article 163, 2023, doi: 10.3390/engproc2023059163
- [21] M. Lichtman et al., "A communications jamming taxonomy," *IEEE Security & Privacy*, vol. 14, no. 1, pp. 47–54, Jan.-Feb. 2016, doi: 10.1109/MSP.2016.13
- [22] D. Kun and N.A. Morgan, "A new low-cost CFAR detector for spectrum sensing with cognitive radio systems," 2009 IEEE Aerospace conference, Big Sky, MT, USA, 2009, pp. 1–8, doi: 10.1109/AERO.2009