I.J. Information Engineering and Electronic Business, **2026, 5, 163-181**
Published Online on October 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2026.05.10

ProCLAMUS: A Matrix Completion Based Proactive Spectrum Allocation Protocol for Security of IoT based CRNs

Charanjeet Singh Chahil*
Department of Computer Science, Chitkara University, Rajpura (Punjab), India
E-mail: Charanjeet1002cse.phd21@chitkara.edu.in
ORCID iD: https://orcid.org/0009-0001-4177-9175
*Corresponding author

Milan Patnaik
Faculty of Informatics, Masaryk University, Brno, Czech Republic
E-mail: 459197@muni.cz
ORCID iD: https://orcid.org/0000-0002-2877-1195

Deepali Gupta
Department of Computer Science, Chitkara University, Rajpura (Punjab), India
E-mail: Deepali.Gupta@chitkara.edu.in
ORCID iD: https://orcid.org/0000-0002-3207-5248

Vashek Matyas
Faculty of Informatics, Masaryk University, Brno, Czech Republic
E-mail: matyas@fi.muni.cz
ORCID iD: https://orcid.org/0000-0001-7957-7694

Received: January 24, 2026; Revised: April 8, 2026; Accepted: June 29, 2026; Published: October 8, 2026

Abstract: Proactive prediction based channel allocation for Cognitive Radio Networks (CRNs) remains a challenging area of research. In Internet of Things (IoT) scenarios, prediction algorithms must balance resource efficiency with high accuracy. The proposed algorithm, ProCLAMUS, a Matrix Completion (MC) Based Proactive Spectrum Allocation Protocol for CRNs, couples nuclear norm matrix completion with short horizon sliding window prediction. By reconstructing sparse spectrum sensing data and down weighting inconsistent reports, ProCLAMUS enables infrequent sensing and decentralized decisions without a fusion center, reducing energy and attack surface. In a network of 40 Secondary Users (SUs), 400 Primary Users (PUs) and 400 channels with malicious data ranging from 5 to 50%, ProCLAMUS sustains the highest channel utilization (avg 89.96%) with the lowest backoff rate (3.81 s^{-1}) and sensing delay (0.58 channels/success). ProCLAMUS achieves the lowest radio energy (16.82×10^{-3} J/s), a 44–53% reduction when compared with recently proposed techniques, while using 33–43% less memory. These gains arise from sparse sensing, fusion (majority voting) and conservative allocation. The results demonstrate superior energy efficiency and robust prediction under malicious data conditions during the Spectrum Sensing Data Falsification Attack.

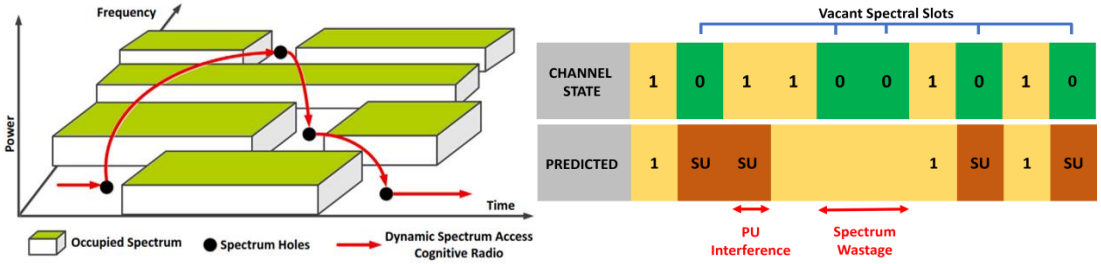
Index Terms: Cognitive Radio Networks, Matrix Completion, Internet of Things, Spectrum Sharing, Spectrum Decision, Spectrum Allocation.

1. Introduction

Cognitive Radios (CR) were introduced by Mitola as an intelligent solution to overcome the spectrum scarcity by introducing a wireless communication system that is capable of altering the transmission and reception parameters based on the environment [1]. The pivotal work by Mitola and G Q Maguire [1] defined Cognitive Radios as “radio that employs model-based reasoning to achieve a specified level of competence in radio related domains”, thus moving beyond the concept of static parameters of wireless devices. The CR system follows a cognition cycle that essentially

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

senses, learns and adapts. Thus, CR offers a novel approach in spectrum utilisation and management. Modern CRNs emphasize dynamic spectrum access, combining cooperative sensing and prediction to maximize utilization while ensuring timely vacation under adversarial scenarios.



(a) Spectrum Holes (b) Spectrum Allocation

Fig. 1. Illustrations Depicting Opportunistic Utilization of Spectrum.

1.1. Concept of CRN for Efficient Spectrum Utilization

The CR concept involves Primary Users (PUs) and Secondary Users (SUs). PUs are licensed users of a frequency band, while SUs are unlicensed users that opportunistically access the band when it is not used by PUs. Thus, CRs detect PU activity and exploit spectrum holes (white spaces) without causing harmful interference [2]. Mitola's cognition cycle includes three key functions: spectrum sensing, spectrum management and spectrum mobility. Spectrum sensing identifies available spectrum holes, spectrum management selects suitable holes for use and spectrum mobility enables seamless switching when PU activity resumes [1, 2]. Figure 1 illustrates opportunistic spectrum utilization, where spectrum holes are identified and allocated based on prediction. The rapid growth of wireless devices has intensified spectrum scarcity; however, studies show that much of the licensed spectrum remains underutilized. McHenry and McCloskey (2005) reported that spectrum occupancy in Chicago was nearly 17.4% across the 30 MHz–3 GHz range, revealing significant temporal and spatial gaps [3, 4]. CR allows SUs to dynamically access such underutilized spectrum, improving spectrum efficiency, network throughput and system performance while preserving PU Quality of Service (QoS) [5]. Cooperative Spectrum Sensing (CSS) enables multiple SUs to share Spectrum Sensing Data (SSD) packets with a fusion center or neighbouring nodes to jointly determine PU occupancy. CSS in CRNs mainly follows two approaches [6]:

- Centralized CSS: SSD packets from SUs are processed by a Fusion Centre to determine spectrum occupancy and assign channels [7].
- Decentralized CSS: Each SU makes autonomous decisions or collaborates with neighbouring nodes to identify spectrum holes [7].

1.2. Security of Cognition Cycle

SSD packets are the basis of the cognitive cycle in CRNs. SUs sense the spectrum in their vicinity and report the status through SSDs. The accuracy of this data is critical to ensure effective spectrum sensing and optimal channel allocation. Collaborative sensing, in which SSDs from multiple SUs are combined, enhances consistency by leveraging spatial diversity and reducing the impact of discrete sensing errors [8]. However, this collaborative process also raises concerns regarding the integrity and trustworthiness of the SSDs; the large data exchange provides an enhanced attack surface to be exploited. Among the most critical security threats in CRN are Spectrum Sensing Data Falsification (SSDF) and Primary User Emulation Attack (PUEA) [9].

- SSDF: In SSDF attacks, malicious SUs intentionally inject falsified SSDs, leading to flawed spectrum allocation decisions [10].
- PUEA: Here the attacker imitates the signal characteristics of PUs, forcing SUs into abandoning the channels needlessly. This creates fake spectrum scarcity [11].

Robust predictive models are important for achieving efficient channel allocation while mitigating the SSDF. To remain effective, they must sustain channel efficiency, reduce collisions with PUs and operate reliably even in conditions of rogue SUs feeding malicious SSDs [12, 13].

The remainder of the paper is organized as follows. Section 2 surveys the evolution of channel prediction algorithms and gives a taxonomy of security issues in CRNs, secure protocols and Technical Contributions. Section 3 introduces Matrix Completion for prediction, explains the reduction of attack surface through Matrix Completion and gives out our approach towards handling the spectrum allocation. Section 4 presents the ProCLAMUS algorithm (system model, design choices and operational steps); Sections 5–7 present the Experimental and Emulation setup

including the transmission model and presents results and comprehensive analysis across varying attack intensities. Section 8 presents the theoretical validation of robustness, while Section 9 concludes by summarizing the key findings and outlining future directions.

2. Related Work

2.1. Evolution of Algorithms for Channel Prediction

Over the past two decades, channel prediction and spectrum-decision mechanisms in CRNs have evolved from low-complexity statistical models to data-driven and security-aware designs. This evolution has been driven by bursty primary-user activity, heterogeneous sensing quality, sparse reporting and strict energy constraints in IoT environments. Consequently, modern designs must balance prediction accuracy with robustness, signaling overhead and implementation cost.

- *Statistical and Signal Processing Methods.* Early approaches relied on Hidden Markov Models (HMMs), Bayesian inference and ARIMA style temporal modelling. These methods remain attractive because of their low complexity and interpretability, but their performance degrades under highly non-stationary, heterogeneous or low-SNR CRN conditions [18].
- *Machine Learning and Deep Learning.* As richer sensing histories became available, machine learning (ML) and deep learning (DL) techniques such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and Convolutional Long Short-Term Memory (ConvLSTM) were adopted for spectrum sensing, occupancy prediction and channel selection. Recent reviews report clear gains from these models in capturing nonlinear temporal and spatio-temporal patterns, although they also introduce training-data, model-update and computation overheads [18,19].
- *Trust, Blockchain and Attack Aware prediction.* More recent work integrates trust scoring, secure provenance and blockchain support to mitigate SSDF and related attacks before prediction and allocation. Protocols such as ProBLeSS and ProMETHEUS illustrate how secure reporting can improve decision reliability, while recent surveys also highlight the associated latency, storage and consensus overheads in IoT-class CRNs [15,16,20, 21].
- *Matrix Completion and Sparse Recovery.* Matrix completion exploits the low-rank structure of partially observed sensing matrices to recover missing or corrupted entries, making it attractive when sensing is infrequent or reporting is incomplete. This direction is especially relevant to CRNs because spatio-temporal correlations can be leveraged to reconstruct channel occupancy from sparse measurements [22, 23, 24].

2.2. Taxonomy of Security in Cognitive Radio Networks

2.2.1. Attack Surfaces and Affected CRN Functions

Security threats in CRNs span multiple layers and functions. At the physical layer, jamming, spoofing, and PUEA can trigger false vacation events [25, 26, 27, 28]. At the sensing layer, SSDF degrades CSS through random, biased, opportunistic or collusive reports, while learning-based sensing remains vulnerable to poisoning and adversarial manipulation [20, 29, 31]. At the MAC and control layers, adversaries disrupt coordination through control channel jamming, forged beacons, reservation injection, selfish channel capture and backoff manipulation [26, 27]. Network and application layers are exposed to routing attacks, Sybil behavior, privacy leakage, unauthorized access and credit abuse [21, 27].

2.2.2. Attacker Characteristics, Behavior, and Objectives

CRN attackers may be external or internal, passive or active and isolated or collusive [26, 27, 29]. Their capabilities range from low-power opportunists to adaptive adversaries that jam, emulate or strategically falsify sensing reports [25, 28]. Threat severity also depends on attacker knowledge, from black-box access to white-box knowledge of thresholds, fusion rules or learning pipelines [20, 29]. Attacks may be continuous, intermittent, reactive or proactive, producing immediate disruption or gradual corruption of trust scores and learned models [20, 29]. Their objectives typically target availability, integrity, confidentiality, privacy or economic fairness in spectrum access [21, 27].

2.2.3. Defense, Deployment, and Governance

CRN defenses combine authentication, access control, trust management, robust fusion, anomaly detection, and adversarially resilient learning [20, 26, 27, 30]. Trust and reputation mechanisms weight reports by historical reliability, while anomaly-aware fusion and malicious-user detection improve robustness against SSDF and collusion [30, 31]. Provenance and audit mechanisms, including blockchain or smart-contract frameworks, strengthen accountability and automate penalties, often with protected control channels, spectrum hopping and fallback signaling [15, 21]. Centralized architectures simplify enforcement but introduce single points of failure, whereas distributed designs improve resilience

at higher coordination cost [26, 27]. Governance further depends on identity management, revocation and privacy policy in open or permissioned environments [21].

2.2.4. Evaluation Criteria

Security mechanisms are assessed by resilience to false vacancy, false occupancy, SSDF, collusion and control-channel disruption [29, 30]. Additional metrics include channel utilization, delay, backoff behaviour, energy consumption, control overhead and sensitivity to SNR variation, mobility, model drift, or poisoning [20, 26, 27]. Practical evaluation should also consider scalability, memory and computation cost, deadline compliance, dependence on external infrastructure and the trade-off between privacy leakage and forensic accountability [21].

2.3. Secure CRN Protocols

2.3.1. Authentication MAC Schemes

Authentication-based MAC protocols protect CRN control and data exchanges from spoofing, replay and frame injection that may compromise sensing beacons or reservation frames. Lightweight handshakes, rolling keys and per-frame message authentication codes preserve integrity and authenticity while limiting latency, header overhead and energy cost, which is essential for IoT-class CRNs [26, 27]. Role-based access control further restricts privilege misuse. However, these schemes do not prevent insider manipulation of sensing reports and are therefore most effective when combined with sensing-layer defenses [27].

2.3.2. Trust-Based SSDF Mitigation

SSDF attacks occur when malicious SUs falsify local observations or decisions, biasing fusion outcomes toward false vacancy or false occupancy. Conventional fusion rules assume equal reliability, making them vulnerable to collusion and adaptive adversaries [29, 30]. Trust-based mitigation assigns each SU a reliability score using agreement with global decisions, neighbourhood consistency or residual behavior, then discounts suspicious reports during fusion. Bayesian, reinforcement-learning and weighted-voting updates can penalize persistent deviators while allowing gradual rehabilitation of honest nodes [20, 30]. Recent malicious-user detection models further strengthen this stage by learning discriminative patterns in falsified sensing behaviour [31]. Effective designs must balance fast adaptation with stability and low overhead, especially in dense IoT-style CRNs [29, 30].

2.3.3. Trust/Robust Cooperative Spectrum Sensing

Trust-aware CSS improves resilience by weighting user reports according to historical reliability and consistency with expected sensing behaviour [30]. Robust statistics, anomaly detection and hybrid trust–anomaly frameworks complement this process by isolating unstable or colluding reporters [26, 29]. Clustered and consensus-based sensing reduce reliance on a single fusion point, although they must balance agility against false penalization of honest users under fading or mobility. Establishing trust for new or intermittent SUs without enabling Sybil-style resets remains a key challenge [30].

2.3.4. Blockchain-Backed Sharing and Provenance

Blockchain-based CRN frameworks improve provenance and auditability by recording sensing reports, access decisions and credit transactions in tamper-evident ledgers [15, 20, 21]. Smart contracts automate enforcement, penalties, and settlement, reducing dependence on centralized authorities. To satisfy CRN timing and energy constraints, practical designs typically store only hashes on-chain and keep latency-sensitive operations off-chain. Despite these benefits, metadata leakage, maintenance cost and consensus overhead require careful tuning of block parameters, committee structure, and privacy mechanisms [16, 21].

2.4. Assumption Violations and Practical Limitations of Matrix Completion for Channel Allocation

2.4.1. Low-Rank Structure, Incoherence and Sampling Bias

Matrix completion assumes that the spatio-temporal occupancy matrix is approximately low rank, incoherent and sampled nearly uniformly. In practical CRNs, heterogeneous users, overlapping services, clustered sensing and policy or energy constraints often violate these assumptions, increasing effective rank and sampling bias and thereby degrading reconstruction quality and channel allocation accuracy [23, 24, 32].

2.4.2. MNAR Acquisition and Temporal Drift

Performance further deteriorates when observations are missing not at random, such as when nodes avoid busy bands or reduce sensing under low battery conditions. In such cases, standard matrix completion becomes biased unless the missingness process is explicitly modeled [33]. Moreover, PU activity is inherently non-stationary due to daily usage patterns, events, and policy changes, so reconstructions based on past data may lag behind current occupancy. Online and streaming variants, including robust PCA and incremental subspace updates, are therefore necessary but remain challenging to deploy effectively [34].

2.4.3. Adversarial Corruption, Noise, and Quantized Sensing

Matrix completion is also sensitive to corrupted reports, measurement noise and coarse sensing. Malicious or faulty SUs can inject sparse outliers that distort factor estimates and mislead allocation decisions, even when only a small fraction of observations is compromised [22, 35]. In addition, CRN sensing often occurs at low SNR with quantized or binary decisions, whereas standard formulations assume real-valued observations with mild noise. One-bit sensing and one-bit matrix-completion perspectives therefore better reflect practical recovery limits in such settings [36, 37].

2.4.4. Scalability and Real-Time Constraints

Wideband CRN sensing is naturally three-dimensional across space, frequency and time. Flattening this structure into a matrix may discard useful correlations, whereas tensor-based alternatives improve fidelity at substantially higher computational cost. Since SVD or factor updates must finish within strict MAC-layer deadlines, latency and resource overhead remain major barriers to real-time deployment [24, 32, 36].

2.5. Motivation

Designing an energy-efficient spectrum-allocation protocol for resource-constrained IoT devices requires balancing channel utilization, sensing overhead, security and robustness. This balance depends on accurate prediction of future channel availability, which relies on the availability and integrity of SSDs. Existing schemes such as PROLEMus use Model Predictive Control (MPC) for proactive channel allocation, while ProBLeSS improves security through blockchain-backed reporting. Similarly, ProMETHEUS combines secure reporting with lightweight prediction and performs well under SSDF attacks [14, 15, 16]. However, these methods still require dense sensing/report exchange, blockchain maintenance or stronger coordination, which become costly in large-scale IoT CRNs [21]. Although trust and blockchain mechanisms improve integrity, they also increase control traffic, storage and computation cost [21, 30]. Therefore, improving SSD integrity and availability while reducing sensing and reporting frequency can enable a lighter prediction pipeline and a smaller attack surface.

2.6. Novelty and Technical Contribution

This work presents ProCLAMUS as a lightweight integration of established techniques for proactive and secure spectrum allocation in IoT oriented CRNs. Prior work has studied predictive channel allocation, blockchain assisted secure sharing, trust-aware SSDF mitigation and matrix-completion-based recovery [14, 15, 16, 21, 22, 23, 29, 30]. Therefore, novelty is not claimed for matrix completion, short-horizon prediction or majority-vote fusion individually. Instead, ProCLAMUS contributes through their low-overhead integration for sparse, adversarial and resource-constrained CRN-IoT operation.

Against this background, the main technical contributions of ProCLAMUS are as follows:

- *Robust proactive spectrum allocation under noisy and incomplete sensing data.* ProCLAMUS reconstructs sparse or SSDF-corrupted SSD matrices using matrix completion and applies short-horizon prediction to sustain allocation accuracy. It maintains acceptable channel utilization up to a sensing-interval multiplication factor of 2.7, i.e., 270 ms and tolerates up to 50% malicious SUs, showing graceful degradation through sparse recovery, majority fusion and localized decisions, as shown in Figure 3, Figure 6, Subsection 6.9, and Subsection 7.8.
- *Consistent operation across varied CRN-IoT conditions.* The proposed scheme performs reliably under rank variation, where matrix rank reflects changes in the number of PUs/SUs, SU spacing, cluster structure, malicious-node presence and mobility. This confirms that the integration remains effective across dynamic and heterogeneous operating conditions, as shown in Figure 7 and discussed in Subsection 7.7.
- *Energy-efficient and attack-aware operation.* By down-weighting inconsistent reports and reducing active sensing frequency, ProCLAMUS lowers sensing overhead and radio energy consumption to 16.8×10^{-3} J/s while reducing SSDF attack exposure. Increasing the sensing interval further lowers data exchange and shrinks the adversarial surface without compromising overall superiority to baseline methods. In this respect, ProCLAMUS achieves higher channel utilization and prediction accuracy while reducing sensing delay, backoff rate, energy use and SSDF exposure compared with Conventional MV-CSS, PROLEMus, ProBLeSS, and ProMETHEUS [14, 15, 16, 17].
- *Decentralized and scalable decision-making without a fusion center.* Each SU performs lightweight local prediction and allocation, which reduces CCC overhead, avoids single points of failure and improves scalability for distributed CRN-IoT deployment.
- *Practical empirical validation in an IoT-oriented CRN testbed.* The complete design is validated in a Raspberry Pi 4 and LimeSDR-based implementation including malicious SUs, demonstrating feasibility beyond simulation.

Thus, the novelty of ProCLAMUS lies in its low-overhead integration, robust operating point, decentralized realization and empirical validation rather than in an individual algorithmic breakthrough.

3. Matrix Completion

3.1. Netflix Problem

The Netflix problem is a famous challenge in collaborative filtering and recommender systems. This was first publicised through the Netflix Prize competition launched in 2006 [38]. Netflix released a dataset of over 100 million movie ratings from nearly 500,000 users and challenged participants to improve the accuracy of its movie recommendation algorithm (Cinematch) by at least 10 percent, measured using RMSE. The Netflix problem has had a major effect on ML, optimization and recommender system research, influencing further work in matrix completion.

3.2. Matrix Completion for Prediction

Matrix Completion (MC) is a key technique for reconstructing sparse data in low-rank matrices and is widely used in wireless communications, traffic monitoring, computer vision and recommender systems. Recent work highlights its value in real-world recovery tasks. Wang et al. [39] combined MC with cache placement for content popularity prediction in mobile networks, addressing storage limits and prediction uncertainty. Chen et al. [40] used low-rank short term stability in sensor networks to reduce recovery errors. Li et al. [41] proposed clustering-based interpolation using correlations among neighboring sensor nodes to improve recovery for inactive nodes. Xie et al. [22] introduced a two-phase PCA based method for successive data corruption. In CRNs, MC supports recovery of sparse channel-status information for spectrum allocation.

3.3. Attack Surface Reduction by Matrix Completion

SSDs aggregated across users and channels typically exhibit low intrinsic dimensionality because spatial correlation causes nearby SUs to observe similar activity, channel states change gradually over time and busy/idle transitions are sparse relative to sampling. Hence, the per-slot SSD matrix is approximately low rank, and matrix completion can recover the global state from a subset of entries rather than full reporting. If the matrix has rank r and satisfies standard incoherence assumptions, recovery is feasible with $O(r(M+N)\log(M+N))$ measurements instead of MN. This reduces the attack surface in two ways:

- *Fewer Exposure Points.* Only a subset of SUs report SSDs in each slot, reducing opportunities for adversaries to inject falsified sensing information on the control channel. With fewer reports, attackers must compromise a larger fraction of active transmissions to meaningfully bias the fused decision.
- *Constrained Error Propagation.* Matrix completion enforces a low-rank structure, keeping SSD entries globally consistent and suppressing outliers that violate this pattern. As a result, isolated malicious SSDs have limited influence. Only large, coordinated attacks can imitate a valid low-rank pattern, while majority-vote fusion further limits local errors from becoming network-wide distortions.

3.4. Proposed Approach

This paper proposes ProCLAMUS, an efficient and secure distributed spectrum allocation protocol for IoT-based CRNs that integrates matrix completion with sliding-window prediction. Its distributed design exploits CRN delay tolerance by executing allocation locally at SUs, eliminating dependence on a fusion center and avoiding a potential choke point. ProCLAMUS uses infrequent sensing to reduce energy consumption, computation and attack exposure, but this also creates sparse SSD matrices due to fewer measurements and limited inter-SU exchange in large heterogeneous deployments. To address sparsity, ProCLAMUS applies nuclear-norm matrix completion to reconstruct missing channel-state data from partially observed SSDs. It then builds an N-slot channel history by fusing completed and binarized sensing data across timeslots. Sliding-window prediction uses this history to guide allocation, ensuring channels predicted as idle are assigned to trustworthy SUs. Experiments show improved prediction accuracy, spectrum utilization, and resilience against malicious SUs compared with PROLEMus [14], ProBLeSS [15], and ProMETHEUS [16]. The algorithm architecture is shown in Figure 2.

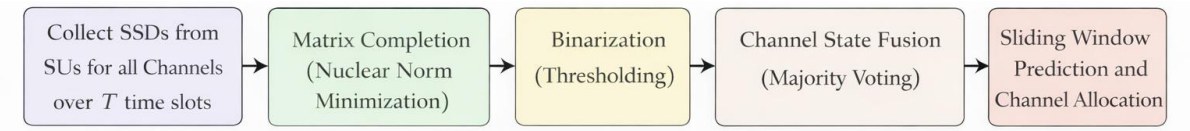


Fig. 2. ProCLAMUS: Flowchart.

Algorithm 1. ProCLAMUS: A Matrix Completion Based Proactive Spectrum Allocation Protocol for Security in IoT Based CRNs

Require: M: Number of SUs, N: Number of Channels, T: History window, K: Prediction horizon, SSD_vectors: Sensing reports ($N \times M \times T$), threshold, sliding window

Ensure: Channel allocation plan for next K slots

Step 1: SSD Matrix Formation – Aggregate sensing reports from all SUs over T slots to form the observation tensor SSD_vectors.
Step 2: Matrix Completion – For each slot t, construct $Y_t \in \mathbb{R}^{N \times M}$ and apply nuclear-norm minimization to recover missing or corrupted sensing entries.
Step 3: Binarization – Threshold the completed matrix Y_t to obtain binary decisions $X_{\text{bin}}[n, i, t] \in \{0, 1\}$ representing channel idle/busy states.
Step 4: Fusion to Obtain Channel State History – Fuse SU-level binary reports using majority or majority-based voting to generate the channel state history $H[n, t]$.
Step 5: Sliding Window Prediction – Predict future channel states $P[n, k]$ using the most recent sliding window entries of $H[n]$ for each channel.
Step 6: Channel Allocation – Allocate channels with predicted idle state ($P[n, k] = 0$) to available SUs, ensuring collision-free and energy-efficient access.

4. Proposed ProCLAMUS Algorithm

4.1. System Model and Inputs

Let there be M SUs sensing N channels over T time slots. The SSD vectors reported by SUs may have missing entries. Let W be the sliding window size for prediction, and K the number of future time slots to predict.

4.2. Step 1: SSD Matrix Formation

The ProCLAMUS algorithm begins by generating SSD matrix for each time slot. During each time instance t, SUs perform spectrum sensing across N channels. Each SU submits a SSD vector indicating the perceived channel status. SSD vectors from SUs are compiled into an incomplete matrix$M$ $Y(t) \in R^{N \times M}$, where missing entries may be a result of transmission failures, path loss etc. The resulting matrices often exhibit a low-rank structure due to spatio-temporal correlations, thus suitable for low-rank matrix completion. Mathematically, for each time slot $t \in \{1, 2, \ldots, T\}$:

- The SSD vectors $\boldsymbol{y}_i^{(t)}$ are received from each SU $i = 1, \ldots, M$, each of length N.
- The data is compiled as an incomplete SSD matrix $Y^{(t)} \in R^{N \times M}$, where $Y_{ni}^{(t)}$ is the observation of SU i on channel n at time t.

4.3. Step 2: Matrix Completion via Nuclear Norm Minimization

To recover missing entries and complete the sparse matrix, a matrix completion step is performed using nuclear norm minimization. For each matrix $Y(t)$, the optimization problem seeks a completed matrix $\hat{Y}(t)$ that minimizes the nuclear norm $\| X \|_*$, subject to preserving the known entries. This process creates a fully populated matrix that reconstructs the channel occupancy information of all SUs. Singular value thresholding is used for decision making. Mathematically, For each $Y^{(t)}$, the convex optimization problem is solved:

$$\min_X \| \boldsymbol{X} \|_* \quad \text{subject to} \quad X_{ni} = Y_{ni}^{(t)}, \ \forall (n, i) \in \Omega \tag{1}$$

where $\|\cdot\|_*$ is the nuclear norm and Ω is the set of observed entries. The completed matrix is denoted $\hat{Y}^{(t)}$.

4.4. Step 3: Binarization

In the matrix $\hat{Y}^{(t)}$ obtained, binarization is carried out using a threshold τ. Each entry in the matrix is compared with τ. The values above or equal to τ are assigned 1 (busy), and values below τ are set to 0 (idle). The output is a binary SSD matrix $\tilde{Y}(t)$, which simplifies the sensing information for the fusion process.

$$\tilde{Y}_{ni}^{(t)} = \begin{cases} 1, & \text{if } \hat{Y}_{ni}^{(t)} \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Thus, the output is a complete binary SSD matrix $\hat{Y}^{(t)}$ for each t.

4.5. Step 4: Channel State History Construction

The binary sensing outcomes for each channel are fused to arrive at a decision per channel per time slot. This fusion is based on majority voting, where each SU contributes to the final decision. This process results in a fused state vector $\boldsymbol{s}(t)$ for each time t. These vectors are then concatenated across the recent T time slots to build the channel state matrix $\boldsymbol{S} \in \{0,1\}^{N \times T}$, where each row represents a channel and each column a time slot. For each channel n, aggregate the M SU reports using a fusion rule (e.g., majority voting):

$$s_n^{(t)} = \text{Fusion}\left(\{\tilde{Y}_{ni}^{(t)} : i = 1, \ldots, M\}\right) \tag{3}$$

Construct the channel state history matrix $\boldsymbol{S} \in \{0,1\}^{N \times T}$, where $S_{n,t} = s_n^{(t)}$.

4.6. Step 5: Sliding Window Prediction

From the fused matrix $\boldsymbol{S}$, the future channel states are predicted using sliding window mechanism. For each channel, the most recent W entries are extracted, forming a window of past observations. The next K time slots are predicted using simple frequency-based longest match pattern (LMP). In frequency prediction, the mode of the window is used to predict the next state. This temporal prediction allows for proactive channel selection. For each channel n, extract the latest W channel states:

$$[S_{n,T-W+1}, \ldots, S_{n,T}] \tag{4}$$

Predict the channel state for the next K slots ($k = 1, \ldots, K$) using the sliding window. For example, with frequency match:

$$\hat{s}_n^{(T+k)} = \text{mode}([S_{n,T-W+1}, \ldots, S_{n,T}]) \tag{5}$$

4.7. Step 6: Channel Allocation

The predicted channel states over the next K time slots guide the spectrum allocation decision. Channels forecasted to be idle (i.e., $\hat{s}_n^{(T+k)} = 0$) are shortlisted, and the allocation is based on round-robin policy. The operational workflow of the proposed algorithm is summarized in Algorithm 1. The algorithm sequentially details the major functional stages—SSD matrix formation, matrix completion, binarization, majority-based fusion, short-horizon prediction and channel allocation.

5. Experimental and Emulation setup

5.1. System Model

We evaluate ProCLAMUS on a Raspberry Pi 4 based CRN testbed. Each node uses a Broadcom BCM2711 quad-core Cortex-A72 processor (1.8 GHz), 8 GB LPDDR4 memory, IEEE 802.11ac connectivity and a LimeSDR Mini for RF sensing. The network comprises 40 Secondary Users (SUs), 400 Primary Users (PUs) and 400 licensed channels, representing a smart-building IoT scenario with fire alarm and building management subsystems. Each SU is single-radio and half-duplex. SUs exchange compressed SSDs over a common control channel (CCC) every 100 ms, while application layer telemetry is transmitted every 3 s. For fairness, all protocols are evaluated under the same sensing history length $T = 10$ and prediction horizon $K = 3$; for schemes employing window-based prediction, a common candidate set $W \in \{3, 4, 5\}$ is used. We first compare all methods under identical window settings and then report a best-window operating point for each protocol, selected using the same criterion, namely the minimum-energy configuration whose channel utilization remains at least 98% of that protocol's maximum. ProCLAMUS applies nuclear-norm matrix completion, threshold-based binarization ($\tau = 0.5$), channel-state fusion, short-horizon prediction and conservative channel allocation. Performance is compared against PROLEMus [14], ProBLeSS [15], ProMETHEUS [16] and conventional MV-CSS [17]. In MV-CSS, each SU submits a binary local decision and the final channel state is obtained by majority voting; no matrix completion, prediction, trust weighting, blockchain validation or malicious-SU filtering is applied, making it a basic CSS reference for isolating the benefit of the proposed design under SSDF attacks.

5.2. Traffic Model

PU activity is modeled as a two-state continuous-time Markov process with exponential holding times in the ON (busy) and OFF (idle) states. The corresponding cumulative distribution functions are given by:

$$F_{\text{Trans}}(t) = 1 - e^{-\kappa_1 t}, \qquad F_{\text{Idle}}(t) = 1 - e^{-\kappa_2 t} \tag{6}$$

The parameters are chosen as $\kappa_1 \in [0.912, 0.971]$ and $\kappa_2 \in [0.063, 0.223]$ to emulate bursty and stochastic PU activity patterns typical of real-world CRN deployments. SUs operate under duty-cycling and rely on the CCC for exchanging compact sensing data packets.

Table 1. Implementation and Reproducibility Parameters.

Parameter	Setting used
Testbed platform	Raspberry Pi 4 + LimeSDR Mini
Network size	40 SUs, 400 PUs, 400 channels
CCC reporting interval	100 ms
Telemetry interval	3 s
Malicious SU fraction	5%–50% (step 5%)
SSD matrix per slot	$Y^{(t)} \in \mathbb{R}^{N \times M}$
Matrix-completion model	Nuclear-norm minimization
Solver family	Singular-value-thresholding-based recovery
Binarization threshold τ	0.5
Fusion rule	Majority voting
History length T	10 slots
Sliding window W	{3,4,5}
Best-window rule	Min. energy with CU ≥ 98% of protocol maximum
Prediction horizon K	3 future slots
Prediction rule	Sliding-window frequency/LMP; next state = mode of latest W states
Allocation condition	Allocate channels predicted idle
Allocation policy	Conservative allocation
Tie-breaking rule	Round-robin

5.3. Threat Model

The robustness of ProCLAMUS is assessed under SSDF attacks, where compromised SUs continue cooperative sensing and CCC based SSD reporting. The malicious-SU fraction is varied from 5% to 50%. Let $y_{i,c}(t) \in \{0,1\}$ denote the actual sensing decision of SU i on channel c at epoch t, and $\tilde{y}_{i,c}(t)$ the reported value. For legitimate SUs, $\tilde{y}_{i,c}(t) = y_{i,c}(t)$; malicious SUs falsify $\tilde{y}_{i,c}(t)$ before fusion. Three SSDF strategies are considered: bit-flipping, where $\tilde{y}_{i,c}(t) = 1 - y_{i,c}(t)$; random bit-flipping, where $\tilde{y}_{i,c}(t) = y_{i,c}(t) \oplus z_{i,c}(t)$, $z_{i,c}(t) \sim \text{Bernoulli}(p_f)$; and vacant-channel flipping, where idle channels $c \in \mathcal{V}(t)$ are falsely reported as busy, i.e., $\tilde{y}_{i,c}(t) = 1$. Falsification is persistent at every sensing epoch, collusion among malicious SUs is allowed, and the adversary does not manipulate PU activity, CCC operation or hardware integrity.

5.4. Rank Sensitivity Setup

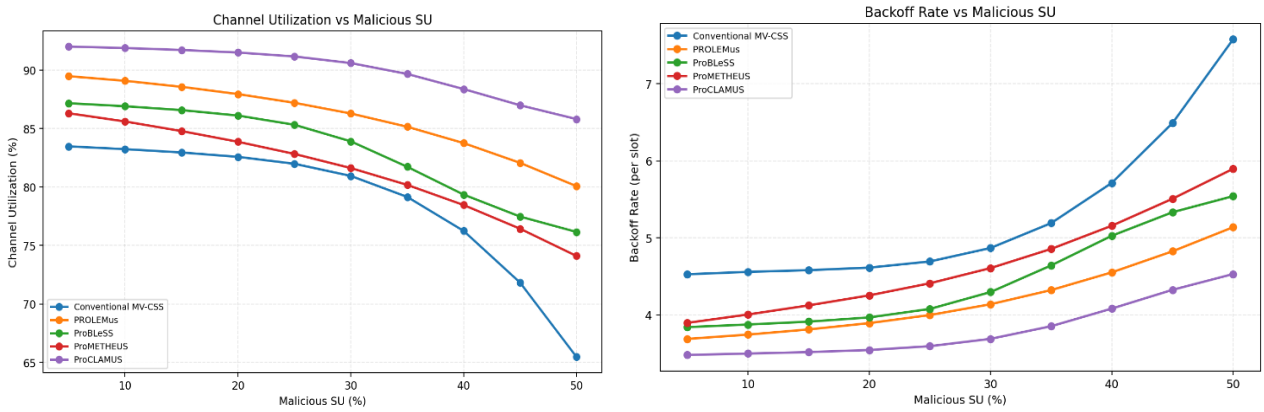
To analyze matrix-rank impact on ProCLAMUS performance, we conduct a sensitivity study by varying the assumed rank in the matrix completion stage as $r \in \{4, 8, 12, 16, 20\}$. Matrix rank emulates multiple CRN conditions, including numbers of PUs/SUs, SU inter-distance, clusters, malicious nodes and node mobility. Lower r values impose stronger correlation assumptions and higher regularization, while higher values provide greater flexibility in representing channel-state variations. Other system parameters remain unchanged to isolate rank effects. Performance is evaluated using channel utilization, prediction accuracy, RMSE, energy consumption and adversarial robustness, capturing trade-offs among reconstruction fidelity, adaptability and SSDF resilience.

5.5. Robustness Validation

Robustness of ProCLAMUS is validated through two complementary experiments. In the first one, the nominal sensing interval is set to $T_s = 100$ ms and scaled by a factor $k \in \{1.0, 1.2, 1.4, \ldots, 3.0\}$, such that $T_s = k.100$ ms. All other parameters are kept fixed and the protocol is evaluated under a constant adversarial setting of 25% malicious SUs. Channel utilization and backoff rate are measured to study the effect of stale sensing. In the second experiment, the sensing-interval factor is fixed at $k = 2.7$, while the percentage of malicious SUs is varied from 5% to 75% in steps of 5%. Channel utilization is measured to examine the effect of increasing adversarial participation under a relaxed sensing regime.

5.6. Metrics and Definitions

Let t_{run} be the experiment duration, t_{busy} the aggregate busy time (PU+SU) on the selected channel(s), N_{backoff} the number of backoff events, SU_{busy} the total SU transmit time, N_{fail} the number of channels probed before successes and SU_{pkt} the number of SU packets.



(a) Channel Utilization vs. Malicious SU (%). (b) Backoff Rate vs. Malicious SU (%).

Fig. 3. Performance comparison with Conventional MV-CSS, PROLEMus, ProBLeSS, ProMETHEUS and ProCLAMUS for Channel Utilization and Backoff Rate.

Table 2. Average Performance Comparison Table of Protocols.

Protocol	CU (%)	BR (s^{-1})	SD (ch/succ.)	ACC	Energy ($\times 10^{-3}$ J/s)	Mem (%)
Conventional MV-CSS	78.78	5.28	0.85	0.76	21.57	20.47
PROLEMus	85.95	4.21	0.64	0.85	30.04	31.90
ProBLeSS	83.06	4.45	0.68	0.83	32.73	35.12
ProMETHEUS	81.41	4.67	0.70	0.81	35.40	37.35
ProCLAMUS	*89.96*	*3.81*	*0.58*	*0.88*	*16.82*	*21.45*

$$\text{Channel Utilization (CU) [\%]\&} = \frac{t_{\text{busy}}}{t_{\text{run}}} \times 100 \tag{7}$$

$$\text{Backoff Rate (BR) [s}^{-1}\text{]\&} = \frac{N_{\text{backoff}}}{t_{\text{run}}} \tag{8}$$

$$\text{Sensing Delay (SD) [ch/succ]\&} = \frac{N_{\text{fail}}}{SU_{\text{pkt}}} \tag{9}$$

The Sensing delay is the average number of channels probed before a successful transmission. Energy consumption is modeled as the aggregate per-second cost of sensing, channel switching, CCC communication and local computation.

Let $e_s = 1.31 \times 10^{-4}$ J denote the energy consumed per sensing operation, $e_{sw} = 10^{-5}$ J the energy per channel switch, ρ_{base} the baseline sensing rate, λ_{tot} the aggregate SU packet arrival rate and μ_{sw} the switching multiplier. The total energy expenditure is approximated as

$$E_{\text{tot}} \approx e_s(\rho_{\text{base}} + \lambda_{\text{tot}} SD) + e_{sw}\mu_{sw} BR + e_{tx} N_{tx}^{\text{CCC}} + e_{rx} N_{rx}^{\text{CCC}} + P_{\text{cpu}} T_{\text{comp}}, \tag{10}$$

where

$$T_{\text{comp}} = t_{\text{mc}} + t_{\text{fus}} + t_{\text{pred}} + t_{\text{alloc}} \tag{11}$$

Here, SD and BR denote the sensing delay and backoff rate, respectively. The terms e_{tx} and e_{rx} represent the transmission and reception energy per CCC packet, while N_{tx}^{CCC} and N_{rx}^{CCC} denote the corresponding numbers of transmitted and received CCC packets. The term $P_{\text{cpu}} T_{\text{comp}}$ captures the local computational energy, with P_{cpu} denoting the average CPU power and T_{comp} comprising the execution times for matrix completion, fusion, prediction and channel allocation. Thus, E_{tot} accounts for the combined sensing, switching, communication and computation costs incurred by each SU.

6. Results

The revised evaluation compares ProCLAMUS with four baselines: Conventional MV-CSS[17], PROLEMus[14], ProBLeSS[15] and ProMETHEUS[16]. Conventional MV-CSS serves as the hard-decision CSS reference, while the other three are recent secure CRN protocols. Average results for 5%–50% malicious SUs are listed in Table 2, and the

corresponding non-linear trends appear in Figures 3, 4, 5, 6 and 7. Conventional MV-CSS is the most vulnerable to SSDF because it relies on binary sensing reports without reconstruction or malicious-report filtering. PROLEMus degrades moderately due to its security-aware learning and allocation design. ProBLeSS and ProMETHEUS offer stronger protection than basic CSS but introduce higher storage, communication, and validation overhead. ProCLAMUS shows the flattest low-attack response and the least degradation at high attack levels, as matrix completion, compact sensing, majority fusion, and conservative allocation together reduce the impact of falsified SSDs.

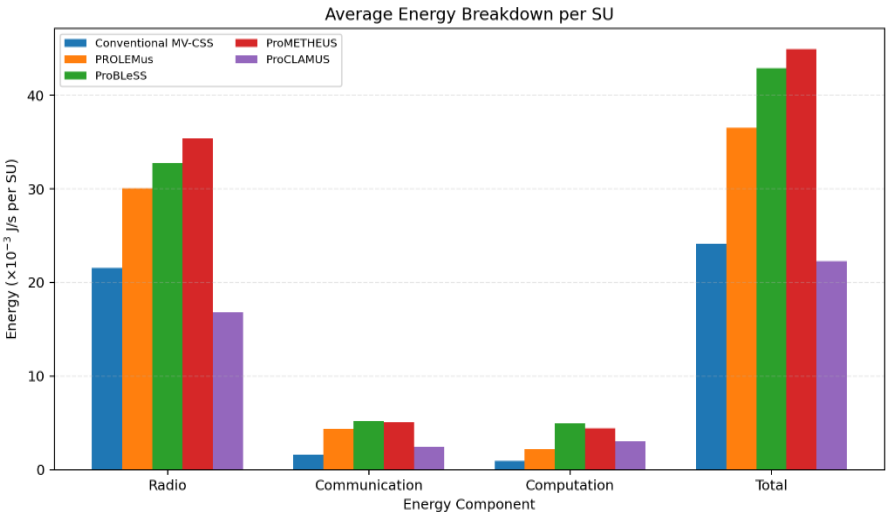


Fig. 4. Average Energy Breakdown per SU for Conventional MV-CSS, ProCLAMUS, PROLEMus, ProBLeSS, and ProMETHEUS, showing Radio, Communication, Computation, and Total Energy Components.

Table 3. Average Energy Breakdown per SU of Protocols.

Protocol	Radio	Communication	Computation	Total
Conventional MV-CSS	21.57	1.58	0.95	24.10
PROLEMus	30.04	4.32	2.20	36.56
ProBLeSS	32.73	5.19	4.94	42.86
ProMETHEUS	35.40	5.09	4.41	44.90
ProCLAMUS	*16.82*	*2.43*	*3.02*	*22.27*

6.1. Channel Utilization vs. Malicious SU (%)

ProCLAMUS sustains the highest channel utilization, averaging 89.96% and degrades more gradually than all baselines as the malicious-SU fraction increases. PROLEMus follows with 85.95% , while ProBLeSS and ProMETHEUS achieve 83.06% and 81.41%, respectively. Conventional MV-CSS gives the lowest average utilization of 78.78%. Its curve remains comparatively stable at very low attack levels but drops sharply at high malicious-SU percentages because majority voting cannot distinguish honest and falsified reports. The results are shown in Figure 3a. The higher utilization of ProCLAMUS confirms that matrix completion before thresholding improves binary recovery and supports more reliable proactive allocation.

6.2. Backoff Rate vs. Malicious SU (%)

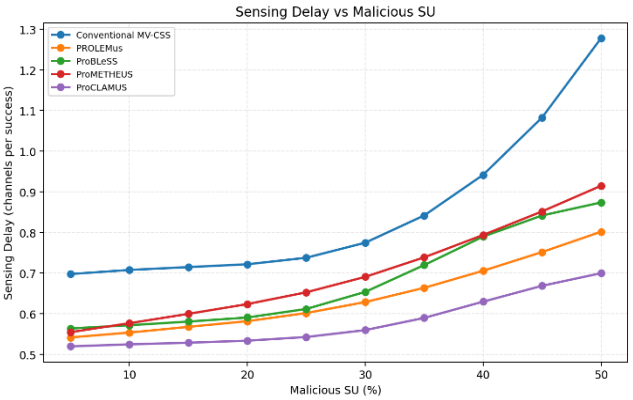
ProCLAMUS attains the lowest backoff rate with an average of $3.81\,s^{-1}$. PROLEMus, ProBLeSS, and ProMETHEUS report higher means of 4.21, 4.45, and $4.67\,s^{-1}$, respectively, while Conventional MV-CSS has the highest average backoff rate of $5.28\,s^{-1}$. MV-CSS grows slowly at first and then increases rapidly when falsified reports dominate the majority decision; ProMETHEUS grows more steadily due to higher validation and coordination cost; ProCLAMUS grows the least because conservative allocation avoids repeated collision-driven retreats. The results are shown in Figure 3b.

6.3. Sensing Delay vs. Malicious SU (%)

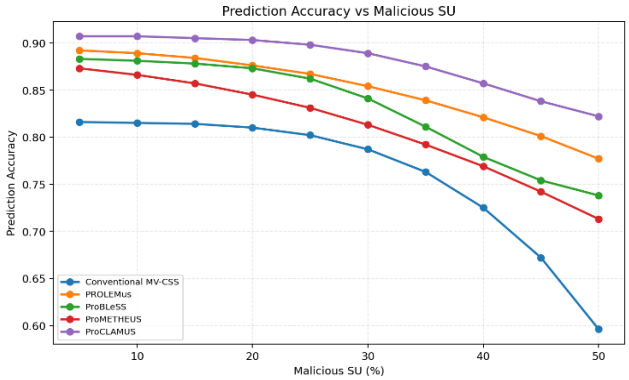
The sensing delay remains lowest for ProCLAMUS with a mean of 0.58 channels/success. PROLEMus, ProBLeSS, and ProMETHEUS show higher means of 0.64, 0.68, and 0.70, respectively, whereas Conventional MV-CSS reaches 0.85 channels/success. The sharp high-attack increase in MV-CSS occurs because more channels are incorrectly classified, forcing additional probing before successful transmission. ProCLAMUS maintains a lower delay by reconstructing sparse SSD matrices and narrowing the probing space before allocation. The results are shown in Figure 5a.

6.4. Prediction Accuracy vs. Malicious SU (%)

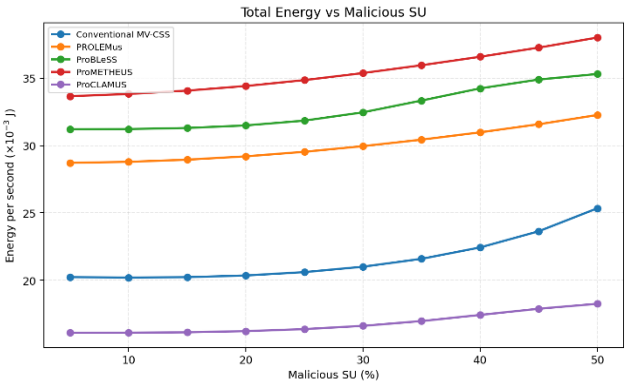
Prediction accuracy declines with increasing malicious inputs for all schemes, but the shape and severity of degradation differ across protocols. ProCLAMUS remains the most accurate with a mean accuracy of 0.88, followed by PROLEMus at 0.85, ProBLeSS at 0.83, ProMETHEUS at 0.81, and Conventional MV-CSS at 0.76. Conventional MV-CSS exhibits the strongest late-stage accuracy collapse because it lacks completion, prediction and malicious-report control. ProCLAMUS shows a flatter early and mid-range response, indicating that completion and majority-vote fusion reduce the propagation of corrupted sensing reports into prediction. The results are shown in Figure 5b.



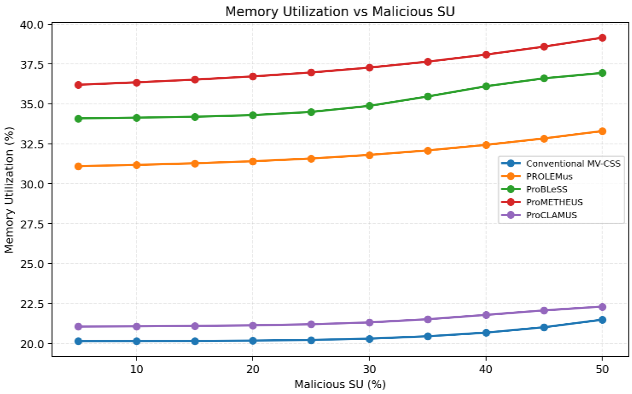
(a) Sensing Delay vs. Malicious SU (%).



(b) Prediction Accuracy vs. Malicious SU (%).



(c) Total Energy per Second ($\times 10^{-3}$ J) vs. Malicious SU (%).



(d) Memory Utilization vs. Malicious SU (%).

Fig. 5. Performance Comparison with Conventional MV-CSS, PROLEMus, ProBLeSS, ProMETHEUS and ProCLAMUS for Sensing Delay, Prediction Accuracy, Total Energy and Memory Utilization.

6.5. Average Energy Breakdown per SU

Table 3 and Figure 4 present the average per-SU energy breakdown. ProCLAMUS achieves the lowest total energy at 22.27×10^{-3} J/s, compared with 24.10×10^{-3} J/s for Conventional MV-CSS, 36.56×10^{-3} J/s for PROLEMus, 42.86×10^{-3} J/s for ProBLeSS, and 44.90×10^{-3} J/s for ProMETHEUS. Although MV-CSS has small communication and computation costs, its higher sensing and collision-related overhead prevent it from outperforming ProCLAMUS. Figure 5c further shows that ProCLAMUS has the slowest energy growth as malicious participation increases.

6.6. Memory Utilization vs. Malicious SU (%)

Memory utilization increases with malicious-SU intensity, but the overhead source differs across protocols. Conventional MV-CSS has the smallest average memory footprint (20.47%) because it stores only simple binary reports. However, this low memory usage comes at the cost of poor accuracy, utilization and backoff performance. ProCLAMUS uses slightly more memory (21.45%) because of compact matrix-completion and prediction state, but it remains far below PROLEMus (31.90%), ProBLeSS (35.12%) and ProMETHEUS (37.35%). The results in Figure 5d therefore show that ProCLAMUS achieves a better security-performance-memory trade-off than both basic CSS and the secure protocol baselines.

6.7. Effect of Rank on Accuracy and RMSE

Figure 7(a)–(b) shows the effect of reconstruction rank on prediction accuracy and RMSE. Accuracy improves from low to moderate rank, reaches a maximum of 89.2% at $r = 12$, and then declines slightly at higher ranks. In contrast, RMSE decreases initially, attains its minimum value of 0.21 at $r = 12$, and rises thereafter. These trends indicate that very low rank leads to oversmoothing, whereas high rank increases sensitivity to noise and adversarial perturbations.

6.8. Effect of Rank on Channel Utilization and Robustness

Figure 7(c)–(d) presents channel utilization and robustness across rank values. Channel utilization follows an inverted-U trend, peaking at 90.4% for $r = 12$. Lower ranks lead to conservative allocation due to oversmoothing, while higher ranks reduce efficiency through increased prediction errors. Robustness decreases monotonically from 0.82 at $r = 4$ to 0.58 at $r = 20$, showing that higher-rank reconstructions retain a larger fraction of adversarial distortions.

6.9. Robustness Validation

Figure 6 summarizes two complementary robustness experiments. In the first, the sensing interval is increased through the factor k under a fixed adversarial setting of 25% malicious SUs. Channel utilization remains nearly flat for $1.2 \leq k \leq 2.0$, declines approximately linearly for $2.0 < k \leq 2.7$, and drops more sharply thereafter, while backoff exhibits the opposite trend. In the second experiment, k is fixed at 2.7 and the malicious-SU fraction is varied from 5% to 75%. Channel utilization remains nearly flat at low attack levels, declines approximately linearly at moderate attack levels, and drops more sharply beyond 50%. In particular, utilization decreases from 84.7% at 5% malicious SUs to 75.0% at 45%, corresponding to a drop of about 9.7%. The empirical trends reported above are consistent with Lemmas 8.1–8.3, which explain the gains in binary recovery after completion, robustness under SSDF, and the rank-dependent trade-off between reconstruction fidelity and adversarial resilience.

7. Performance Analysis and Theoretical Interpretation

7.1. Baseline Diversity and Interpretation

The expanded comparison now includes a basic non-secure CSS baseline, Conventional MV-CSS [17], in addition to the secure CRN baselines PROLEMus [14], ProBLeSS [15] and ProMETHEUS [16]. This broader baseline set separates three effects: the limitation of simple hard-decision fusion, the overhead of secure protocol mechanisms and the contribution of ProCLAMUS's matrix-completion-based proactive allocation. The non-linear trends in Figures 3 and 5 show that the protocols do not fail at the same rate as adversarial participation increases.

7.2. Channel Utilization and Attack Sensitivity

ProCLAMUS sustains the highest channel utilization (Figure 3a), averaging 89.96%. PROLEMus, ProBLeSS and ProMETHEUS average 85.95%, 83.06%, and 81.41%, respectively, while Conventional MV-CSS averages 78.78%. The MV-CSS curve has a threshold-like behaviour: degradation is limited at low attack levels but becomes steep once falsified reports are numerous enough to bias the majority decision. ProCLAMUS avoids this behaviour because completion before thresholding reduces missing and corrupted SSD effects before binary fusion. This is consistent with Lemma 8.1, which shows that completion improves binary recovery prior to thresholding.

7.3. Collisions, Backoff, and Probing

Backoff rate and sensing delay reveal the strongest difference between simple CSS and ProCLAMUS. Conventional MV-CSS has the highest backoff ($5.28\,\mathrm{s}^{-1}$) and sensing delay (0.85 channels/success), indicating frequent wrong channel decisions and repeated probing. PROLEMus, ProBLeSS and ProMETHEUS reduce this effect but still incur higher collision or validation overhead than ProCLAMUS. ProCLAMUS achieves the lowest backoff ($3.81\,\mathrm{s}^{-1}$) and sensing delay (0.58 channels/success). This supports Lemma 8.2, because majority fusion remains reliable when the honest-report margin is preserved by completion and conservative allocation.

7.4. Prediction Accuracy

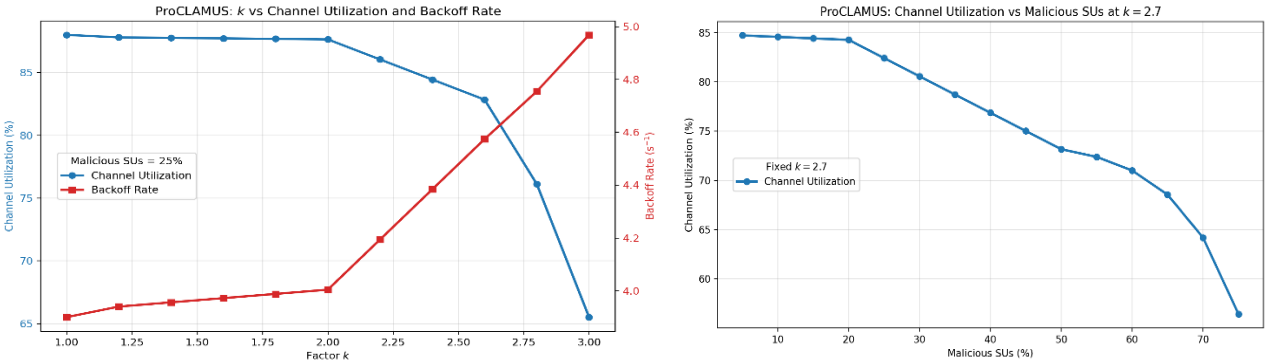
The accuracy trends in Figure 5b show protocol-specific degradation rather than a uniform linear decline. MV-CSS suffers the largest late-stage drop because its decision rule has no mechanism to recover missing SSDs or down-weight inconsistent reports. ProMETHEUS and ProBLeSS retain better accuracy than MV-CSS but are affected by coordination and validation overhead. PROLEMus remains closer to ProCLAMUS but still degrades more sharply at high malicious-SU levels. ProCLAMUS maintains the highest average accuracy (0.88) because reconstructed channel histories reduce the probability that falsified SSDs dominate prediction.

7.5. Energy Efficiency

Table 3, Figure 4, and Figure 5c show that ProCLAMUS has the lowest total energy among all compared schemes. Conventional MV-CSS has lower communication and computation cost than the secure baselines, but its lack of SSDF control increases probing, backoff and sensing related energy. Blockchain or validation heavy protocols, especially ProBLeSS and ProMETHEUS, incur higher radio, communication, and computation costs. ProCLAMUS reduces total energy by combining sparse sensing, compact CCC exchange, lightweight local prediction and conservative allocation. Since Eq. (10) depends on both sensing delay and backoff rate, the lower SD and BR values of ProCLAMUS directly explain its lower energy trend.

7.6. Memory Utilisation

The memory results require a different interpretation after adding MV-CSS. Conventional MV-CSS has the smallest memory footprint because it stores only binary sensing decisions and does not maintain completion or prediction state. However, this low memory use is achieved by sacrificing robustness, accuracy and utilization. ProCLAMUS uses slightly more memory than MV-CSS but much less than PROLEMus, ProBLeSS and ProMETHEUS. Therefore, ProCLAMUS offers the best robustness-aware memory trade-off rather than the absolute lowest memory footprint.



(a) CU and BR vs Sensing Interval Factor k under 25% Malicious SUs.

(b) CU vs Malicious SU percentage at fixed $k = 2.7$.

Fig. 6. Robustness Validation of ProCLAMUS.

7.7. Analysis of Rank Sensitivity

The results show that matrix rank governs how ProCLAMUS responds to varying CRN conditions. At low ranks, stronger regularization suppresses noise and adversarial perturbations, improving robustness but reducing accuracy by discarding valid channel state variations, consistent with Lemma 8.3. As rank increases, the model captures finer dynamics, improving accuracy and utilization up to an optimum near $r = 12$. Beyond this point, robustness declines and RMSE increases as additional noise and malicious perturbations are retained during reconstruction. The U-shaped RMSE and inverted-U channel-utilization trends in Figure 7 confirm an optimal intermediate rank. Together, Lemmas 8.1–8.3 explain why moderate rank provides the best balance between reconstruction fidelity, adaptability, and SSDF resilience. Thus, rank is a critical design parameter influencing ProCLAMUS performance and robustness.

7.8. Analysis: Robustness Validation

The results in Figure 6 show that ProCLAMUS robustness depends on both sensing periodicity and adversarial intensity. In the first experiment, the protocol remains effective over a moderate sensing interval range, as matrix completion and short-horizon prediction compensate for reduced sensing frequency. However, when k exceeds the practical operating region, stale sensing reduces allocation quality, lowering utilization and increasing backoff. In the second experiment, increasing the malicious-SU fraction at fixed $k = 2.7$ causes similar degradation. The nearly flat response at low attack levels indicates tolerance to limited falsification, while the steeper decline at higher attack fractions shows that malicious SSDs increasingly distort fusion and allocation. This agrees with Lemma 8.2, which shows that robustness holds while the honest-report margin dominates but weakens when adversarial participation becomes sufficiently large. Together, these experiments show that ProCLAMUS is robust under moderate sensing relaxation and adversarial participation, but both factors must be jointly controlled to preserve performance.

7.9. Security Implications of Matrix Completion and Sparse Reporting

The security benefit of ProCLAMUS arises from sparse reporting combined with matrix completion. Since dense sensing reports from all SUs are not required at every sensing interval, the amount of exchanged SSDs is reduced, which lowers the number of exposure points through which falsified reports can affect cooperative decisions under SSDF. At the same time, matrix completion reconstructs a more consistent channel-state view from sparse and partially

corrupted observations, improving the reliability of fusion and prediction. Thus, the security implication of matrix completion in ProCLAMUS should be interpreted as a practical reduction in attack surface and error propagation, supported by the robustness results and performance analysis, rather than as a universal formal security guarantee.

8. Theoretical Validation of Robustness

Lemma 8.1 (Completion before thresholding improves binary recovery over naive thresholding). Consider a low-rank SSD matrix $Y \in \mathbb{R}^{N\times M}$ with i.i.d. missing entries of probability p_{drop}. Let $\hat{Y}$ denote the nuclear-norm completion in (1), let $\tau = 0.5$, let $\hat{B} = \mathbf{1}\{\hat{Y} \geq \tau\}$, and let B^{naive} denote the binary matrix obtained by thresholding a naive imputation. Define $R_{\text{comp}}(\tau) := \Pr(\hat{B} \neq B)$ and $R_{\text{naive}}(\tau) := \Pr(B^{\text{naive}} \neq B)$. Then $R_{\text{comp}}(\tau) \leq R_{\text{naive}}(\tau)$, with strict inequality whenever $p_{\text{drop}} > 0$ and the low-rank structure is informative.

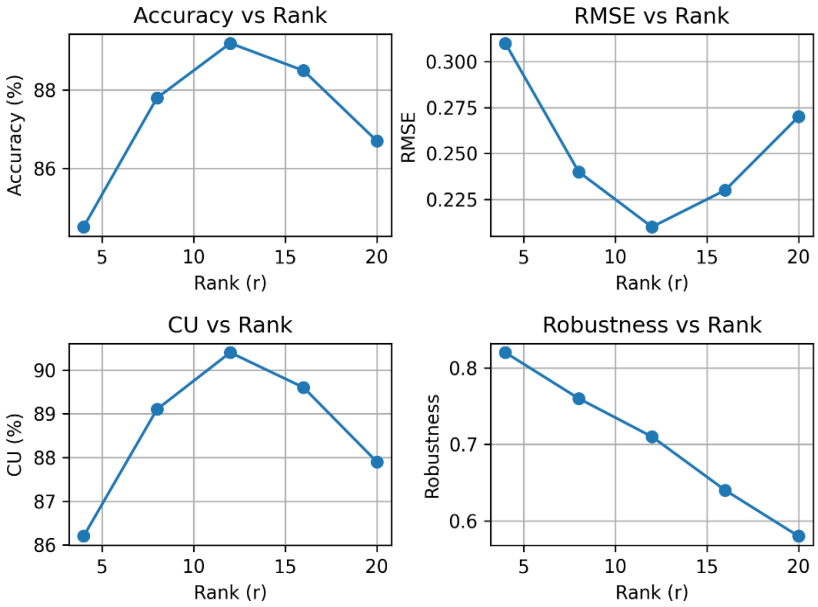


Fig. 7. Performance of ProCLAMUS as a Function of Rank: (a) Accuracy (b) RMSE (c) Channel Utilization and (d) Robustness.

Proof. Under random erasures and low-rank structure, nuclear-norm completion achieves smaller reconstruction error than naive imputation, that is,

$$\mathbb{E} \parallel \hat{Y} - Y \parallel_F^2 \leq \mathbb{E} \parallel Y^{\text{naive}} - Y \parallel_F^2.$$

Thresholding errors occur when estimation noise pushes entries across τ. Hence, a smaller estimation error implies a smaller mis-threshold probability. Therefore, $R_{\text{comp}}(\tau) \leq R_{\text{naive}}(\tau)$, with strict inequality whenever $p_{\text{drop}} > 0$ and the low-rank structure is informative.

Lemma 8.1 explains why completion before thresholding improves binary recovery, which is consistent with the gains in accuracy and RMSE observed in Fig. 7(a)–(b).

Lemma 8.2 (Majority fusion reduces effective fusion error under SSDF attacks). Consider a single channel in a single time slot with *M* sensing users. *Let $p < 1/2$ denote the fraction of malicious users, d denote the report-drop probability, and q denote the probability that a malicious user flips its report. Let $\Delta = (1-p)(1-d) - p$. If $\Delta > 0$, then after completion and majority fusion, the effective fusion error probability satisfies $q_{eff} \leq e^{-2M\Delta^2} + q$.*

Proof. After completion, the expected number of non-dropped honest reports is $M(1-p)(1-d)$, whereas the number of malicious reports is Mp. Hence, the honest reports have a net margin of $M((1-p)(1-d) - p) = M\Delta$. Majority fusion fails only if this margin is overturned. By Hoeffding's inequality, the probability of that event is at most $e^{-2M\Delta^2}$. In addition, malicious flipping contributes at most an additive error probability of q. Therefore, the effective fusion error probability satisfies $q_{\text{eff}} \leq e^{-2M\Delta^2} + q$.

Lemma 8.2 supports the robustness trend in Fig. 7(d) by showing that the effective fusion error remains controlled whenever the honest-report margin is positive.

Lemma 8.3 (The completion rank in ProCLAMUS gives a variance–bias trade-off). Consider the rank-r completion $\hat{Y}^{(r)}$ and the best rank-r approximation Y_r of Y. If

$$\mathbb{E} \parallel \hat{Y}^{(r)} - Y_r \parallel_F^2 \leq \frac{C}{1 - p_{\text{drop}}} r,$$

then, for any threshold margin $\gamma > 0$,

$$\mathbb{E}[\text{Acc}^{(r)}] \geq 1 - \pi_1 q_{\text{busy}}^{\text{eff}} - \pi_0 q_{\text{idle}}^{\text{eff}} - \beta d_A - \frac{2}{\gamma^2 NM} \left(\frac{C}{1 - p_{\text{drop}}} r + \sum_{k>r} \sigma_k^2 (Y) \right).$$

Hence, ProCLAMUS exhibits a variance–bias trade-off: larger r reduces truncation bias, whereas smaller r reduces completion variance.

Proof. The total reconstruction error satisfies

$$\parallel \hat{Y}^{(r)} - Y \parallel_F^2 \leq \parallel \hat{Y}^{(r)} - Y_r \parallel_F^2 + \parallel Y - Y_r \parallel_F^2, \qquad \parallel Y - Y_r \parallel_F^2 = \sum_{k>r} \sigma_k^2 (Y).$$

Using the thresholding-error bound $2/(\gamma^2 NM)$ and the assumption

$$\mathbb{E} \parallel \hat{Y}^{(r)} - Y_r \parallel_F^2 \leq \frac{C}{1 - p_{\text{drop}}} r,$$

Eq. (8) gives the stated lower bound. This proves the variance–bias trade-off.

Lemma 8.3 formalizes the rank-sensitive behavior in Fig. 7, including the gain at moderate rank and the degradation at higher rank due to the variance–bias trade-off.

9. Conclusion

This paper presented ProCLAMUS, the Matrix Completion Based Proactive Spectrum Allocation Protocol for Security of IoT Based CRNs, which integrates nuclear-norm matrix completion with short-horizon sliding-window prediction in a distributed, fusion-center-free architecture. In simulations with 40 SUs, 400 PUs, 400 channels and 5–50% malicious SSDs, *ProCLAMUS* consistently outperformed PROLEMus [14], ProBLeSS [15], ProMETHEUS [16] and the MV-CSS [17]. It achieved the strongest energy-performance-robustness trade-off, sustaining the highest channel utilization (avg. 89.96%), lowest backoff rate (avg. 3.81 s^{-1}), lowest sensing delay (avg. 0.58 channels/success), and lowest energy consumption (avg. 16.82×10^{-3} J/s), yielding a 44–53% reduction over the secure protocol baselines. Although its memory footprint is slightly higher than that of simple MV-CSS, it remains substantially lower than secure protocol baselines, averaging 21.45% (~33–43% lower). These gains arise from combining sparse sensing, matrix completion, majority-vote fusion, conservative allocation, and recency-aware prediction, which jointly limit overhead, reduce exposure points, and constrain error propagation under malicious-SU pressure. Its decentralized design avoids fusion-center dependence, improving scalability and resilience. Overall, ProCLAMUS provides a robust, low-overhead solution that scales favorably in noisy and adversarial cognitive radio environments.

All the Declarations and Statements

Author Contributions Statement

Charanjeet Singh Chahil – Conceptualization, Methodology, Data Curation, Software implementation, Writing Original Draft, Writing Review and Editing.
Milan Patnaik – Supervision, Conceptualization, Methodology, Data Curation, Software Implementation, Writing Review and Editing.
Deepali Gupta – Supervision, Writing Review and Editing.
Vashek Matyas – Supervision, Writing Review and Editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This work was partly supported and funded by the Cybersecurity Excellence Hub in Estonia and South Moravia

(CHESS, 101087529).

Data Availability Statement
The dataset used for analysis is available from the corresponding author upon reasonable request.

Ethical Declarations
The authors declare that this article does not contain any studies with human participants or animals.

Acknowledgment
The authors would like to thank colleagues and reviewers for their valuable comments and suggestions that helped improve the quality of this manuscript.

Declaration of Generative AI in Scholarly Writing
During manuscript preparation, the authors used generative AI tools to assist with language editing and to improve text clarity.

Abbreviations
The following abbreviations are used in this manuscript:

CRN - Cognitive Radio Network
SU - Secondary User
PU - Primary User
SSDF - Spectrum Sensing Data Falsification
PUEA - Primary User Emulation Attack
SSD - Spectrum Sensing Data
MC - Matrix Completion
CSS - Cooperative Spectrum Sensing
CU - Channel Utilization
BR - Backoff Rate
SD - Sensing Delay
ACC - Prediction Accuracy
RMSE - Root Mean Square Error
EWMA - Exponentially Weighted Moving Average

References

[1] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, 1999.
[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2006.
[3] Federal Communications Commission, “Spectrum Policy Task Force Report,” ET Docket No. 02-135, Nov. 2002.
[4] M. A. McHenry and D. McCloskey, “Spectrum occupancy measurements: Chicago, Illinois, November 16–18, 2005,” Shared Spectrum Company Tech. Rep., 2005.
[5] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, 2005.
[6] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, “CRAHNs: Cognitive radio ad hoc networks,” *Ad Hoc Netw.*, vol. 7, no. 5, pp. 810–836, 2009.
[7] K. Tumuluru, P. Wang, and D. Niyato, “A survey on spectrum sensing algorithms for cognitive radio,” in *Proc. IEEE ICCS*, 2010, pp. 1–5.
[8] F. F. Digham, M.-S. Alouini, and M. K. Simon, “On the energy detection of unknown signals over fading channels,” *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 21–24, 2007.
[9] G. Ganesan and Y. Li, “Cooperative spectrum sensing in cognitive radio networks,” in *Proc. IEEE DySPAN*, 2005, pp. 137–143.
[10] T. C. Clancy and N. Goergen, “Security in cognitive radio networks: Threats and mitigation,” in *Proc. IEEE CrownCom*, 2008, pp. 1–8.
[11] R. Chen, J.-M. Park, and J. H. Reed, “Defense against primary user emulation attacks in cognitive radio networks,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 25–37, 2008.
[12] Y.-C. Liang, Y. Zeng, E. Peh, and A. T. Hoang, “Sensing-throughput tradeoff for cognitive radio networks,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326–1337, 2008.
[13] Y. Ding *et al.*, “On the limits of predictability in real-world radio spectrum state dynamics,” *IEEE Commun. Mag., vol. 53, no. 7, pp. 178–183, 2015*.

[14] M. Patnaik, V. Kamakoti, V. Matyáš, and V. Řchák, "PROLEMus: A proactive learning-based MAC protocol against PUEA and SSDF attacks in energy constrained cognitive radio networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 2, pp. 400–412, 2019.
[15] M. Patnaik, G. Prabhu, C. Rebeiro, V. Matyáš, and V. Kamakoti, "ProBLeSS: A proactive blockchain based spectrum sharing protocol against SSDF attacks in cognitive radio IoBT networks," *IEEE Netw. Lett.*, vol. 2, no. 2, pp. 67–70, 2020.
[16] M. Patnaik, K. Bharati, and V. Matyáš, "ProMETHEUS: A secure lightweight spectrum allocation protocol against SSDF attacks in cognitive radio IoT networks," *IEEE Commun. Lett.*, vol. 27, no. 11, pp. 2919–2923, Nov. 2023.
[17] A. Tani and D. Marabissi, "Distributed cooperative spectrum sensing via push–sum consensus for full-duplex cognitive aerial base stations," *Future Internet*, vol. 18, no. 1, Art. no. 10, Dec. 2025.
[18] M. U. Muzaffar and R. Sharqi, "A review of spectrum sensing in modern cognitive radio networks," *Telecommun. Syst.*, vol. 85, pp. 347–363, 2024.
[19] S. K. Jagatheesaperumal, I. Ahmad, M. Höyhtyä, S. Khan, and A. Gurtov, "Deep learning frameworks for cognitive radio networks: Review and open research challenges," *J. Netw. Comput. Appl.*, vol. 233, Art. no. 104051, 2025.
[20] I. E. Ezhilarasi, A. K. Rai, R. Kumar, and A. A. A. Ari, "A survey on cognitive radio network attack mitigation Using machine/deep learning and blockchain," *EURASIP J. Wireless Commun. Netw.*, 2023.
[21] N. Y. Al-Matari, A. T. Zahary, and A. A. Al-Shargabi, "A survey on advancements in blockchain-enabled spectrum access security for 6G cognitive radio IoT networks," *Sci. Rep.*, vol. 14, Art. no. 30990, 2024.
[22] K. Xie, X. Ning, X. Wang, D. Xie, J. Cao, G. Xie, and J. Wen, "Recover corrupted data in sensor networks: A matrix completion solution," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1434–1448, 2016.
[23] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.
[24] A. Narayanan, B. Natarajan, and K. P. Subbalakshmi, "Spatio-temporal spectrum sensing in cognitive radio networks using matrix completion," in *Proc. ACM MSWiM*, 2014, pp. 73–80.
[25] Z. Chen, T. Cooklev, C. Chen, and C. Pomalaza-Ráez, "Modeling Primary User Emulation Attacks and Defenses in Cognitive Radio Networks," in *Proc. IEEE Int. Perf. Comput. Commun. Conf. (IPCCC)*, 2009, pp. 208–215.
[26] S. Shrivastava, S. Tiwari, and D. Bhargava, "A survey on security issues in cognitive radio-based cooperative networks," *IET Commun.*, vol. 15, no. 7, pp. 900–918, 2021.
[27] P. Ahuja, P. Sethi, and N. Chauhan, "A comprehensive survey of security threats, detection, countermeasures, and future directions for physical and network layers in cognitive radio networks," *Multimed. Tools Appl.*, vol. 83, no. 11, pp. 32715–32738, 2024.
[28] M. Imran *et al.*, "Anti-jamming for cognitive radio networks with Stackelberg game-assisted DSSS approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2024, Art. no. 73, 2024.
[29] F. D. Zemo and S. Bakkali, "A survey on security threats in cognitive radio networks based on cooperative spectrum sensing," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 30, no. 4, pp. 433–466, 2024.
[30] J. Wang, I.-R. Chen, J. J. P. Tsai, and D.-C. Wang, "Trust-based mechanism design for cooperative spectrum sensing in cognitive radio networks," *Comput. Commun.*, vol. 116, pp. 98–110, 2018.
[31] L. A. Almuqren, M. Maray, F. A. Alotaibi, A. Alzahrani, A. Mahmud, and R. R. Mohammed, "Optimal Deep Learning Empowered Malicious User Detection for Spectrum Sensing in Cognitive Radio Networks," *IEEE Access*, vol. 12, pp. 35300–35308, 2024.
[32] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Commun. ACM*, vol. 55, no. 6, pp. 111–119, 2012.
[33] B. Marlin, R. S. Zemel, S. Roweis, and M. Slaney, "Collaborative filtering and the missing-at-random assumption," *arXiv preprint arXiv:1206.5267*, 2012.
[34] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE CVPR*, 2012, pp. 1568–1575.
[35] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 23, 2010.
[36] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters, "1-bit matrix completion," *Inf. Inference*, vol. 3, no. 3, pp. 189–223, 2014.
[37] S. Majumder, "One-bit spectrum sensing using Gustafson–Kessel fuzzy clustering for cognitive radio network," *Phys. Commun.*, vol. 62, Art. no. 102239, 2024.
[38] J. Bennett and S. Lanning, "The Netflix Prize," in *Proc. KDD Cup Workshop*, 2007, pp. 3–6.
[39] J. Tan, W. Liu, T. Wang, M. Zhao, A. Liu, and S. Zhang, "A high-accurate content popularity prediction computational modeling for mobile edge computing using matrix completion technology," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 6, e3871, 2021.
[40] J. Cheng, Q. Ye, H. Jiang, D. Wang, and C. Wang, "STCDG: An efficient data gathering algorithm based on matrix completion for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 850–861, Feb. 2013.
[41] M. Kortas, O. Habachi, A. Bouallegue, V. Meghdadi, T. Ezzedine, and J.-P. Cances, "Robust data recovery in wireless sensor network: A learning-based matrix completion framework," *Sensors*, vol. 21, no. 3, p. 1016, 2021.

Authors' Profiles



Charanjeet Singh Chahil received the BTech degree in electronics and information technology from Jawaharlal Nehru University, New Delhi, India, in 2007, and the MTech degree in electronics and communication in 2012. He is currently pursuing the PhD degree in computer science from Chitkara University, India. He has professional experience in planning and fielding large enterprise IP/MPLS communication networks and in managing communication infrastructure in difficult and remote terrain. His research interests include cognitive radio networks, secure spectrum allocation, network resilience, and communication systems for constrained and mission-critical environments.



Milan Patnaik received the B.Tech. degree in telecommunication and IT from Jawaharlal Nehru University, New Delhi, India, the M.Tech. degree in computer science and engineering from the Indian Institute of Technology Madras, Chennai, India, and the M.Sc. degree in ICT security from Masaryk University, Brno, Czech Republic. He has completed his Doctorate from Indian Institute of Technology Madras, Chennai. His research interests include designing proactive and predictive algorithms for securing systems to be used on constrained devices. He has been instrumental in developing security products designed on constrained hardware for operational technology and industrial control systems that has been deployed in various critical information infrastructures in India. His research in securing maritime infrastructures has been accepted by global maritime management organizations for deployment in more than 1000+ ships.



Deepali Gupta is currently working as Professor-Research at Chitkara University Research and Innovation Network (CURIN) and Associate Dean (PhD Programs) at Chitkara University, Punjab, India. Dr. Gupta specializes in software engineering, machine learning, cloud computing, IoT and genetic algorithms. She has published more than 180 research papers in national and international journals and conferences. Based on these areas she has guided many PhD and ME scholars. Dr Deepali has worked at various administrative positions and is an active member of various professional bodies like IEI (India), IETE, ISTE etc. apart from being Editor-in-chief of MMU journal, she is an editorial board member and reviewer of various journals.



Vaclav (Vashek) Matyas is a Professor at Masaryk University, Brno, CZ, heading its Centre for Research on Cryptography and Security. His research interests are related to applied cryptography and security; he has published well over 200 peer reviewed papers and articles and has co-authored several books. Vashek is a member of The Research, Development and Innovation Council of the Government of the Czech Republic. He worked in the past with Cybernetica, Red Hat Czech, CyLab at Carnegie Mellon University, as a Fulbright-Masaryk Visiting Scholar at the Center for Research on Computation and Society of Harvard University, Microsoft Research Cambridge, University College Dublin, Ubilab at UBS AG, and as a Royal Society Postdoctoral Fellow with the Cambridge University Computer Lab. Vashek also worked on the Common Criteria and in ISO/IEC JTC1 SC27. He received his PhD degree from Masaryk University, Brno.

How to cite this paper

Charanjeet Singh Chahil, Milan Patnaik, Deepali Gupta, Vashek Matyas, "ProCLAMUS: A Matrix Completion Based Proactive Spectrum Allocation Protocol for Security of IoT based CRNs", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.18, No.5, pp. 163-181, 2026. DOI:10.5815/ijieeb.2026.05.10