Defending Against Byzantine Attacks in CRNs: PCA-Based Malicious User Detection and Weighted Cooperative Spectrum Sensing

[1]Ankit Chouhan , [1]Ashok Parmar, [1]Kamal Captain, [2]Miguel López-Benítez
[1]Sardar Vallabhbhai National Institute of Technology, Surat, India.
[2]a. Department of Electrical Engineering and Electronics, University of Liverpool, UK.
b. ARIES Research Centre, Antonio de Nebrija University, Madrid, Spain

***Abstract*—Cognitive radio (CR) technology is a viable solution for assisting secondary users to share the licensed radio spectrum of primary users. Cooperative spectrum sensing (CSS) enhances the accuracy of spectrum sensing in a CR network. However, the effectiveness of CSS can be compromised by malicious users (MUs) who intentionally send false sensing information to the fusion center. This letter focuses on enhancing the CSS performance and detecting the MUs. We propose a machine learning technique to identify and classify MUs in a CR network using the Principal Component Analysis algorithm. The performance of the proposed algorithm in detecting MUs and enhancing CSS performance is validated through simulation experiments.**

***Index Terms*—Cooperative Spectrum Sensing, Cognitive Radio Network, Byzantine attack, Machine Learning, Principal Component Analysis.**

I. Introduction

Wireless network technology has evolved significantly, but more spectrum resources are needed to meet future wireless communication systems' high-speed and reliable requirements. To solve this issue, a cognitive radio network (CRN) is introduced, allowing cooperative secondary users (CSUs) to access licensed bands intelligently based on the PU's current state (occupied or idle). This technique, known as spectrum sensing, is critical for CRNs to enhance spectrum utilization efficiency while eliminating interference. However, decisions on PU's actual state are sensitive to channel impairments, such as shadowing and multi-path fading.

Cooperative spectrum sensing (CSS) is a solution to this problem, which has become a significant technology for CRNs, showing high tractability and robustness. It has been recommended in the IEEE 802.22 standard and can be implemented in a centralized approach [1]. In centralized CSS, the fusion center (FC) receives local sensing observations periodically and then implements a specific combination rule to make the final judgment. In CSS, multiple CSUs share their sensing results with the FC for decision-making. The results shared with the FC can be binary or reliability metrics, such as the signal-to-noise (SNR) ratio or the energy detector. In CSS, hard combining involves CSUs making binary decisions (1 or 0) about PU signal presence in their local bands. These decisions are sent to the FC, which combines them using rules like AND, OR, or Majority Voting to determine overall PU presence or absence in the spectrum. In soft combining, CSUs share their local decisions as reliability metrics, often using SNR or energy detector values. The FC then combines these metrics from all CSUs, typically through weighted averages or maximum selection rules, to make a final decision.

Now consider the Byzantine attack, also known as spectrum sensing data falsification (SSDF). The Byzantine attack is a malicious attack in CSS involving one or more malicious users (MUs) intentionally providing false sensing results to the FC. The objective of this attack is to deceive the FC, which can result in an incorrect decision on the presence or absence of the PU's signal. Since the FC relies on the correctness and honesty of the sensing results from all CSUs, a Byzantine attack can devastate CSS. Even a few MUs providing false information can significantly affect the overall detection performance.

Authors in [2] present block outlier identification methods based on the Tietjen-Moore (TM) test. These methods are intended to identify SSDF attacks initiated by MUs during CSS. Within this context, the paper explores the difficulties in calculating the number of MUs in this scenario. Authors in [3] introduce an algorithm that involves the calculation of the variance in the received sensing signal. Suppose the computed variance surpasses a predefined attack threshold. In that case, the FC does not incorporate this potentially manipulated value into the data fusion process. The isolation forest-based anomaly detection (IFAD) algorithm presented in [4] shows potential for detecting MUs in the CRN. However, when the likelihood of MUs in the CRN approaches 40 %, its performance is affected. In [5], the authors present an alternative method known as the modified outlier removal spectrum sensing (MORSS) approach. The MORSS strategy relies on quartile-based criteria to systematically filter out outliers from the data before initiating the fusion process. Authors in [6] introduce a Gaussian mixture model based anomaly detection (GMMAD) algorithm for defense against Byzantine attack, where a weighted sum based CSS algorithm is proposed to improve the detection performance.

The primary contributions of this letter are as follows:

- We propose a principal component analysis (PCA)-based MU detection algorithm. This algorithm detects MUs even when the probability of attack is minimal, irrespective of the number of MUs is small or large in the CRN.
- Furthermore, we also propose a PCA-based weighted sum algorithm for CSS. The proposed algorithm improves CSS performance, particularly in the presence of Byzantine attacks. The results section contains plots illustrating the algorithm's effectiveness in improving channel detection probability.

II. System Model

We consider a CRN in which M CSUs cooperatively sense the PU's signal using a centralized CSS approach. The n^{th} of the received signal at the i^{th} CSU can be expressed as

$$y_i(n) = \begin{cases} w_i(n), & H_0 \\ h_i(n)s_i(n) + w_i(n), & H_1, \end{cases} \tag{1}$$

where $i = 1, 2, \ldots, M$, $h_i(n)$ represents the channel gain between PU and i^{th} CSU, and $s_i(n)$ and $w_i(n)$ are the n^{th} sample of the PU's received signal and thermal noise respectively, with $n = 1, 2, \ldots, N$. The signal and noise samples, denoted as $s_i(n) \sim \mathcal{CN}(0, \sigma_{s_i}^2)$ and $w_i(n) \sim \mathcal{CN}(0, \sigma_{w_i}^2)$, are assumed to be independent and identically distributed. Here, $\mathcal{CN}(\mu_x, \sigma_x^2)$ represents the circularly symmetric complex Gaussian distribution with mean μ_x and variance σ_x^2.

Each CSU conducts energy detection over a time period denoted as T. When the bandwidth is represented as ω the energy detector acquires ωT base-band complex signal samples within this time frame. We consider that the received power of the PU is fixed to $\upsilon = \sum_{n=1}^{\lfloor \omega T \rfloor} \mathbb{E}[|s_i(n)|^2]/T$ and noise spectral density is represented by $\gamma = \mathbb{E}[|w_i(n)|^2]$. We have adopted this system model from [7], [8]. The energy detection mechanism employed by the i^{th} CSU estimates the energy value normalized by the noise spectral density. The computed energy value E_i at the i^{th} CSU is obtained using the received signal samples as

$$E_i = \frac{2}{\gamma} \sum_{n=1}^{\lfloor \omega T \rfloor} |y_i(n)|^2. \tag{2}$$

Each CSU reports its estimated energy value to the FC at the j^{th} sensing instance, where the FC compiles the value received from all the CSUs into an energy vector, defined as

$$\mathbf{E^j} = [E_1, E_2, \ldots, E_M]^{'}, \tag{3}$$

where $j = 1, 2, \ldots, L$, $[\cdot]^{'}$ represents the transpose operator and L represents the number of sensing instances.

The FC records the reported energy vectors for all sensing instances to form a matrix E as

$$E = [\mathbf{E^1}, \mathbf{E^2}, \ldots, \mathbf{E^L}]. \tag{4}$$

Since there are M CSUs and L sensing instances, the dimension of matrix E is $M \times L$.

We assume that the PU and CSUs are stationary. The power attenuation from i^{th} CSU to the PU is computed as

$$A_i = \rho\left(\| \mathbf{c}^{PU} - \mathbf{c}_i^{CSU} \|\right) \cdot \varphi_i \cdot \vartheta_i. \tag{5}$$

Here, $\| \cdot \|$ represents the Euclidean distance, $\rho(d) = d^{-\alpha}$ stands for the path-loss factor concerning the relative distance d and the path-loss exponent α. At the same time, φ_i denotes the shadow fading, and ϑ_i denotes multi-path fading components. $\mathbf{c}^{PU}$ and $\mathbf{c}_i^{CSU}$ represent the coordinate of PU and CSU, respectively.

A. Byzantine Attack Model

The CRN is susceptible to various sensing threats owing to its software-defined radio and wireless channel integrity. Among these threats is the Byzantine attack, where the MU reports false information to the FC to manipulate global decisions. The attacker manipulates the sensing decision with a varying probability of disrupting the CSU and PU performance. It is important to note that the smart attacker does not consistently report false information to the FC. The general probabilistic attack can be formulated as

$$\begin{cases} P(O_{MU}^{'} = O_{MU} + \Delta | O_{MU} < \eta) = & \alpha_0 \\ P(O_{MU}^{'} = O_{MU} - \Delta | O_{MU} > \eta) = & \alpha_1. \end{cases} \tag{6}$$

where O_{MU} represents the observations made by the malicious user, Δ is the attack strength and η signifies the attack threshold. $O_{MU}^{'}$ denotes the manipulated sensing outcomes, with α_0 and α_1 representing the probability of a false alarm attack and a miss detection attack, respectively.

III. Proposed Algorithm

PCA is an unsupervised machine learning method for feature extraction and dimensionality reduction. PCA also serves for anomaly detection. It accomplishes this by identifying essential features, known as principal components (PCs), which enable the reduction of data dimensionality. By projecting data into a more compact representation, PCA minimizes information loss. Moreover, it is possible to efficiently reconstruct the original data from its low-dimensional representation while retaining the critical characteristics of the original data. However, this reconstruction is lossy, and the degree of loss depends on the number of PCs selected during dimensionality

```
Algorithm 1 PCA based MU Detection Algorithm
Let an energy matrix $E$ contains reported energy values from all the CSUs to form a matrix of shape $M \times L$.
  • Step 1: Compute the covariance matrix of $E$ as $E_{cov} = [E^T \times E]/L$.
  • Step 2: Compute eigenvalues and eigenvectors of $E_{cov}$ matrix by solving $E_{cov} \cdot X = \lambda \cdot X$, where $\lambda$ represents the eigenvalue and $X$ is the eigenvector.
  • Step 3: Select $K$ principal components, and form a matrix $P$ with dimensions $L \times K$.
  • Step 4: Perform dimensionality reduction of energy matrix $E$ as $\widetilde{E} = E \times P$.
  • Step 5: Reconstruct estimate $\widehat{E}$ of $E$ using $\widetilde{E}$ as $\widehat{E} = \widetilde{E} \times P^T$.
  • Step 6: Compute the reconstruct error for the $i^{th}$ CSU as $e^i = \frac{1}{L} \sum_{i=1}^{L} \left( |\widehat{E}(i,:) - E(i,:)| \right)$, where $E(i,:)$ represents the $i^{th}$ row of matrix E.
  • Step 7: Classify an $i^{\text{th}}$ CSU as a MU if error $\text{e}^i$ exceeds a specified threshold, denoted as $\lambda_{otsu}$. The value of $\lambda_{otsu}$ is determined by applying the Otsu algorithm, as outlined in algorithm 2.
```

reduction [9]. PCA-based anomaly detection employs the reconstruction error as a tool to identify anomalies. Anomalous data typically showcase higher reconstruction errors compared to non-anomalous data, thus data with elevated reconstruction errors can be identified as anomalous. This study utilizes PCA for MUs detection in CRNs, distinguishing between HCSUs and MUs based on data reconstruction errors. The detailed steps of the proposed algorithm can be found in Algorithm 1. The energy matrix E given in Eq. (4) is input to the algorithm. The algorithm first computes the covariance matrix E_{cov} of matrix E as given in step 1, and determines its eigenvalues and eigenvectors in step 2. We then select K eigenvectors corresponding to the K largest eigenvalues, referred to as the PCs, and stack them into matrix form to get matrix P in step 3. As given in step 4, using matrix P, perform dimensionality reduction on matrix E to obtain lower dimension representation $\widetilde{E}$. Now, using $\widetilde{E}$, reconstruct the estimate of original matrix E denoted as $\widehat{E}$ as given in step 5. Next, we compute reconstruction error e^i of every row of matrix E as given in step 6. Since the i$^{\text{th}}$ row of matrix E records the data received from the i$^{\text{th}}$ CSU, the error computed will represent error for i$^{\text{th}}$ CSU. Finally, in step 7, we compare reconstruction errors of all CSUs with a threshold λ_{otsu}. If the error is greater than the threshold, they are declared a MUs.

Algorithm 2 Otsu Thresholding to compute λ_{otsu}

Input: Let a vector $\mathbf{e} = [e_1, e_2, \ldots, e_M]$ contain the number of MUs in each CSU. Initially take i = 1.

- **Step 1:** Choose e_i as threshold and split the vector **e** in lower (e_l) and upper (e_u) sets as
$$e_l = \{e_j : \forall e_j < e_i\}$$
$$e_u = \{e_j : \forall e_j \geq e_i\},$$
for $j = 1, 2, \ldots, M$.
- **Step 2:** Find: $\mathcal{N}_{all}$ number of elements in **e**, $\mathcal{N}_l$ number of elements in e_l and $\mathcal{N}_u$ number of elements in e_u.
- **Step 3:** Compute weights W_l and W_u as: $W_l = \frac{\mathcal{N}_l}{\mathcal{N}_{all}}$, $W_u = \frac{\mathcal{N}_u}{\mathcal{N}_{all}}$
- **Step 4:** Compute variances σ_l^2 and σ_u^2 of e_l and e_u respectively.
- **Step 5:** Find the variance for the threshold e_i using: $\sigma_{e_i}^2 = W_l\sigma_l^2 + W_u\sigma_u^2$.
- **Step 6:** Increase i by 1 and repeat steps 1 to 5 for $i = 1, 2, \ldots, M$.
- **Step 7:** The threshold with least variance $(\sigma_{e_i}^2)$ is selected as λ_{otsu}.

We utilize Otsu's thresholding algorithm to determine the optimal threshold, denoted as λ_{otsu} [10]. Otsu's algorithm is widely employed for automatic image thresholding, segmenting images based on pixel intensity into distinct regions. It aims to find an optimal threshold value that minimizes intra-class variance of pixel intensities, maximizing separation between object and background classes. This is particularly effective for bimodal pixel intensity distributions, indicating clear distinctions between foreground and background. In Algorithm 1, our proposed approach involves a similar scenario with a bimodal distribution, where MUs' and HCSUs' reconstruction errors form distinct groups. The steps for Otsu's algorithm are outlined in Algorithm 2, applied to errors obtained for each CSU to derive the threshold. In the CSS context, Otsu's algorithm analyzes reconstructed errors, categorizing them into HCSUs and MUs by minimizing intra-class variance between these two classes.

Finally, errors obtained in Algorithm 1 are used to determine weights in the weighted sum-based CSS algorithm outlined in Algorithm 3. The algorithm assigns different weights to FC-received data based on Eq. (7) and computes the decision statistic as given in Step 2. The decision statistic D is then compared with the threshold Λ to make the final decision. By assigning lower weights to MUs due to higher reconstruction errors, the algorithm doesn't eliminate MUs but gives less importance to their data, improving overall performance.

Algorithm 3 Weighted Sum based CSS Algorithm

Input: Let a vector $\mathbf{e} = [e_1, e_2, \ldots, e_M]$ contain the reconstructed error of each CSU.

- **Step 1:** Compute weights for each CSU as
$$W_i = 1 - \left(\frac{e_i - \mu}{\sigma_e}\right), \tag{7}$$
where σ_e and μ represent the standard deviation and mean of vector e, respectively.
- **Step 2:** Using the calculated weights in step 1, derive the final decision statistic D at the FC as $D = \sum_{i=1}^{M} W_i \times E_i$.
- **Step 3:** The FC compares D with the global decision threshold Λ and if $D < \Lambda$, the PU channel is declared as free otherwise it is declared as occupied.

IV. Results Analysis

In this letter, we adopt a scenario where CSUs participating in CSS are positioned in a 5-by-5 grid topology, totaling $M = 25$ CSUs, within a $4000m \times 4000m$ area, as depicted in Fig. 1. Noteworthy simulation parameters are set as follows: the bandwidth (ω) is 5 MHz, the sensing duration (t) is 100 μs, the noise spectral density (γ) is -174 dBm/Hz, the path-loss exponent α is 4, and PC is 1. We assume that shadow fading and multi-path fading components are fixed, with $\varphi_i = 1$ and $\vartheta_i = 1$, respectively. The transmitted power of the PU is 200 mW. Furthermore, we consider single PUs with fixed coordinates at $(500m, 500m)$, the probability of a PU being active is set at 0.5 and 10000 Monte Carlo simulations are run. The proposed algorithms are executed on a 64-bit computer equipped with an Intel Core i3 processor and 12 GB RAM using MATLAB 2022a. We initially explore an application of the proposed algorithm 1 to detect MUs in the CRN. Fig. 2 illustrates various CSUs' Error (e) values. In this simulation, we consider $\alpha_0 = \alpha_1 = 0.1$ at $\Delta = 80$, and the CSU$_1$ to CSU$_5$ are intentionally introduced as MU into the CRN. Notably, the error associated with these MUs

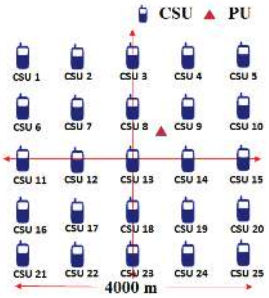

Fig. 1: The CSU's position in the CRN.

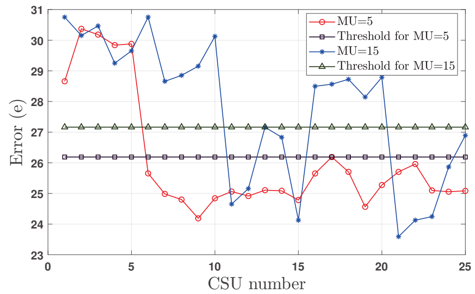

Fig. 2: Plot for malicious users detection in the CRN.

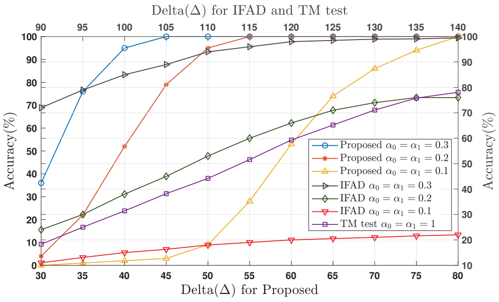

Fig. 3: Plot for accuracy by varying delta (Δ) at different attacking probabilities for $M = 25$ including $MU = 5$.

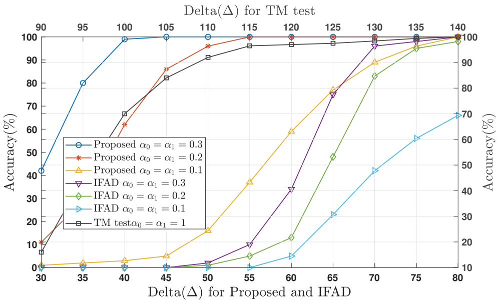

Fig. 4: Plot for accuracy by varying delta (Δ) at different attacking probabilities for $M = 25$ including $MU = 15$.

is considerably higher than other HCSUs. As we extend the number of MUs within the CRN to 15 as CSU$_1$ to CSU$_{10}$ and CSU$_{16}$ to CSU$_{20}$, the proposed algorithm 1 continues to detect them effectively. This is evident in Fig. 2, where the CSU errors exceed the threshold, signifying the presence and detection of MUs.

Fig. 3 shows the MU detection accuracy of the proposed algorithm 1 compared to existing algorithms TM Test [2] and IFAD [4] at different delta (Δ) values. The TM test algorithm makes decisions based on single sensing instances, making it reasonable to evaluate performance under the always-attack scenario $(\alpha_0 = \alpha_1 = 1)$. Additionally, the TM test relies on prior knowledge of the number of MUs obtained using a clustering-based algorithm. The IFAD algorithm requires higher delta value to detect MUs at low attack probability. The proposed algorithm do not require any prior information and can detect MUs blindly. Even with small Δ values, the proposed algorithm demonstrates high accuracy. To ensure comparability, higher Δ values have been chosen for the TM test and IFAD algorithms, and yet they lag behind the proposed algorithm in performance. Remarkably, the proposed algorithm 1 achieves 99.96% accuracy even when the probability of attack is notably low at $\alpha_0 = \alpha_1 = 0.1$ and $\Delta = 80$. In contrast, IFAD's accuracy is only 11.16% at $\Delta = 90$ with attack probabilities $\alpha_0 = \alpha_1 = 0.1$. Notably, the MU detection accuracy of the proposed algorithm improves significantly even with higher attack probabilities. For $\alpha_0 = \alpha_1 = 0.3$, the MU detection accuracy reaches 100% at $\Delta = 45$, while the IFDA algorithm accuracy is 79.43% at $\Delta = 95$. Another scenario is shown in Fig. 4; with MU=15 as CSU$_1$ to CSU$_{10}$ and CSU$_{16}$ to CSU$_{20}$ in the CRN, the performance of the proposed algorithm remains relatively stable. For example, at $\Delta = 50$, the MU detection accuracy is 98.43%, and for MU=5 with the same delta value, the accuracy is 98.13% at $\alpha_0 = \alpha_1 = 0.2$. The approximate complexities of different algorithms are $O(M \cdot L^2 + K \cdot L^2)$ for PCA, $O\left(L \cdot Q \cdot M \log(M)\right)$ for IFAD, and $O\left(M^2 + I \cdot k \cdot M\right)$ for TM test, where Q, I, and k are the depth of the tree, number of iterations, and number of clusters, respectively. Despite PCA having higher complexity, the proposed algorithm outperforms significantly. Note that, innovative techniques can be applied to further reduce the complexity of PCA algorithm.

We evaluate the MU detection accuracy of the proposed algorithm 1 with varying PU locations in the CRN, considering CSU_1 to CSU_5 as MUs. Two scenarios are considered: one with the PU at location 1 $(500, 500)$ and another at location 2 $(1500, -1500)$. When the PU is at location 1, the detection accuracy is 99.96% for $\Delta = 55$ and $\alpha_0 = \alpha_1 = 0.2$. In contrast, with the PU at location 2, the detection accuracy is 93.89% which is 6.07% lower than the case when the PU is at location 1, as shown in Fig. 5. In the first scenario, with the PU nearby, MUs receive higher energy, leading to a higher miss detection attacks. In the second scenario, where the PU is farther from MUs, they receive less energy, resulting in fewer attacks. Consequently, MUs may exhibit behaviors resembling HCSUs, making it challenging to distinguish them from HCSUs accurately. Note that, even when MUs are positioned far from the PU, they still have the potential to launch attacks, impacting the FC's global decision. Existing algorithms in TM Test [2] and IFAD [4] exhibit significantly poorer detection accuracy in these scenarios.

The impact of MUs on the CSS performance is demonstrated in Fig. 6. The performance is evaluated using the

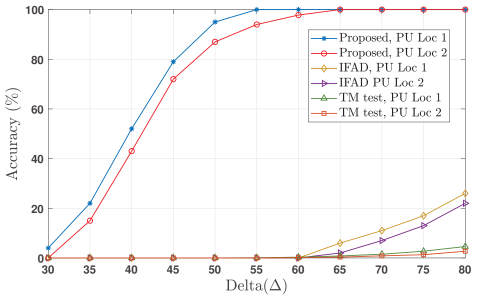

Fig. 5: Plot for accuracy by varying delta (Δ) for different location of the PU from $M = 25$ CSUs.

TABLE I: Comparison of the proposed algorithm with the existing algorithms.

P_d values at a fixed P_f of 0.1.						
All HCSU	With MUs	Without MUs	Proposed	MORSS	Algorithm in [3]	GMMAD
0.8226	0.4516	0.6788	0.7753	0.6629	0.5734	0.7124

receiver operating characteristic (ROC) curve, which is a plot of probability of false alarm, i.e., $P_f = P(D > \Lambda|H_0)$, versus the probability of detection, i.e., $P_d = P(D > \Lambda|H_1)$. The presence of MUs substantially degrades detection performance. For instance, at the false alarm probability $P_f = 0.1$, CSS with all HCSU yields the probability of detection P_d is 0.8226, while CSS with MU=15 results in $P_d = 0.4516$, representing a 45.10% reduction in detection probability due to the MU's attack. The performance degradation would be even more significant with higher attack probabilities. The detection using proposed algorithm 1 and exclusion of MUs from the decision-making process leads to improved performance. In Fig. 6, detection probability increases to 0.6788 after MUs are removed, signifying performance enhancement. To further improve CSS performance, the proposed algorithm 3 assigns weights to different CSUs based on the reconstruction error using Eq. (7), and a weighted sum is calculated at the FC to determine the decision statistic as outlined in Step 2 of algorithm 3. The P_d vs. P_f plot in Fig. 6 reveals that the weighted sum algorithm enhances CSS detection performance and mitigates the impact of MUs. Now we evaluate performance of the proposed algorithm 3 against three existing algorithms, specifically algorithm in [3], MORSS [5] and GMMAD [6]. At $P_f = 0.1$, the proposed algorithm significantly boosts the P_d by 26.04% compared to algorithm in [3], 14.49% compared to algorithm in [5], and 8.11% compared to algorithm in [6], as illustrated in Table I and Fig. 6. This improvement demonstrates the proposed algorithm's superiority. It is worth noting that the performance of algorithm in [3] is hindered by its method of eliminating MUs based on their variance exceeding the attack value. In contrast, MORSS removes MUs with data values beyond the interquartile range, resulting in relatively poor performance. In [6], a homogeneous scenario is considered where all the CSUs experience same SNRs, whereas in this paper all the CSUs receive different power based on their distance from the PU representing a heterogeneous scenario. The GMMAD algorithm performs worse because it is proposed considering homogeneous scenario. These findings underscore the effectiveness of the proposed algorithm in enhancing detection accuracy, especially when compared to existing approaches.

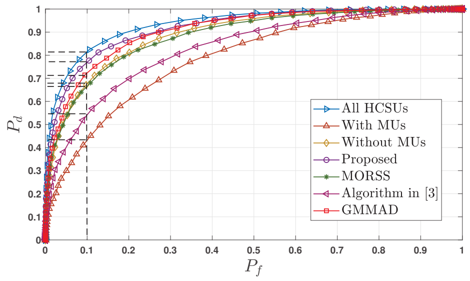

Fig. 6: Plot for P_d vs. P_f with $M = 25$, $MU = 15$, $\alpha_0 = \alpha_1 = 0.2$, $\Delta = 80$, and PU's transmission power 800 mW.

V. Conclusion

This letter proposes a PCA-based MU detection and weighted algorithm for CSS. A comparative analysis is conducted between the proposed and existing algorithms, revealing that the proposed MU detection algorithm surpasses its counterparts. The investigation delves into the impact of various parameters on the algorithm's performance, demonstrating a noteworthy enhancement in detection accuracy. Notably, the proposed algorithm effectively identifies MUs in the CRN even when the probability of attack is exceedingly low. Additionally, the proposed weighted algorithm significantly enhances CSS detection performance against Byzantine attacks.

References

[1] I. . W. Group *et al.*, "Ieee 802.22 d1: Draft standard for wireless regional area networks," 2008.

[2] S. S. Kalamkar, P. K. Singh, and A. Banerjee, "Block outlier methods for malicious user detection in cooperative spectrum sensing," in *2014 IEEE 79th Vehicular Technology Conference*. IEEE, 2014, pp. 1–5.

[3] R. Gao, Z. Zhang, M. Zhang, J. Yang, and P. Qi, "A cooperative spectrum sensing scheme in malicious cognitive radio networks," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–5.

[4] D. Mehmuda, C. Bhagat, D. Patel, K. Captain, and A. Parmar, "Defense against byzantine attack in cognitive radio using isolation forest," in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2023, pp. 314–318.

[5] L. Guo, W. Chen, Y. Cong, and X. Yan, "A robust mor-based secure fusion strategy against byzantine attack in cooperative spectrum sensing," in *International Congress on Communications, Networking, and Information Systems*. Springer, 2023, pp. 81–94.

[6] A. Parmar, K. Shah, K. Captain, M. López-Benítez, and J. Patel, "Gaussian mixture model based anomaly detection for defense against byzantine attack in cooperative spectrum sensing," *IEEE Transactions on Cognitive Communications and Networking*, 2023.

[7] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2209–2221, 2013.

[8] N. A. Khalek and W. Hamouda, "Intelligent spectrum sensing: An unsupervised learning approach based on dimensionality reduction," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 171–176.

[9] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[10] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.