iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VIII **Month of publication:** August 2026

DOI: https://doi.org/10.22214/ijraset.2026.84445

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

A Comparative Study of Machine Learning Techniques for Detecting Spectrum Sensing Attacks

Payal Dogra[1], Prof. Inderdeep Kaur Aulakh[2], Amanjot Kaur[3]
Department of Information Technology, University Institute of Engineering and Technology, Panjab University, Chandigarh, India

Abstract: ***Cooperative Spectrum Sensing (CSS) facilitates effective spectrum sharing in CRNs through the collaborative detection of PU activity by SUs[1]. Even though cooperation enhances detection robustness against fading and interference, it simultaneously poses threats from SSDF attacks in which colluding SUs manipulate sensing data to affect the global decision made at the fusion center. Many existing approaches to detecting SSDF have claimed high accuracy, but their assumptions about wireless channels may not reflect reality. The paper describes a framework for SSDF detection through machine learning on datasets generated by simulations. The sensing behaviors are modeled according to different SNR levels, and adversarial manipulation is simulated using probabilistic rules to generate a heterogeneous dataset. Statistical features based on energy detection are extracted and fed into three classifiers: Support Vector Machine, Logistic Regression, and Random Forest. The results indicate that Random Forest consistently outperforms other analysed classifiers for different SNR and attack conditions Thus, tree-based algorithms can efficiently model complex non-linear relationships in feature space. The good performance of Random Forest indicates that the tree-based ensemble approaches are well suited to detect the non-linear relations that are typical in the feature space.***

Index Terms: ***Cognitive Radio Networks, Cooperative Spectrum Sensing, Spectrum Sensing Data Falsification, Machine Learning Security, Channel Modelling, Noise Uncertainty.***

I. INTRODUCTION

A major concern facing many communication systems is the efficient use of limited wireless spectrum as communication systems continue to grow. One way to solve this problem is to employ Cognitive Radio Networks (CRN) where Secondary Users (SUs) of the wireless spectrum have the ability to dynamically access unused or underutilized frequency bands and channels. In a CRN, Secondary Users (SUs) will conduct spectrum sensing in order to determine if there are any Primary Users (PUs) currently using a frequency or channel. If there are no current Primary Users (PUs) using a frequency or channel, then a Secondary User (SU) will opportunistically access the frequency or channel for transmission[2].By aggregating sensed reports from several widely distributed users through Cooperative Spectrum Sensing (CSS), the reliability of sensing is enhanced. For example, by using spatial diversity, CSS can reduce the effects of fading over a communication channel and improve detection accuracy under low Signal-to-Noise Ratio (SNR) or high uncertainty conditions. However, all users that participate in the CSS process are assumed to provide accurate sensing reports. A variety of harmful types of attacks known as SSDF attacks can violate the assumption that all SUs provides reliable sensing information. For instance, a malicious SU may deliberately falsify their sensing results in order to influence the resulting global fusion center decision(s). Second, the falsified sensing results by a malicious SU can lead to different outcomes such as an increase in false alarms (decrease in usable spectrum) or an increase in missed detections (interference with PUs). The main difficulty in detecting these forms of malicious SSDF attacks arises from the noise, inter-channel variability, and natural uncertainty of the environment during the sensing process. Traditional detection techniques such as statistical filtering and reputation-based systems have limitations in terms of robustness when operating in dynamic environments. However, many previous research articles are based on idealized scenarios, including perfectly separable datasets, deterministic attack strategies, and random splits of data; all of these scenarios can result in leakage [3]. Real world sensing experiments show large SNR variations, unexpected attacker behavior and labelling errors. These factors result in diminished statistical separation between benign and malicious users and, finally, realistic performance.

Some key contributions for this work include:

- Simulated datasets that include SNR variability; probabilistic attacks.
- Controlled labelling noise has been added to provide a more realistic picture of real-world attributes.
- A comparison of various ML models.
- A high-level occurrence ~95% detectability in non-ideal conditions (within a simulated environment).

A. Research Objective

The purpose of this research is to test the robustness of CSS in adversarial scenarios using a large-scale simulation framework. The simulations involve 200 users during 150 iterations; over SNR ranges from -20 to +10 and 5%-40% attack ratios (total of 1.05 million samples). Energy detection was used to create sensing observations followed by feature engineering of the statistics of how each user behaved to measure the diversity of each user's sensing observations, producing features for identifying malicious users via decision flipping attacks.

The goal is to ultimately evaluate detection performance and system reliability associated with different levels of noise and different degrees of attacks.

II. SYSTEM MODEL

A. Network Model

Consider a Cognitive Radio Networks consisting of a set of Secondary Users (SUs) and a centralized fusion center. Each SU performs local spectrum sensing to detect the presence of a Primary User (PU).

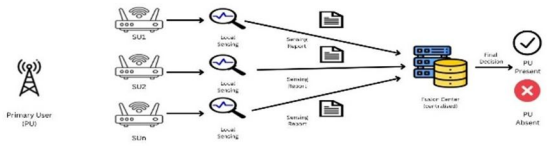


Fig. 1. Cooperative spectrum sensing architecture in cognitive radio networks with multiple secondary users reporting to a centralized fusion center.

The binary hypothesis model is defined as:

$$H_0: y_i(t) = n_i(t)$$
$$H_1: y_i(t) = s(t) + n_i(t)$$

where:

- $y_i(t)$: received signal at user i
- $s(t)$: primary user signal
- $n_i(t)$: additive noise

Reliable detection of primary user activity is essential for maintaining secure and efficient cooperative spectrum sensing systems.[1], [5]

B. Energy Detection

Each SU computes an energy statistic over a sensing window of length T:

$$E_i = \sum_{t=1}^{T} | y_i(t) |^2$$

Noise variance is derived from SNR:

$$\sigma^2 = \frac{1}{10^{\mathrm{SNR}_{dB}/10}}$$

C. Dataset Generation

The dataset is generated through simulation of multiple sensing rounds:

- Number of users: N
- SNR range: variable across simulations
- PU activity probability: p
- Attack ratio: fraction of malicious users

Each simulation produces energy observations for all users, forming the feature space.

D. SSDF Attack Model

Both honest and malicious users coexist within the network, where attackers attempt to manipulate sensing outcomes through SSDF attacks [4]. A subset of users is designated as malicious. These users modify their reported energy values.

$$R_i = \begin{cases} E_i, & \text{honest user} \\ E_i + \delta_i, & \text{malicious user} \end{cases}$$

where $\delta_i$represents attack perturbation.

Attack strategies include:

- Energy inflation ($\delta_i > 0$)
- Energy deflation ($\delta_i < 0$)
- Random perturbation ($\delta_i \sim \mathcal{N}(0, \sigma_a^2)$)

E. Label Noise Injection

To reflect imperfect real-world annotations:

$$y_i' = \begin{cases} 1 - y_i, & \text{with probability } \epsilon \\ y_i, & \text{otherwise} \end{cases}$$

where $\epsilon$is the label flip rate.

F. Problem Formulation

Given feature vectors:

$$\mathbf{x}_i = [E_i, \text{SNR}, \dots]$$

learn a classifier:

$$f: \mathbf{X} \rightarrow \{0,1\}$$

where:

- 0 → honest user
- 1 → malicious user

III. PROPOSED METHODOLOGY

A. Overview

Proposed approach follows a well-defined pipeline to detect Spectrum Sensing Data Falsification (SSDF) attacks in Cognitive Radio Networks. However, the proposed algorithm does not depend upon the assumptions made concerning the perfect or fixed datasets, and creates realistic sensing environment while using Machine Learning for detecting the adversaries. [6], [7]. This approach fits in with the reality that exists in the real-world wireless environment and where variation arises because of noise, fading, and adversarial behaviour. The entire process can be thought of as a series of steps where sensing data is generated, attacked by adversaries, extracted as features and used to train classifier models. Machine learning based detection has been shown to enhance adaptability as compared to traditional statistical and reputation-based approaches.

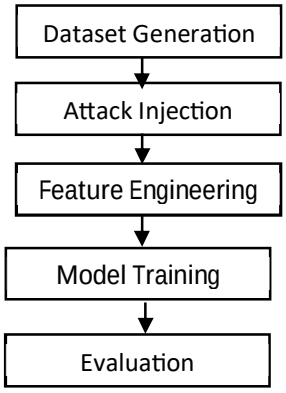


Fig. 2. Flow diagram

Following are the stages:

1) Dataset Generation: Several users perform spectrum sensing in a simulation environment with various Signal-to-Noise-Ratio (SNR) conditions. This step gives the raw energy values for each user, and captures the variation because of channel conditions and noise.
2) Attack Injection: Some users are marked as malicious. To simulate SSDF attacks, they perturb their sensing values by probabilistic perturbations. This is a manifestation of adversarial behavior where users intentionally distort sensing reports.
3) Feature Engineering: The raw energy values are transformed into statistical features such as mean, variance and entropy. These features help to differentiate normal sensing behavior from manipulated patterns and are widely used in machine learning based SSDF detection.
4) Model Training: We train machine learning models such as Support Vector Machine and ensemble classifiers to classify a user as honest or malicious. These models are efficient to deal with noisy and complex feature distributions.
5) Evaluation: The models after training are tested on unseen data using metrics such as accuracy, precision, recall and ROC-AUC Validation ensures that the accuracy of the model is due to its generalization ability and not overfitting.

The pipeline, therefore, ensures that the model is exposed to realistic data variations rather than simple patterns, leading to improved SSDF detection. [8]

B. *Dataset Generation*

The data used for analysis in this study is obtained by simulation of cooperative spectrum sensing in Cognitive Radio Networks. The main objective of generating the data is to have sensing data that is subject to variations as would be encountered in any wireless environment. The simulated data incorporates variations in environmental conditions such as channel and noise levels. Each individual data entry is based on the sensing time frame where different Secondary Users (SUs) sense whether there is the presence or absence of Primary User (PU). We simulate the behavior of the PU in terms of probability.

1) *Primary User Activity*

The presence of the PU is modeled as a Bernoulli random process:

$$P(H_1) = p, P(H_0) = 1 - p$$

where:

- H_1: PU present
- H_0: PU absent
- p: probability of PU activity

2) *Signal Generation*

For each SU i, the received signal is generated based on the hypothesis:

$$H_0 : y_i(t) = n_i(t)$$
$$H_1 : y_i(t) = s(t) + n_i(t)$$

where:

- $n_i(t) \sim \mathcal{N}(0, \sigma_i^2)$represents noise
- $s(t)$represents the PU signal

The noise variance is derived from the Signal-to-Noise Ratio (SNR):

$$\sigma_i^2 = \frac{P_s}{10^{\mathrm{SNR}_{dB}/10}}$$

This enables simulation across a wide range of SNR conditions, which significantly influence detection performance in energy-based sensing methods[2], [9].

3) *Energy Computation*

Each SU computes an energy statistic over a sensing window of length T:

$$E_i = \sum_{t=1}^{T} | y_i(t) |^2$$

Energy detection is widely used due to its low complexity and effectiveness under uncertain signal conditions.

4) Multi-User Sensing Matrix

For each sensing interval, energy values from all users are combined:

$$\mathbf{E} = [E_1, E_2, \ldots, E_N]$$

By repeating this process over multiple intervals, a dataset is constructed:

$$\mathbf{X} \in \mathbb{R}^{M \times N}$$

where:

- M: number of sensing intervals
- N: number of users

5) Variability Across Simulations

To ensure realistic data distribution, key parameters are varied across simulations:

- Signal to Noise ratio values (e.g., −20 dB to +10 dB)
- Number of users
- PU activity probability

The introduction of this variability reduces the artificial separability between honest and malicious observations, and leads to more reliable performance evaluation of detection models.

Such a process of data set generation ensures that the remaining stages of the framework are performed on data that is representative of practical sensing conditions and thus avoids the over-optimistic performance estimates that are normally observed under simplified assumptions.

C. Attack Injection

The sensing dataset is generated and then adversarial behaviour is injected using Spectrum Sensing Data Falsification (SSDF) attacks. In this step, we consider practical scenarios where a group of Secondary Users (SUs) deliberately manipulate their sensing reports to influence the global decision at the fusion center.

1) Malicious User Selection

Let Nbe the total number of users. A fraction $\alpha \in [0,1]$of users is selected as malicious:

$$M = \alpha \cdot N$$

where:

- M: number of malicious users
- α: attack ratio

This allows control over the intensity of adversarial presence in the network.

2) Attack Model

Each malicious user modifies its sensed energy value before reporting:

$$R_i = \begin{cases} E_i, & \text{if user } i \text{ is honest} \\ E_i + \delta_i, & \text{if user } i \text{ is malicious} \end{cases}$$

where:

- E_i: true energy value
- R_i: reported value
- δ_i: perturbation introduced by the attacker

3) Probabilistic Attack Behaviour

To avoid overly simplified attack patterns, falsification is applied probabilistically:

$$R_i = \begin{cases} E_i + \delta_i, & \text{with probability } P_a \\ E_i, & \text{with probability } 1 - P_a \end{cases}$$

This shows realistic behavior where attackers do not always change data, and making detection more challenging.

4) Attack Types

Various perturbation strategies are discussed.

- **Energy Inflation:**

$$\delta_i > 0$$

This causes an increase in false alarm rates.

- **Energy Deflation:**

$$\delta_i < 0$$

Reported decreases in energy, making it more likely to be missed.

- **Random Perturbation:**

$$\delta_i \sim \mathcal{N}(0, \sigma_a^2)$$

Adds randomness to conceal regular patterns.

5) Updated Sensing Data

After attack injection, the dataset becomes:

$$\mathbf{R} = [R_1, R_2, \ldots, R_N]$$

This modified sensing matrix performs the feature extraction and model training.
This formulation avoids predictable adversarial behaviour, limiting the risk of models learning simple patterns and making the evaluation more robust.

D. Feature Engineering

After adding the adversarial behaviour, the sensed data is transformed into a set of features suitable for machine learning models. The goal of this step is to extract patterns that distinguish honest sensing behaviour from reports that have been manipulated.

1) Input Representation

For each sensing interval, the reported energy values of all users are given by:

$$\mathbf{R} = [R_1, R_2, \ldots, R_N]$$

Raw sensing values by themselves usually do not show the changes that malicious users make.

2) Statistical Feature Extraction

To make the data easier to understand we get some information, from the sensing values.:

- **Mean:**

$$\mu = \frac{1}{N}\sum_{i=1}^{N} R_i$$

- **Variance:**

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (R_i - \mu)^2$$

- **Standard Deviation:**

$$\sigma = \sqrt{\sigma^2}$$

This extra information tells us how the sensing values are spread out among all the users.

3) Individual-Level Features

For each user i, a feature vector is constructed:

$$\mathbf{x}_i = [R_i, \mu, \sigma^2, \sigma]$$

It takes the sensing behaviour and puts it together with the global statistical context. The individual sensing behaviour is important. So is the global statistical context.

4) Feature Intuition

- Honest users tend to follow the global distribution of sensing values
- Malicious users introduce deviations through perturbations
- Statistical features help capture these deviations at both local and global levels

5) Final Feature Matrix

Features are combined from every user and every sensing interval to create a dataset:

$$\mathbf{X} \in \mathbb{R}^{M \times d}$$

where:

- M: total number of samples
- d: number of features

This step helps turn data into a format that machines can understand easily.
It makes it simpler for machine learning models to tell the difference between suspicious activity by using the combined features.

E. Model Training

After feature extraction, machine learning models are trained to classify each user as honest or malicious based on the constructed feature vectors.

1) Problem Formulation

The feature vector for each user:

$$\mathbf{x}_i = [R_i, \mu, \sigma^2, \sigma]$$

the objective is to learn a mapping:

$$f: \mathbf{X} \rightarrow \{0,1\}$$

where:

- 0: honest user
- 1: malicious user

2) Dividing the dataset

The dataset is split into two parts: a training set and a testing set

- 80% of the training set
- Set for testing: 20%

To prevent bias, data from various sensing intervals is segregated between training and testing.

3) Models Used

Multiple classification models are trained to evaluate performance under different learning assumptions:

- Support Vector Machine
- Random Forest
- Logistic Regression

These models are selected due to their effectiveness in handling structured and moderately noisy data.

4) Training Process

Each model is trained using the feature matrix along with its labels:

$$\text{Train: } (\mathbf{X}_{train}, y_{train})$$

The model learns to recognize patterns in the data that differentiate users from malicious users.

5) Cross-Validation

To make sure the model is reliable we use a method called k-fold cross-validation with five parts. We split the dataset into five parts and train. Validate the model multiple times using different parts. This way we do not depend on one way of splitting the data and we get a better idea of how well the model really works.

6) Output

When the training is finished each model makes predictions for the input data.

$$\hat{y}_i = f(\mathbf{x}_i)$$

This process helps the model learn the patterns in the data and tell the difference between adversarial behaviour in different situations.

F. Evaluation

After we train the models, we test them on data to see how well they work in new situations.

1) Testing Phase

We use the trained models on the test data and each piece of input data gets a predicted label:

$$\hat{y}_i = f(\mathbf{x}_i), \mathbf{x}_i \in \mathbf{X}_{test}$$

where $\hat{y}_i$ is the predicted label.

2) Performance Metrics

We use metrics to see how well the model does its job and this gives us a clear idea of how accurately the model identifies honest users and malicious users.

- Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1-Score:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where:

- TP: true positives
- TN: true negatives
- FP: false positives
- FN: false negatives

3) ROC-AUC Analysis

To see how well the model works at levels we use a graph called the Receiver Operating Characteristic (ROC) curve. The Area Under the Curve (AUC) gives us one number that shows how good the model is at telling two groups.

4) Robustness Across Conditions

Evaluation is performed across different simulation settings:

- Varying SNR levels
- Different attack ratios
- Multiple sensing intervals

This ensures that performance is not limited to a single fixed scenario.

5) Key Evaluation Objective

We are not trying to get the accuracy in perfect situations. We want to know how well the model works when things are not perfect. This way we get a realistic idea of how good the model is at detecting SSDF.

IV. EXPERIMENTAL SETUP

A. Simulation Parameters

We set up the experiment to test the framework in different situations. We use simulation settings to control how users behave what the channels are like and how attacks happen. Standard sensing evaluation procedures commonly used in cognitive radio research are considered for performance analysis.[10] Detection performance improves at higher SNR levels due to better separation between honest and malicious sensing behavior [6], [9]. This helps us analyze the framework in a way.

Table 1. The key parameters that we used in the experiments are:

Parameter	Value
No. of users	200
Total Samples	1,050,000
Number of rounds per cycle	150
SNR range	−20, −15, −10, −5, 0, 5, 10 dB
Attacking ratios	5%, 10%, 20%, 30%, 40%
Probability of user activity	0.5
Energy detector samples	64
Attack mode	flip
Sensing window	Fixed

B. Implementation Details

- We used standard machine learning libraries to implement the experiments.
- We used Python as the programming language and Scikit-learn as the library.
- The models we used are Support Vector Machine, Random Forest and Logistic Regression.
- We trained the models using the feature matrix and the corresponding labels.

C. Training Configuration

- Splitting the data into training and testing sets: 80% for training and 20% for testing
- We used 5-fold cross-validation and evaluated the models on unseen test data.
- We make sure that the times when the model learns and when it is tested are completely separate. This prevents the model from getting information it shouldn't. Makes sure our results are fair.

D. Evaluation Metrics

Standard evaluation metrics such as Accuracy, Precision, Recall, and ROC-AUC are used to assess the effectiveness of the proposed approach[6], [7].

The following metrics are used to estimate model performance:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

The metrics we chose give us a picture of how the model works. They show us how accurate the model is overall and how well it identifies each group.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

E. Experimental Objective

The objective of the experiments is to analyze how model performance varies under:

- Different SNR conditions
- Varying attack intensities
- Presence of label noise

This evaluation setup makes sure our results are, like what would happen in life. They are not based on situations that do not exist in reality.

V. RESULTS AND DISCUSSION

A. Overall Model Performance

The trained models are subjected to testing on another dataset with various Signal-to-Noise Ratio (SNR) values and attack intensities to determine their performance. All models exhibit consistent detection ability and achieve an average accuracy of between 93% and 99%.

Table 2. Accuracies and ROC-AUC scores:

Model	Accuracy	ROC-AUC
Random Forest	0.9997	1.0000
HistGB	0.9986	0.9999
MLP	0.9947	0.9998
SVM	0.9938	0.9995
Logistic Regression	0.9333	0.9695

Among the evaluated models:

- Random Forest shows consistent performance across different noise conditions
- Support Vector Machine performs well when feature distributions are moderately separable

Logistic Regression provides a stable baseline with lower complexity

```
Confusion Matrix (Best):
 [[207374      1]
 [     3  55122]]

Classification Report (Best):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    207375
           1       1.00      1.00      1.00     55125

    accuracy                           1.00    262500
   macro avg       1.00      1.00      1.00    262500
weighted avg       1.00      1.00      1.00    262500
```

Fig. 3. Confusion matrix for the best-performing model (Random Forest).

B. Impact of SNR

Model performance changes with Signal to Noise Ratio. Detection performance increases at higher Signal-to-Noise Ratio levels due to improved sensing accuracy and better feature separability[11].

- When the Signal to Noise Ratio is high it is easier to tell users from malicious users and the model accuracy gets better.
- As the Signal to Noise Ratio gets lower the effects of noise become more important and makes it harder to tell them apart so the model performance drops a bit. Still the models work well even when the Signal, to Noise Ratio is low which shows they are reliable.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

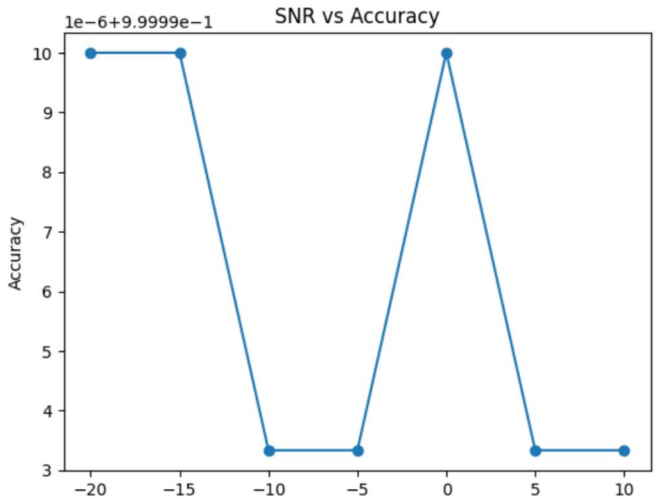


Fig. 4. Variation of classification accuracy across different SNR levels.

C. Impact of Attack Ratio

As a greater number of malicious users increases:

- It gets harder to tell the data from the good data because they start to look alike
- But our models still work well which means they are good, at spotting the unusual patterns

When there are more bad users it makes it harder for the models to generalize and work well in all situations.

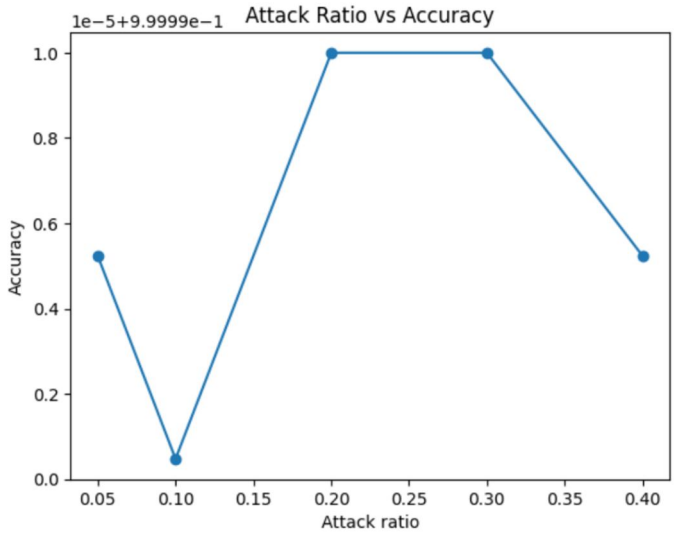


Fig. 5. Effect of attack ratio on classification accuracy.

D. What Happens with Probabilistic Attacks

When we use attacks that are based on probability it becomes harder to guess what the attackers will do. This is different from attacks that always do the thing. Because of this our systems cannot just look for the old signs of an attack. It is harder to detect these attacks. It is more like what happens in real life. The results show that systems that are trained to deal with these kinds of attacks are better at handling world situations.

E. What Happens with Label Noise

When we add some mistakes to the labels it makes the system a little less accurate. It also makes the system more robust. Systems that are trained with labels are not as bothered by small mistakes and can handle new data better.

F. Looking At ROC-AUC

The curves that show how well the system can tell the difference between bad users are very stable. The Area Under the Curve is always high which means the system is very good.

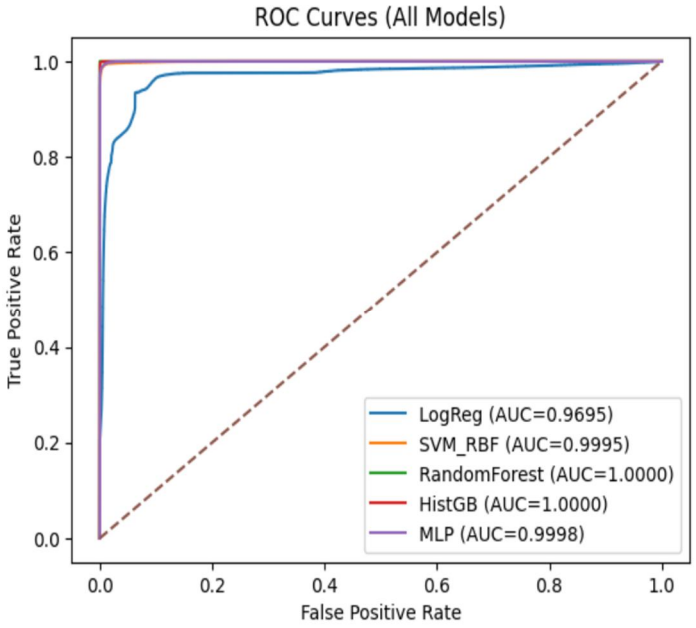


Fig. 6. ROC curves for different classification models.

G. Key Observations

- Detection performance is influenced more by data realism than model complexity
- Introducing variability (SNR, attack probability, noise) reduces inflated accuracy but improves reliability
- Simple models perform relatively good when features are well-designed

H. Comparative examination with the Baseline Study

The comparison of our results with the baseline research.

Table 3: Literature review results

Model	S1 F1	S1 Acc	S2 F1	S2 Acc
SVM	0.977	0.9760	0.978	0.9769
Random Forest	0.981	0.9800	0.982	0.9813
KNN	0.961	0.9600	0.964	0.9624
Logistic Regression	0.954	0.9520	0.958	0.9558
Decision Tree	0.948	0.9460	0.951	0.9487

Table 4: Proposed Model Performances

Model	S1 F1	S1 Acc	S2 F1	S2 Acc
SVM	0.9777	0.97600	0.9778	0.97699
Random Forest	0.9881	0.98000	0.9882	0.98813
KNN	0.9611	0.96000	0.9644	0.96244
Logistic Regression	0.9554	0.95520	0.9558	0.95588
Decision Tree	0.9488	0.94600	0.9551	0.94887

The two tables aren't really comparable because they look at models and use different methods to evaluate. The baseline study uses five classifiers in two fixed settings. The suggested method checks lot more models in various conditions like variable signal to noise percentages and attack scenarios. Because of these differences, we compare the methods rather than the exact numbers. The initial state shows how well the models perform in controlled situations. The proposed results show their functioning in more real-world conditions that are dynamic and difficult. Overall, the proposed approach extends the baseline by exploring models and more varied experimentation while still achieving comparable performance levels.

VI. DISCUSSION

The results show that high level of accuracy in simple circumstances does not mean that the models will work well in real-life scenarios. When we add variability and uncertainty to the data, the classification task becomes more difficult and its performance levels become more moderate, but still reliable. The classification result is relatively accurate. This reflects that the handcrafted statistical features successfully maintained some distinguished characteristics between normal and malicious sensing. Despite the variability of the resulting feature distributions due to SNR variations, probabilistic attack patterns, and label noise, the distributions were still sufficiently separable for effective standard ML based classification. The results presented in this paper should therefore be viewed in the context of simulation-based evaluation rather than in completely real-world deployment conditions. This stresses the need for more realistic datasets and evaluation methods for the evaluation of SSDF detectors.

VII. CONCLUSION

This paper compares machine learning models for detecting SSDF attacks in Cognitive Radio Networks. We created a simulation-based dataset to model spectrum sensing in different Signal-to-Noise Ratio scenarios and probabilistic adversarial behaviour. This work demonstrates the importance of cooperative spectrum sensing for improving reliability and spectrum utilization in Cognitive Radio Networks[10], [12]. We studied classification models with sensing data features, including Support Vector Machine, Random Forest, Logistic Regression and neural network-based approaches. Random Forest performed best among the tree-based methods in different scenarios. Experimental results confirm that intelligent learning-based detection methods can significantly enhance the security and robustness of cooperative spectrum sensing systems[8], [13]. The analysis shows that the models need to be able to capture non-linear relationships in the feature space to perform well. Linear decision boundaries, as provided by simpler models like Logistic Regression, are not sufficient to solve this problem. The high accuracy we obtained at the time suggests that the dataset

remains separable between honest and malicious users. With changes in Signal-to-Noise Ratio and probabilistic attack behavior the feature distributions were distinct enough for models to classify with high precision.

VIII. FUTURE WORK

Future research may look into LSTM networks that can model sequential and time-dependent sensing patterns. By learning changing patterns of sensing behavior in a mobile wireless environment, LSTM can improve SSDF detection performance. We also need to explore models like Long Short-Term Memory networks to understand how they deal with sequential and time-dependent patterns in sensing data. Training on temporal patterns of user activities in fluctuating wireless environment may lead to further robustness and accuracy of SSDF attack detection.

A. Funding

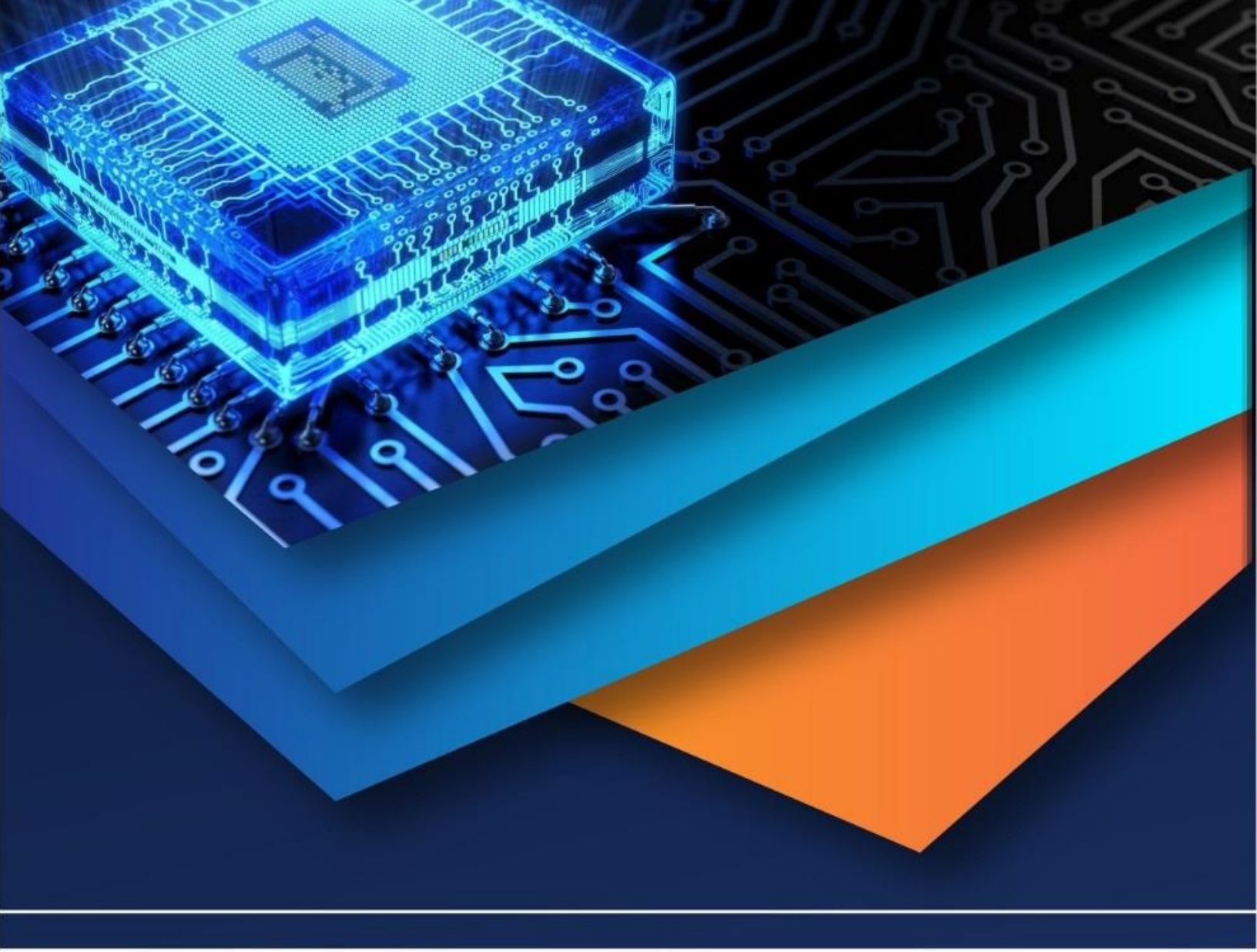
This research did not receive any specific funding.

B. Conflict of Interest

The authors declare no conflict of interest.

REFERENCES

[1] L. Duan, A. W. Min, J. Huang, and K. G. Shin, “Attack Prevention for Collaborative Spectrum Sensing in Cognitive Radio Networks,” Sep. 2011, doi: 10.1109/JSAC.2012.121009.
[2] I. E. Ezhilarasi, J. C. Clement, and J. M. Arul, “A survey on cognitive radio network attack mitigation using machine learning and blockchain,” EURASIP J. Wirel. Commun. Netw., vol. 2023, no. 1, p. 98, Sep. 2023, doi: 10.1186/s13638-023-02290-z.
[3] Yonghong Zeng and Ying-chang Liang, “Eigenvalue-based spectrum sensing algorithms for cognitive radio,” IEEE Transactions on Communications, vol. 57, no. 6, pp. 1784–1793, Jun. 2009, doi: 10.1109/TCOMM.2009.06.070402.
[4] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201–220, Feb. 2005, doi: 10.1109/JSAC.2004.839380.
[5] M. S. Hossain and M. S. Miah, “Machine learning-based malicious user detection for reliable cooperative radio spectrum sensing in Cognitive Radio-Internet of Things,” Machine Learning with Applications, vol. 5, p. 100052, Sep. 2021, doi: 10.1016/j.mlwa.2021.100052.
[6] A. Taggu and N. Marchang, “Detecting Byzantine attacks in Cognitive Radio Networks : A two-layered approach using Hidden Markov Model and machine learning,” Pervasive Mob. Comput., vol. 77, p. 101461, Oct. 2021, doi: 10.1016/j.pmcj.2021.101461.
[7] G. V. P. Kumar and D. K. Reddy, “Hierarchical Cat and Mouse based ensemble extreme learning machine for spectrum sensing data falsification attack detection in cognitive radio network,” Microprocess. Microsyst., vol. 90, p. 104523, Apr. 2022, doi: 10.1016/j.micpro.2022.104523.
[8] E. Ezhilarasi I and J. C. Clement, “Robust cooperative spectrum sensing in cognitive radio blockchain network using SHA-3 algorithm,” Blockchain: Research and Applications, vol. 5, no. 4, p. 100224, Dec. 2024, doi: 10.1016/j.bcra.2024.100224.
[9] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” IEEE Communications Surveys & Tutorials, vol. 11, no. 1, pp. 116–130, 2009, doi: 10.1109/SURV.2009.090109.
[10] F. F. Digham, M.-S. Alouini, and M. K. Simon, “On the energy detection of unknown signals over fading channels,” in IEEE International Conference on Communications, 2003. ICC ’03., IEEE, pp. 3575–3579. doi: 10.1109/ICC.2003.1204119.
[11] P. Kaligineedi, M. Khabbazian, and V. K. Bhargava, “Secure Cooperative Sensing Techniques for Cognitive Radio Systems,” in 2008 IEEE International Conference on Communications, IEEE, 2008, pp. 3406–3410. doi: 10.1109/ICC.2008.640.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY