e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:03/Issue:12/December-2021 Impact Factor- 6.752 www.irjmets.com

# TACKLING SSDF BY IDENTIFICATION OF MALICIOUS USERS BY APPLYING MACHINE LEARNING TECHNIQUES IN COGNITIVE RADIO NETWORKS

**Shreyash Kalal*1, Dr. Vijayalakshmi M.N*2**

*1Data Analyst, Pre-Sales, Aurigo Software Technologies, Bangalore, Karnataka, India.

*2Associate Professor, Master Of Computer Applications, RV College Of Engineering, Bangalore, Karnataka, India.

## ABSTRACT

Spectrum Sensing data falsification (SSDF) attack is one of the significant threats in cognitive radio networks. In this paper, a neural network (NN) machine learning (ML) is trained with multifactor trust-based computed from the spectrum sensing output of the SU's to identify the Malicious Users (MU's) in the SSDF attack. The performance of NN, Support Vector Machine (SVM), Naïve Bayes (NB), and Logistic Regression (LR) are compared. Also, the performance of all the mentioned classifiers is validated with the 'k'- fold cross-validation method. Simulations have proved that the logistic regression ML is the best model to evaluate the trust-based data set with 100% accuracy.

**Keywords:** SSDF, NN, SVM, NB, LR, Trust-Based Data.

## I. INTRODUCTION

Spectrum scarcity is the primary threat to the proliferation of wireless devices, having an exponential increase in their usage day by day. Cognitive radio networks form a solution to spectrum scarcity. It intelligently manages the spectrum usage by changing its transmission parameters according to the available spectrum and enables the communication between users. Cognitive Radio (CR) provides a new range to the secondary users- unlicensed users from the primary user –licensed users without causing interference to the latter when idle.

Cooperative spectrum sensing (CSS) exploits the spatial diversity for sensing to improve the detection performance in multipath fading, shadowing, and receiver uncertainty issues [1]. Sensing techniques, hypothesis testing, data fusion, channel assignment to a user, etc., form CSS's significant elements. The two major fusion rules are soft decision and hard decision fusion. In the former, the fusion center (FC) makes a global decision by fusing the sensing results of each SU using maximal ratio combining (MRC) or equal gain combining (EGC)[2]. In the latter, binary decisions of each SU are incorporated in the FC using fusion rules such as OR, AND, n out k, etc.

Apart from the advantages of CSS, it is more liable to threats than distributed sensing. Spectrum data falsification attack (SSDF) forms the primary threat in CSS as falsification of sensing report would hash the entire system. Literature provides many mitigation techniques for SSDF attack with trust and machine learning techniques.

## II. RELATED WORKS

In [3], Feng.et.al. Have proposed a trust fluctuation clustering mechanism to correct the dynamic collusive SSDF attackers. Reputation-based CSS is proposed by Xinyu et al. [4]. The area under one FC is divided into different cells, and users' reputation is updated after each sensing time to identify the malicious user (MU)from legitimate users. W.Wang et al. have given a heuristic approach to iteratively identify MU's by calculating the suspicious level of users [5].

Recently machine learning (ML) based techniques have been employed to detect misbehaving nodes more accurately. Both supervised and unsupervised methodologies have been used to detect malicious users. Using training, the ML algorithm learns the features in the input data and classifies the testing data depending on the known elements. The authors in [65] have proposed a mitigation technique using k-means clustering to identify the MU's by using the historical sensing data of the SU's. However, sensitivity in initializations of centroids would lead to different results. F.Farmani et al. have employed a support vector data description (SVDD)classifier to exclude the MU's from the trusted nodes in the decision phase of the FC. This method only mitigates yes or always no attack as SVDD is one class classifier[6].In [66], the Weighted Baseyain model has been proposed where the FC updates the weights of the SU's depending on their respective sensing results concerning the global decision. This method proves its proficiency unless the international decision does not go wrong. An unsupervised ML technique using Artificial Neural Network (ANN) has been proposed [68]. The method uses average suspicious value(ASD) to segregate the trusted SU's from MU's. The probability of miss-detection is high when noise is high forms the disadvantage. In [7] H.Zhu.et al. proposed a support vector machine learning scheme to mitigate different types of SSDF attacks. A classification accuracy index describing the behavior of SU is generated based on energy measurement values from multiple rounds of measurement. The simulations have proved the best results even in the case of low SNR and more MU's. This paper proposes a mitigation method to identify the MU's using a trust-based learn supervised neural network machine learning algorithm.

- Training efficiency of the ML relies on the input dataset. Thus, the ML is trained with the multifactor trusts computed for different SU's using their sensing results measured over time. These trusts act as features for the training of the ML.
- A Learn supervised Neural network ML model is designed with analysis on the optimal choice of hidden layers, activation function, and epochs.
- Comparative analysis of NN, SVM, LR, and NB concerning the precision, recall, f1 score, and accuracy is performed to prove the best ML for identifying MU's.

## III. SYSTEM MODEL

The system model under consideration consists of one primary user(PU) whose spectrum is to be sensed for idle(free), 'S' secondary users who sense the PU's spectrum, one fusion center (FC), which finalizes the decision results and allocates the idle spectrum to SU. Let 'U' be the number of malicious users within the 'S' number of SU's. The MU's to be detected falls under three conditions:

(i) always 'yes' – the SU always reports that the spectrum is idle when it is not so.
(ii) always 'No'- the SU always reports that the spectrum is busy even when not.
(iii) Mischievous – reports randomly irrespective of the actual report.

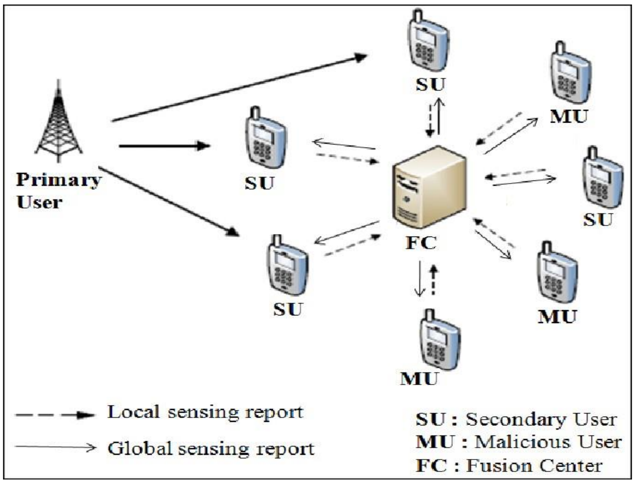

**Figure 1:** System Model

Each SU over a time slot 't' senses the spectrum and sends the perceived result to the FC using an energy detection technique. Each SU estimates the energy values under two hypotheses – H0& H1 as given by equation -1.

$$Y_k = \begin{cases} \sum_{u=1}^{m} n_k(u) ----> H_0 \\ \sum_{u=1}^{m} [h_k(u) + n_k(u)]^{2} ----> H_1 \end{cases} \quad (1)$$

**Figure 2:** Equation 1

here, $nk(u)$ is the additive white Gaussian noise with zero mean & variance –σx.$hk(u)$is the (u)th sample of primary user signal at the kth SU [8]. H0& H1 are the hypothetical representation of the absence and presence of PU signals. As the measured signal Yk is the sum of squares of i.i.d Gaussian variable, using central limit theorem, it can be shown that (for large values of 'k'),

$$Y_k = \begin{cases} \aleph(m\sigma_k^2, 2m\sigma_k^4)H_0 \\ \aleph\left((m+\delta_k)\sigma_k^2, 2(k+\delta_k)\sigma_k^4\right)\ H_1 \end{cases} \qquad (2)$$

**Figure 3:** Equation 2

$$\text{Where, } \delta_k = \frac{\sum_{i=1}^{m} h_i^2}{\sigma_x^2} s^2(i)$$

**Figure 4:** Equation (i)

Eq(1)represents the measured signal by k^th SU from a set of {1,2, S} at a particular time 't1'.

To obtain a final decision, the FC employs the majority voting rule. The performance of the decision rule is based on two metrics – Probability of detection (Pd) & probability of false alarm (pf). [9]

$$P_\sigma = Q_u(\sqrt{2\lambda}, \sqrt{\lambda})(3) \qquad \mathrm{P_f} = \frac{\sqrt{(\mathrm{K}, \lambda/2)}}{\sqrt{\lambda/2}}$$

$$(4)$$

$$\mathrm{P_m} = 1 - \mathrm{P_f} \qquad (5)$$

**Figure 5:** Equation 5

Equation 5 gives the probability of miss detection.

## IV. NN MODEL

Supervised and Unsupervised are the two types of ML algorithms. In supervised learning, the model learns the features from the data set with which it is trained, unlike unsupervised learning, where the model learns from its environment. NN is one reliable ML that imitates the neuron's artificial functioning. The NN model contains three layers :1. Input layer 2. Hidden layer 3. Output layer. Generally, the input layer has the number of features in the data set as the number of neurons [11]. The abstract representation of the inputs is learned by one or more hidden layers comprising neurons. The output layer represents the binary classification with one neuron. The NN adapts to the changes in the inputs by adjusting its weights (wi) and bias (b) to give the best possible output. Each neuron comprises an activation function (G) for learning complexities in the data. The result of each neuron can be mathematically modeled as the weighted sum of the input applied with the activation function as given in Eq (6).

$$O = G(\textstyle\sum_{i=1}^{n} w_i\, x_i + b) \qquad (6)$$

**Figure 6:** Equation 6

Sigmoid, Rectified Linear Unit (ReLU), their more commonly used activation functions. NN employs the feedforward technique to calculate the output. The hidden layers process the input fed to the input layer with its activation function, and then the data is forwarded to the subsequent layers.

The optimal model selection of NN depends on the type of input dataset, number of hidden layers, number of neurons, activation function, optimization algorithm, epochs, and batch size. The NN model consists of one input layer, one or many hidden layers, and one output layer.

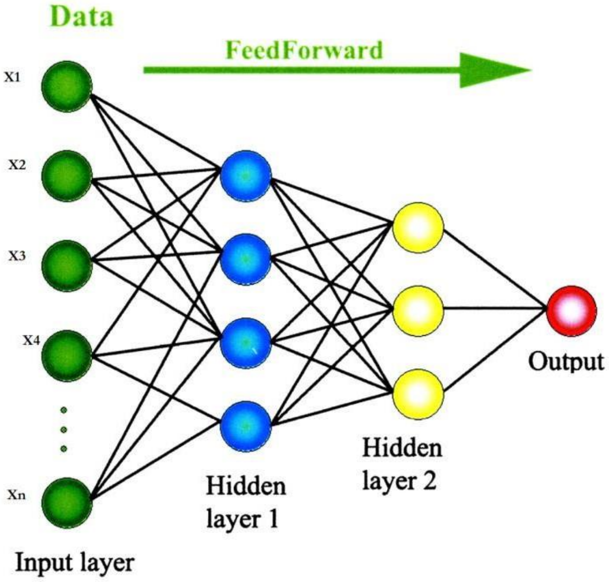

**Figure 7:** Model of NN

The selection of several hidden layers and the number of neurons for the proposed model was made based on the trial-and-error method as specified in [10] until the loss function reached minimum value and maximum accuracy. Also, the number of epochs and batch size were determined based on the mentioned criteria. ReLU activation function is used in the hidden layer [comparing ML] and sigmoid activation for the output layer to overcome the vanishing problem. To optimize the biases with high accuracy and speed, and adam optimizer is used. And the Binary cross-entropy is used as the loss function[13]. Also, K fold cross-validation of data is performed to prevent overfitting of data. Fig.2 shows the generalized model of NN where [X1,X2…Xn] are the inputs with 'n' features. The features are the five different trusts calculated for the SU's. Mathematically, Eq.(7) represents the inputs. $X = [X1, X2, X3, X4, X5\ ]$(7) The output of the ith neuron of the kth hidden layer can be formulated as Eq.(8)

$$h_i(k) = G_{ik}(\textstyle\sum_{j=1}^{m} w_{ik} h_{i(k-1)} + b_{ik} \qquad (8)$$

**Figure 8:** Equation 8

Where $h_i(k)$is the output, $G_{ik}$ is the activation function of the kth hidden layer, $w_{ik}$ and $b_{ik}$is the weight and bias of the ith neuron of kth hidden layer respectively.

$$Y_i(k) = G(\textstyle\sum_{i=1}^{m} w_i\,(k) h_i(k))(9)$$

**Figure 9:** Equation 9

Eq.(9) shows the output of the NN model, where G is the activation function of the output layer.

$$\textstyle\sum_{i=1}^{m} w_i\,(k) h_i(k)$$

**Figure 10:** Summation of all the neurons output of the kth hidden layer

The update of weights and biases are done by gradient descent algorithm, and the cross-entry loss function gives the change in error when the weights and biases are changed. The input data set to the NN model is generated from different trusts calculated from the sensing results of other SU's [12]. The features are trust history, requite confidence, registry trust, reliability trust, and cumulative trust.

**Naïve Bayes**

The Naïve Bayes ML is a family of ML algorithms that uses Bayes theorem to assume that the features are independent. This algorithm's high accuracy and speed on large data sets prove its advantage. The premise of independence in the input features simplifies the algorithm. The mathematical representation of Bayes Theorem by Eq.(10),

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

**Figure 11:** Equation 10

Where $P(A/B)$ and $P(B/A)$ are the conditional probabilities, P(A) and P(B) are the probabilities of the individual events. The input data set consists of various trusts computed using independent analytics about the proposed model. Therefore, Naïve Bayes ML could be used as a training algorithm.

**SVM (Support Vector Machine)**

Support Vector Machine is supervised ML which creates a hyperplane (decision boundary) to discriminate the classes for the labeled data given as the input. It constructs an optimal hyperplane with the help of support vectors. SVM also performs a mathematical transformation on the information using Kernels. Sigmoid, Radial Basis functions, linear are a few examples. The SVM is trained with the input dataset, and the performance is compared with the other MLs in this work.

**Logistic Regression**

LR is another type of ML from the field of statistics. It is a fast and efficient classifier for binary classifications. The predicted output is transformed into a logistic function using a sigmoidal curve. The output of the LR can be used as the probability of input data belonging to either class. The LR is also trained with the input data set for its performance comparison. The performance metrics chosen for comparison are accuracy, precision, recall, and f1-score.

## V. RESULTS AND DISCUSSION

The dataset is generated for 70 different SU's, randomly distributed over an area of 500sqm. A single PU is considered to have a transmitting power of about -4db over this area. All the SU's perform CSS to the FC that computes the trust values based on the sensing results given by the SU's. The channel used for communication is assumed to be free from fading and shadowing. Nearly 293 categories for 70 SU's have been analyzed, and five different trusts for each class have been computed. Thus, the MLs are trained with 1465 samples (293*5). Also, 80% of the input data set is used for training and 20% for validation.

**Optimal Model of NN**

First, an optimal model of NN is chosen with one input layer with several features as the number of neurons, three hidden layers with a total of 32 neurons, and one output layer with one neuron. The distribution of the neurons is considered as the 2i,i=0,1,2.[Detection] until the highest value of accuracy is reached. Table.1 gives the comparative values for the number of neurons, several hidden layers, epochs, batch size, and optimizer for optimal model selection of NN.

**Table 1.** Values of Optimal Model NN selection

| Sl.no | Activation Function | Number of Epochs | Batch Size | Number of Hidden Layers | Number of Neuron in Hidden Layer | Accuracy | Training Loss | Validation Loss |
|---|---|---|---|---|---|---|---|---|
| 1. | Sigmoid | 80 | 10 | 1 | 8 | 97.06 | 0.0881 | 0.2261 |
| 2. | Relu | 100 | 10 | 2 | 16 | 98.13 | 0.0661 | 0.2011 |
| 3. | Relu | 100 | 10 | 2 | 20 | 98.44 | 0.0578 | 0.1597 |
| 4. | Sigmoid | 50 | 20 | 3 | 24 | 88.47 | 1.214 | 1.345 |
| 5. | Relu | 100 | 10 | 3 | 24 | 95.99 | 0.0600 | 0.1442 |
| 6. | Relu | 100 | 10 | 3 | 32 | 99.47 | 0.0299 | 0.1432 |
| 7. | Relu | 100 | 10 | 2 | 36 | 95.72 | 0.0742 | 0.0742 |
| 8. | Relu | 100 | 10 | 2 | 40 | 98.6 | 0.0482 | 0.1260 |

The optimum value was chosen when accuracy was the highest with minor training and validation error weights. For the proposed system, with 32 neurons in the hidden layer – the model gave the highest accuracy of 99.47, training loss of 0.0299 and validation loss of 0.1432. For more than 40 neurons in the hidden layer, the model accuracy was reduced by 0.87, but the training and validation losses were minimal. Also, when the epoch was reduced below 100, there was a significant increase in training and validation loss. Thus, the optimal model was selected based on the highest accuracy, minimum losses, and 100 epochs.

Fig.12(a&b) gives the variation of training and validation loss with several epochs for the NN model as the number of ages nears 100, the loss functions near the minimum value.

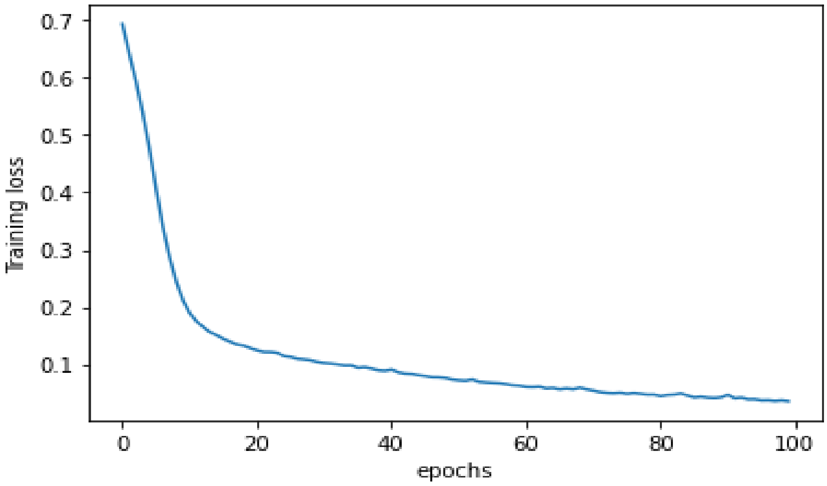

**Figure 12(a):** Variation of Training Loss with epochs for NN model

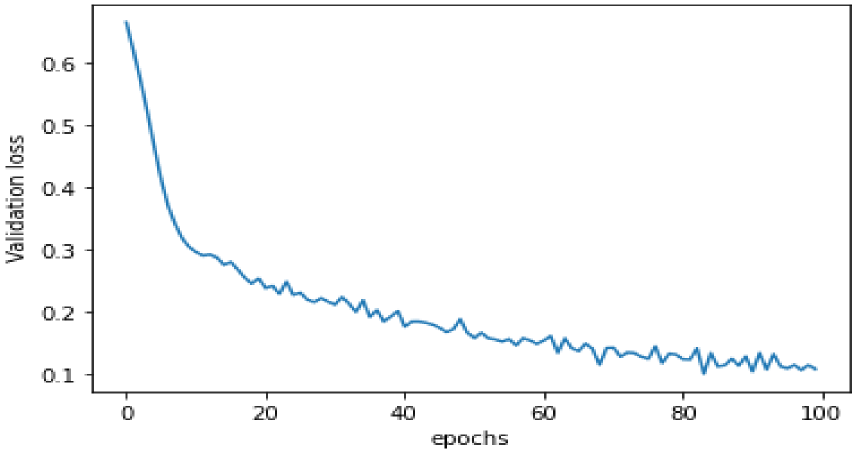

**Figure 12(b):** Variation of Validation Loss with epochs for NN model

Fig.13 shows the variation of accuracy with the number of epochs. There is a stipulated increase in accuracy value with several periods.

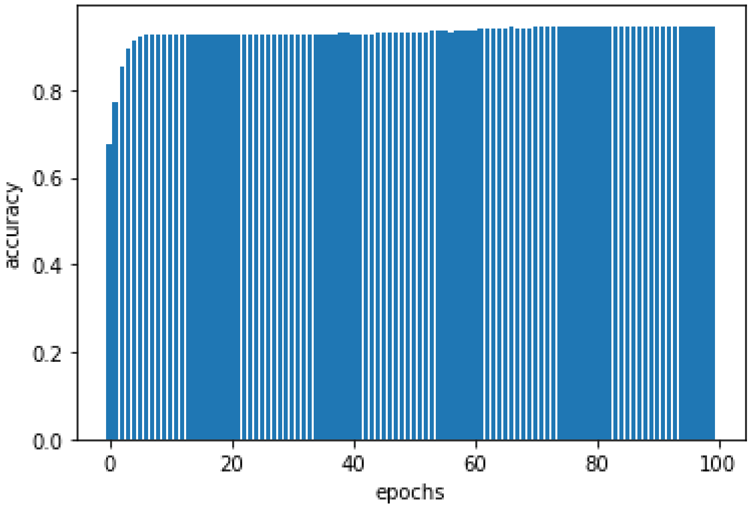

**Figure 13:** Variation of Accuracy with epochs for NN model

SVM, NB, and LR MLs are trained with the same dataset and measured performance metrics. Table.2 gives the comparative values of various performance metrics for different ML's.

**Table 2.** Performance Metrics for different ML algorithms

| SL. No | ML | Accuracy | Precision | | Recall | | F1-score | |
|---|---|---|---|---|---|---|---|---|
| | | | Malicious (Class 0) | Honest (Class 1) | Malicious (Class 0) | Honest (Class 1) | Malicious (Class 0) | Honest (Class 1) |
| 1. | NN | 99.4 | 95 | 75 | 98 | 50 | 96 | 60 |
| 2. | SVM | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 3. | NB | 98 | 98 | 100 | 100 | 80 | 99 | 89 |
| 4. | LR | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

The Logistic Regression and the SVM prove their best accuracy, precision, recall, and f1 score for the given data set.

**Validation of the model**

After the model is trained and tested, cross-validation is performed using k-Fold cross-validation (cv) to prove an unbiased evaluation of the optimal model fit on the training data set. Also, cross-validation will avoid overfitting the data. In k-fold cv, the training set is divided into smaller k sets called folds. Then the model is trained using k-1 folds as training data, and the kth fold is used to validate the trained model. The performance metric fork fold is accuracy; the average values are computed using various k folds. Less variability of error, less bias, less computation are the advantages of k-fold cv. The k-fold validation for different values of k for all the MLs has been tabulated in Table.3.

**Table 3.** k-Fold cross-validation values for different ML algorithms

| SL No. | ML | k-Fold | Accuracy(%) |
|---|---|---|---|
| 1. | NN | 2 | 96.07 |
| | | 5 | 97.94 |
| | | 10 | 97.79 |
| | | 20 | 97.40 |
| 2. | SVM | 2 | 92.8 |
| | | 5 | 92.8 |
| | | 10 | 92.8 |
| | | 20 | 92.9 |
| 3. | NB | 2 | 98.31 |
| | | 5 | 98.31 |
| | | 10 | 98.31 |
| | | 20 | 95.6 |
| 4. | LR | 2 | 100 |
| | | 5 | 100 |
| | | 10 | 100 |
| | | 20 | 100 |

The k-fold cv is done for four different values of k, k=2,5,10&20 for all the ML's. It can be inferred that the accuracy is 100% in the LR ML for all the 'k' values. LR has identified all the malicious users as malicious and honest SU's as simple for the data used for cross-validation. Thus, LR ML is the best algorithm for the multifactor trust-based data set.

## VI. CONCLUSION

In this paper, the performance metrics of different classifiers as NN, SVM, NB, and LR are calculated for identifying the MUs by training with a data set created by calculating multifactor trusts for each SU. The multifactor trust data set relies on the spectrum sensing results of the SU's. Also, an optimal NN model was designed for the given data set, and its performance was also measured. The MLs were cross-validated for different 'k' values, and the accuracy was measured. Overall, LR ML proved to be the best classifier for the dataset. As future work, multiclass classifiers for identifying different types of SSDF attackers using the same and other ML techniques could be performed with the same dataset.

## VII. REFERENCES

[1] Olga León and K. P. Subbalakshmi (2017) "Cognitive Radio Network Security," Cognitive Radio, 30 pages.

[2] Doha Hamza, Sonia Aissa and Ghassane Aniba (2014) "Equal Gain Combining for Cooperative Spectrum Sensing in Cognitive Radio Networks", IEEE Trans Wireless Communications 13(8), Pages 4334-4345.

[3] Jingyu Feng ,Man Zhang ,Yun Xiao and Hongzhou Yue (2018) "Securing Cooperative Spectrum Sensing Against Collusive SSDF Attack using XOR Distance Analysis in Cognitive Radio Networks", Sensors,18(2) pp(1–14).

[4] [4]X. Wang, M. Jia, Q. Guo, X. Gu and G. Zhang (2017) "Reputation-based cooperative spectrum sensing algorithm for mobile cognitive radio networks," in China Communications, vol. 14, no. 1, pp. 124-134,

[5] [5]L. Zhang, Q. Wu, G. Ding, S. Feng, J. Wang (2014),"Performance analysis of probabilistic

[6] softssdf attack in cooperative spectrum sensing", EURASIP Journal of Advance Signal Processing (1) (2014) pp 1–9.

[7] Farmani. F, Jannatabad. A and Berangi. R (2011), "Detection of SSDF attack using SVDD algorithm in cognitive radio networks," Proc. of Int. Conf. on Computational Intelligence, Communication Systems and Networks, pages 201-204.

[8] H. Zhu, T. Song, and J. Wu (2018) "Cooperative spectrum sensing algorithm based on support vector machine against SSDF attack," in Proc. IEEE Int. Conf on Communication Workshops, Pages 1–6.

[9] H. Chen, M. Zhou, L. Xie, K. Wang and J. Li,(2016) "Joint Spectrum Sensing and Resource Allocation Scheme in Cognitive Radio Networks with Spectrum Sensing Data Falsification Attack," IEEE Trans. on Vehicular Tech., vol.65, no.11, pp.9181 9191.