2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)

Malicious Relay Node Detection with Unsupervised Learning in Amplify-Forward Cooperative Networks

Yeliz Yengi
Dept. of Computer Eng.
Kocaeli University
Kocaeli, Turkey
yelizyengi@gmail.com

Adnan Kavak
Dept. of Computer Eng.
Kocaeli University
Kocaeli, Turkey
akavak@kocaeli.edu.tr

Hüseyin Arslan
Dept. of Electrical and Electronics Eng.
İstanbul Medipol University
İstanbul, Turkey
arslan@usf.edu

Kerem Küçük
Dept. of Computer Eng.
Kocaeli University
Kocaeli, Turkey
kkucuk@kocaeli.edu.tr

Halil Yiğit
Dept. of Information Systems Eng.
Kocaeli University
Kocaeli, Turkey
halilyigit@kocaeli.edu.tr

***Abstract*—This paper presents malicious relay node detection in a cooperative network using unsupervised learning based on the received signal samples over the source to destination (S-D) link at the destination node. We consider the situations in which possible maliciousness of the relay is the regenerative, injection or garbling type attacks over the source signal according to attack modeling in the communication. The proposed approach here for such an attack detection problem is to apply unsupervised machine learning using one-class classifier (OCC) algorithms. Among the algorithms compared, One-Class Support Vector Machines (OSVM) with kernel radial basis function (RBF) has the largest accuracy performance in detecting malicious node attacks with certain types and also detect trustable relay by using specific features of the symbol constellation of the received signal. Results show that we can achieve detection accuracy about 99% with SVM-RBF and k-NN learning algorithms for garbling type relay attacks. The results also encourage that OCC algorithms considered in this study with different feature selections could be effective in detecting other types of relay attacks.**

Keywords—physical layer security, cooperative communication, unsupervised learning, one class classifier, detection

I. INTRODUCTION

Physical layer (PHY) security is critical for wireless communications, while it serves as a dam for increasing the reliability and performance of communication [1]. PHY security is also the centrepiece of the cooperative communication which can achieve the diversity with relay nodes is of great importance to obtain desired performance [2]-[4]. Under some circumstances, it may be unreliable to cooperate with some relays which might be malicious, although diversity achieved with these relays is quite good. This is because of the inherent vulnerability of such collaborative systems [5]. Therefore, detection of the maliciousness of the relay node before cooperating with that relay can prevent possible attacks. Detection of the malicious relay has been the focus of some recent studies [6]-[12]. In [6]-[9], malicious node detection studies are based on statistical approaches with the corner of detectable or undetectable situations. However, it is complicated to distinguish two signals with a very similar distribution using the statistical approaches. In [9]-[11], studies use supervised learning for separating signals or detecting malicious sources. In [12], reinforcement learning is applied for spoofing detection based on some hypothesis. Unsupervised learning in PHY security is one of the rarely studied approaches for cooperative networks [13].

The contribution of this work is to prevent possible relay attacks in the PHY layer of the cooperative network before further processing its received signal at the destination. Specifically, we propose to apply machine learning for malicious relay detection in cooperative communication networks. The pattern of the received signals from the source is determined by unsupervised learning techniques that employ one-class classifier algorithms (OCC) such as Support Vector Machines (SVM) with various kernel functions, k-Nearest Neighbors (k-NN), and Isolation Forest [14]. We consider that the relay node possesses various relay attacks such as false data injection, regenerative data, and garbling type attacks over the signal received from the source node.

In Section II, we describe the system model and problem definition. In Section III, we define the feature sets and briefly explain the algorithms applied. In Section IV, we present the performance evaluation results.

II. SYSTEM MODEL AND PROBLEM DEFINITION

A. System Model

The system is modeled as a single relay cooperative networks as in Fig. 1 which describes the scenario in which communication between a source node (S) and a destination node (D) is supported via a relay node (R). We assume that the relay node implements amplify-forward (AF) relaying protocol and has no feedback to the source node. The destination node receives signals from the source and relay nodes separately within two consecutive time frames.

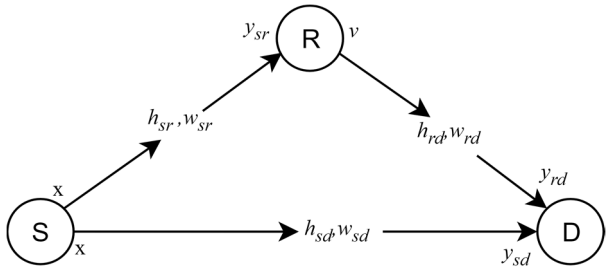

Fig. 1. The system model.

978-1-7281-3012-5/19/$31.00 ©2019 IEEE

The source transmits QPSK modulated i.i.d symbols x. Within a frame period, the total of N QPSK symbols are transmitted, i.e., it transmits signal vector $\mathbf{x}^{1\times N}$. We assume that the destination and relay nodes receive the signal transmitted from the source during the first time period (t_1) as given by,

$$y_{sr}=h_{sr}\mathrm{x}+w_{sr}, \tag{1}$$

$$y_{sd}=h_{sd}\mathrm{x}+w_{sd}. \tag{2}$$

Then, during the next time period (t_2), the relay node amplifies y_{sr} by an amplification factor of α, and broadcasts this signal. The received signal at the destination node is written as,

$$v=\alpha y_{sr}, \tag{3}$$

$$y_{rd}=h_{rd}v+w_{rd}. \tag{4}$$

In Equations (1)-(4), w_{sr} , w_{sd}, and w_{rd} are i.i.d zero mean additive white Gaussian noise (AWGN) with variance σ^2 in the source-to-relay (S-R), source-to-destination (S-D), relay-to-destination (R-D) links, respectively. The complex fading channel coefficients are denoted by h_{sr} , h_{sd}, and h_{rd} in the S-R, S-D, and R-D links, respectively. We assume channel coefficients remain unchanged during symbol period.

B. Problem Definition

In traditional cooperative networks, the received signals from the source and relay nodes are combined using some diversity techniques at the destination node to achieve high signal quality. However, in some cases, the relay node may act maliciously and transmits incorrect information to the destination node. Cooperative combining techniques ignore this malicious behavior of the relay nodes, and just tries to make use of diversity provided by them. Given that the signal transmission from source S to destination D is received correctly and reliably, the crucial issue is to detect whether the relay transmission is trustable before processing its signal for diversity combining purposes. So, the purpose of this study is to find a solution to this problem under various relay attack types. We define three different malicious relay attack models as described below.

1) False Data Injection Attack,

In false data injection attack, the malicious relay node simply inserts another signal to the received signal at the relay node, i.e.,

$$\widetilde{y_{sr}}=y_{sr}+a, \tag{5}$$

where $a^{1\times N}$ is the injection data vector which is assumed to be a non-zero complex random variable and independent from source signal x.

2) Garbling Attack

In this case, the malicious relay node randomly creates a new permutation index from the symbol constellation, i.e., relocating the indices of the current symbol vector instead of transmitting the original data,

$$\widetilde{y_{sr}}=P(y_{sr}), \tag{6}$$

where P is the permutation function that changes the index of data using uniform distribution.

3) Regenerative Attack

Relay node replaces the original transmitted signal x with the newly generated symbols $\tilde{\mathrm{x}}$, before forwarding to the destination.

$$\widetilde{y_{sr}}=\tilde{\mathrm{x}}. \tag{7}$$

For all attack models, we assume that attack signal power is equal to signal power generated at the source, i.e. $\sigma_{\mathrm{x}}^2=\sigma_{\widetilde{y_{sr}}}^2$. When the malicious relay node transmits the signal using one of the attack models, the received signal by the destination is written as,

$$\tilde{v}=\alpha\widetilde{y_{sr}}, \tag{8}$$

$$\widetilde{y_{rd}}=h_{rd}\tilde{v}+w_{rd}. \tag{9}$$

III. Attack Detection with Machine Learning

In this work, we propose to apply unsupervised machine learning techniques to detect malicious relay attacks. In multi-class classification (supervised learning), there are predefined categories and undefined data object may not be suitable to be classified in any category of the classifier. It is difficult to choose features that would be used to handle the best distinction between the target and the outlier class objects. OCC builds a model of positive samples in the absence or weakness of the negative sample. Outlier detection and specific sample learning have been addressed and implemented as OCC problem in many research themes [15]. Here, we investigate the feasibility of using OCC algorithms for untrustable relay detection in a cooperative relay network. We consider that signal samples received over S-D link constitute positive class objects for training. We try to determine whether signal samples received over R- D link fall in this positive class and attack models fall in an outlier.

A. Feature Extraction

The signals received at the destination have some features related to the QPSK symbols as shown in Fig. 2. Let y_n be the n-th received symbol at the destination over the S-D link in Fig. 1. We define the features of y_n as the following: The first feature $f_n^{(1)}$ is the amplitude of the symbol y_n, i.e., $f_n^{(1)}=|y_n|$. The second feature is the position information of symbol y_n in the constellation, i.e. $f_n^{(2)}=\angle y_n$. The third feature, $f_n^{(3)}$ is the phase difference between consecutive symbols as given by,

$$f_n^{(3)}=\angle y_n-\angle y_{n+1}, \tag{10}$$

The feature vector of symbol y_n is expressed as,

$$\mathbf{f}_n=[f_n^{(1)}f_n^{(2)}f_n^{(3)}]. \tag{11}$$

This feature vector provides information about changes in symbol position of the symbol y_n in the constellation considering the noise and channel effects on the signal. Feature vectors of N consecutive symbols is merged to

construct the feature vector of the m-th frame in the training phase,

$$\mathbf{z}_m=[\mathbf{f}_1\ \mathbf{f}_2\cdots\ \mathbf{f}_N]^T \quad m=1,2,\ldots,M \tag{12}$$

where $L=3N$ is the size of the feature vector $\mathbf{z}_m$ and M is the number of frames generated during the training phase. Hence, the overall training feature set for our problem is given by

$$\mathbf{Z}=[\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \ldots\ \mathbf{z}_M\ ] \tag{13}$$

where the size of feature set matrix $\mathbf{Z}$ is $L\times M$, i.e., $\mathbf{Z}\subseteq R^E$, $E=L\times M$.

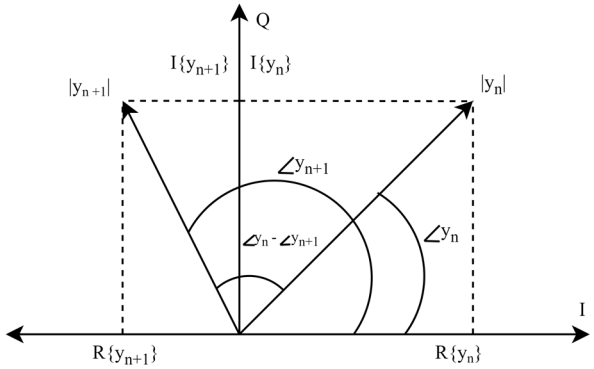

Fig. 2. Features of two consecutive symbols in the constellation.

B. One Class Classifier

A variety of one-class classifier algorithms exists in the literature. Here, we investigate the feasibility of applying One-class Support Vector Machines (OSVM), Isolation Forest and One-class Nearest Neighbor algorithms for our problem description.

1) One Class SVM (OSVM)

We briefly describe the OSVM proposed by Scholköpf et al. in [15] taking into account for our specific problem definition. The main goal of OSVM is to produce a decision function based on the feature vectors in Eq. (13) within training dataset. The goal is formulated as the optimization problem given by [16],

$$\min_{w,\xi_m,\rho} \quad \frac{1}{2}\|\mathbf{w}\|^2+\frac{1}{vM}\sum_{m=1}^{M}\xi_m-\rho \tag{14}$$

$$\text{subject to } \mathbf{w}^{\mathrm{T}}\phi(\mathbf{z}_m)\geq\rho-\xi_m,\ \xi_m\geq 0,\ m=1,2,\ldots,M$$

where $\phi(.)$ is nonlinear mapping function for feature vectors in the training set, $\mathbf{w}$ is the weight vector for the model, ξ_m is the regularization parameter, ρ is the bias for margin maximized, the parameter $v\in(0,1)$ is an upper bound on the fraction of outliers and lower bound on the fractions of support vectors. In order to decide whether the test sample vector $\mathbf{z}_j$ falls above hyperplane or outlier, nonlinear kernel function $\mathrm{K}(\mathbf{z}_m,\mathbf{z}_j)$ is used in the decision function which is given by [17],

$$\mathrm{g}_{\mathrm{OSVM}}(\mathbf{z}_j)=\mathrm{sgn}\left(\sum_{m=1}^{M}\mu_m\mathrm{K}(\mathbf{z}_m,\mathbf{z}_j)-\rho\right) \tag{15}$$

$$0<\mu_m<\frac{1}{vM}$$

where μ_m is the Lagrange multiplier obtained by maximization of margin with $\phi(.)$ function. In our malicious relay node attack detection problem, we employ three different kernel functions $\mathrm{K}(\mathbf{z}_m,\mathbf{z}_j)$, namely Polynomial, Sigmoid, and Radial Basis Function (RBF) function as given in Equations (16), (17), and (18), respectively [18].

$$\mathrm{K}(\mathbf{z}_m,\mathbf{z}_j)=(\mathbf{z}_m^T\mathbf{z}_j+1)^r \tag{16}$$

$$\mathrm{K}(\mathbf{z}_m,\mathbf{z}_j)=\tanh(\alpha\mathbf{z}_m^T\mathbf{z}_j+c) \tag{17}$$

$$\mathrm{K}(\mathbf{z}_m,\mathbf{z}_j)=\mathrm{e}^{-\gamma\|\mathbf{z}_m-\mathbf{z}_j\|^2} \tag{18}$$

where r is the degree of the polynomial, α and c are the intercept constants.

2) One Class Nearest Neighbor (ONN)

The Nearest Neighbor method in OCC is called Nearest Neighbor Description (NN-d). This method uses a distance of first nearest neighbor by avoiding the explicit density estimation to derive local density estimation. In NN-d estimation, cells are centered around test objects and hypersphere in d dimensions, and are grown until capturing k objects from the training set. For our problem definition, the distance of the feature set $\mathbf{z}_m\in\mathbf{Z}$ to the nearest neighbour in the training set $\mathbf{z}_j\in\mathbf{Z}$ is given by [14],

$$\mathrm{g}_{\mathrm{ONN}}(\mathbf{z}_j)=\frac{k/M}{V_k(\|\mathbf{z}_m-\mathbf{z}_j\|)} \tag{19}$$

where V_k is the volume of the cells containing the training set object.

3) One Class Isolation Forest (OIF)

The main idea of One Class Isolation Forest is to define the anomaly score for malicious relay signals instead of a trustable source signal. The Isolation Forest isolates samples by randomly selecting the features from the feature vector $\mathbf{z}_j$ in Eq. (12) and comparing it with min-max values. Recursive partitioning randomly creates binary tree structure to isolate a sample. The average path length of random trees is a measurement of normality for our decision function. In our problem, the anomaly score of Isolation Forest algorithm is written as [18],

$$\mathrm{g}_{\mathrm{OIF}}(\mathbf{z}_j,\mathbf{Z})=2^{-\frac{E(h(\mathbf{z}_j))}{c(\mathbf{Z})}} \tag{19}$$

where $E(.)$ denotes average of $h(\mathbf{z}_j)$, which is path length, and $c(\mathbf{Z})$ is the average path length of given $\mathbf{Z}$ for all feature sets. If the score is greater than 0.5, it indicates anomalies.

IV. Experimental Results

The simulation results are investigated for two perspectives, i.e., trustable relay detection and malicious relay detection. The first step of simulation is to generate signals as described in Section II. Next, the features are extracted for generated IQ constellation. The results are presented in terms of detection accuracy with respect to varying SNR conditions for the aforementioned attack models.

A. Dataset Generation

Matlab is used to generate a dataset of the system model shown in Fig. 1 and to obtain the feature vectors defined for each signal sample in Fig. 3. Table I summarizes the characteristics of the dataset. The training dataset is generated for the SNR scales from 5 dB to 50 dB. Positive and negative (trustable and malicious) samples are presented to the classifier separately for each SNR value.

B. Simulations

Monte Carlo simulations are performed (repeatedly 100 times) to generate datasets. One Class Classifier algorithms are simulated using Scikit Learn machine learning library [18]. We set v parameter as 0.1 for smoothness in Eq. (14), and r=3 in Eq. (15) for the polynomial kernel in OSVM classifier. The number of the base estimator in the ensemble is 100 for the Isolation Forest. The number of neighbors k in Eq. (19) is set to 64 for nearest neighbors. Classifiers are executed 20 times for the same condition and we calculate the average of the accuracy various these executions.

TABLE I. Dataset Properties

The number of symbols (N) per frame	64
Total number of training samples (M)	10K (1K every SNR value)
The number of features (L) per frame	64×3
Fading channel type	Flat Fading (Rayleigh)
Modulation type	QPSK
SNR values	5-50 dB

C. Results

The results are presented in Fig. 3 through Fig. 6. The results show that SNR changes have a small effect on accuracy for both trustable and malicious relay detection. This shows us that the learning process is robust to detect any noise effect on the signal. Trustable relay detection is important as well as malicious relay detection for the reliability of the system model.

OSVM with RBF kernel, Nearest Neighbor and Isolation Forest are powerful to detect trustable relays in any SNR condition higher than 80% accuracy level, but OSVM with RBF kernel has 99% accuracy as given in Fig. 3. OSVM with RBF kernel also has the best performance for detecting malicious relays for data injection and garbling type attacks, i.e., for false data injection nearly 70% accuracy in Fig. 4, for garbling signal attack nearly 99% accuracy in Fig. 6. Nearest Neighbors approach performs best on garbling signal attack with the accuracy of about 99% accuracy whereas its performance is among the worst with accuracy about 25% for injection type and regenerative type attacks. All kernels in OSVM have the accuracy performance between 63% and 73% for the regenerative attack as presented in Fig. 5.

According to simulation results, our proposed feature sets in Eq. (12) is feasible to detect any maliciousness in the relay node especially when it creates garbling or injection type attacks.

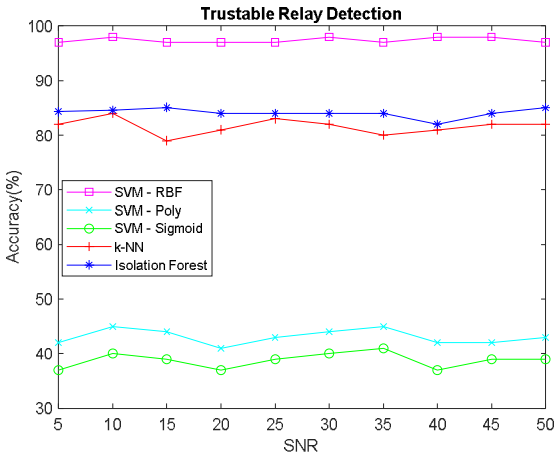

Fig. 3. Trustable relay detection accuracy for all algorithms.

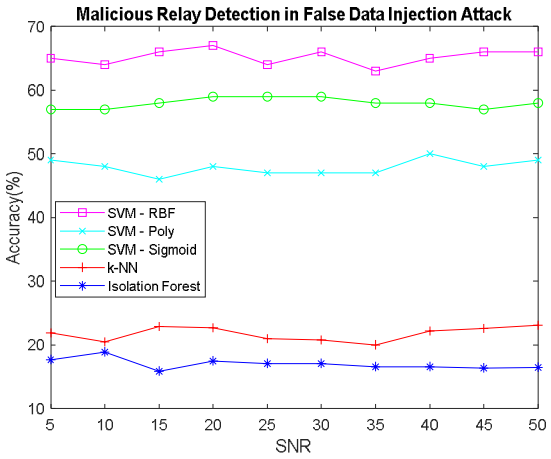

Fig. 4. Malicious relay detection for false data injection attack, accuracy for all algorithms.

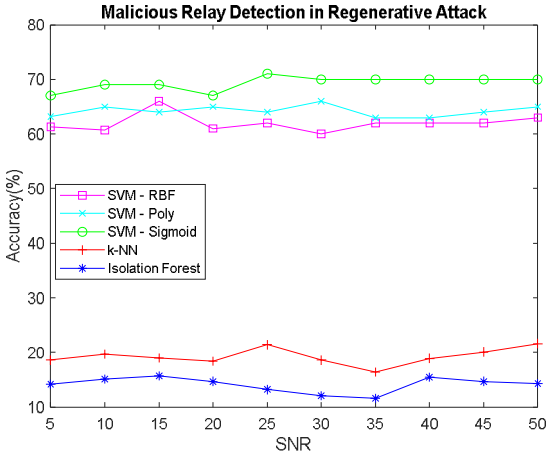

Fig. 5. Malicious relay detection for the regenerative attack, accuracy for all algorithms.

Trustable relay detection performance of OSVM poly and OVSM sigmoid kernel are observed to be 42% and 38% accuracy in Fig. 3, respectively, because of overfitting in training datasets. Isolation forest method fails through to detect all attack types with approximately 20% accuracy. Since Isolation Forest chooses the subfeature space randomly for binary trees and isolates negative samples, this approach results in missing the attack samples.

Regarding complexities of the methods used in this study, SVM and Isolation Forest methods have the training phases during classification, and therefore overall compleities of these methods are higher than the complexity of ONN. Complexities of SVM and Isolation Forest in the training phase are $O(M^2L+M^3)$ and $O(tM\log M)$, respectively, where t is the number of trees in the Isolation Forest method.

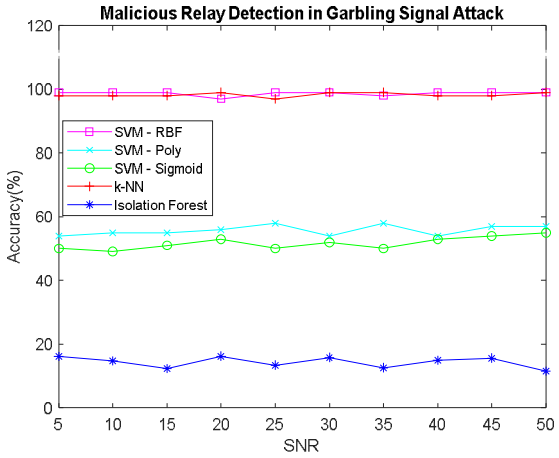

Fig. 6. Malicious relay detection for garbling signal attack, accuracy for all algorithms.

V. Conclusion

In this paper, we have investigated physical layer attack detection of malicious relay nodes for the cooperative network using unsupervised learning. We have assumed a single relay and also direct link between source to destination node. For this network model, we have employed OCC algorithms namely SVM, KNN, and Isolation Forest to detect various malicious relay attacks such as data injection, data regeneration, and garbling type attacks. Our approach uses symbol properties of QPSK symbols to set the feature vectors of the algorithms. The best detection accuracy is obtained by OSVM-RBF kernel algorithm with 99% percent accuracy for the trustable relay and between the ranges of 70%-99% accuracy for the attack detection. For the feature set we define, SVM with RBF kernel and k-NN learning algorithms are effective in detecting garbling type relay attacks with accuracy about 99%. Results encourage that with different feature selections unsupervised learning methods studied could be effective in detecting another type of relay attacks. Results in this paper are encouraging to demonstrate the effectiveness of unsupervised learning for detecting malicious relay attack.

References

[1] Yang, N., Wang, L., Geraci, G., Elkashlan, M., Yuan, J. and Di Renzo, M., "Safeguarding 5G wireless communication networks using physical layer security," *IEEE Commun. Mag., vol. 53, no. 4, pp. 20–27, Apr. 2015.*

[2] Sendonaris, A., Erkip, E. and Aazhang, B., "User cooperation diversity. Part I. System description," *IEEE Trans. Commn., vol. 51, no. 11, pp. 1927–1938, Nov. 2003.*

[3] Laneman, J. N., Tse, D. N. C. and Wornell, G. W., "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. *50, no. 12, pp. 3062–3080, Dec. 2004.*

[4] Nabar, R. U., Bolcskei, H. and Kneubuhler, F. W. "Fading relay channels: Performance limits and space-time signal design," *IEEE J. Sel. Areas Commun., vol. 22, no. 6, pp. 1099–1109, Aug. 2004.*

[5] Dehnie, S., Sencar, H. T., Memon, N., "Cooperative diversity in the presence of a misbehaving relay: Performance analysis," in *Proc. IEEE Sarnoff Symp., Princeton, NJ, USA, May 2007, pp. 1–7.*

[6] Lv, T., Yin, Y., Lu, Y., Yang, S., Liu, E., and Clapworthy, G., "Physical Detection of Misbehavior in Relay Systems with Unreliable Channel State Information", *IEEE Journal on Selected Areas in Communications, 2018.*

[7] Mao, Y. and Wu, M., "Tracing malicious relays in cooperative wireless communications," *IEEE Trans. Inf. Forensics Security, vol. 2, no. 2, pp. 198–212, Jun. 2007.*

[8] Lo, L.-C. and Huang, W.-J., "Misbehavior detection without channel information in cooperative networks," in *Proc. IEEE 74th Veh. Technol. Conf. (VTC Fall), San Francisco, CA, USA, Sep. 2011, pp. 1–5.*

[9] Hou, W., Wang, X., and Refaey, A., "Misbehavior detection in amplifyand-forward cooperative OFDM systems," in *Proc. IEEE Int. Conf. Commun. (ICC), Budapest, Hungary, Jun. 2013, pp. 5345–5349.*

[10] Riyaz, S., Sankhe, K., Ioannidis, S., and Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification", *IEEE Communications Magazine, 56(9), 146-152, 2018.*

[11] Kulin, M., Kazaz, T., Moerman, I., and De Poorter, E., "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications.", *IEEE Access, 6, 18484-18501, 2018.*

[12] Xiao, L., Li, Y., Han, G., Liu, G., and Zhuang, W., "PHY-layer spoofing detection with reinforcement learning in wireless networks." *IEEE Transactions on Vehicular Technology, 65(12), 10037-10047, 2016.*

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 2825-2830, 12 October 2011.

[14] Tax, D. M. J., "One-class classification: concept-learning in the absence of counter-examples" [Ph. D. thesis]. *Delft University of Technology, 2001.*

[15] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. "Support vector method for novelty detection.", *In Advances in neural information processing systems*, pp. 582-588,2000.

[16] Cortes, C. and Vapnik, V., "Support vector networks", Machine Learning, 20(3):273 – 297, September 1995.

[17] Fletcher, R., "Practical methods of optimization". John Wiley & Sons, 2013.

[18] Liu, F. T., Ting, K. M., & Zhou, Z. H. Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 3, 2013.

[19] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.