Cognitive Radio Jamming Attack Detection Using an Autoencoder for CRIoT Network

V. Nallarasan · Kottilingam Kottursamy

Accepted: 4 August 2021 / Published online: 15 August 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
IoT network-connected devices are increasing day by day. It is impossible to allocate a spectrum for all IoT devices. This spectrum scarcity can be solved by cognitive radio-based dynamic spectrum sharing, which is referred to as Cognitive Radio Internet of Things (CRIoT). The jamming physical layer attack in CRIoT resulting in a denial of cognitive radio services and make spectrum underutilization. Continuous jamming can be detected quickly based on time delay on spectrum access, but discrete random jamming detection is challenging. This article proposes an autoencoder-based jamming attack detection in cognitive radio. The jamming detection problem is modeled as anomaly detection. The autoencoder is used to detect the anomaly signal component and the jammer. The simulation of random jamming attack detecting with time instant of attack is carried out to mitigate it. The proposed mechanism able to detect the jammer with 89% of accuracy.

Keywords Cognitive radio · Internet of Things · Various attacks · Architecture-based jamming attack

1 Introduction

1.1 Background
Internet of things is ever-growing with an increased number of connected devices in the order of millions and millions of devices. In the future, it will not be possible to allocate the spectrum for those connected devices because most of the spectrum is allocated for the other wireless services. Even though there is a new spectrum band proposed for the scarcity issue like millimeter-wave spectrum and Tera Herz spectrum, they are still at laboratory and simulation level only. There is a lot of deployment issue need to be addressed in those new spectral band. Moreover, those new spectral band hardware costs will be very
high, making it possible to accommodate IoT applications. The only feasible solution for this issue will be applying the cognitive radio concept in the IoT network for dynamic spectrum sharing. Cooperative cognitive is suitable primarily [1] for IoT applications since, by default, all IoT nodes are engaged with sensing jobs already. It also provides energy efficiency [2]. The limited availability of spectrum and inefficient usage of range opens a concept called cognitive Radio Networks (CRN) [3], which can be effectively utilized for the IoT network. Routing algorithms for Cognitive Radio Based IoT network analyzed with channel switching cost, end-to-end delay cost, energy efficiency and bandwidth dependent cost [3]. Machine learning and deep learning applied Cognitive IoT network provides much promising solution because it establishes mathematical models using observations, i.e., training data then the model can be used, to predict or make decisions [4–6]. Those approaches enable learning and improvement from experience automatically. An integrated cognitive radio (CR) with the IoT called CR-IoTNet is introduced [7]. the proposed CR-IoTNet, given with multiple primary user (PU) base stations and SU devices and joint spectrum sensing and optimal allocation of spectrum being proposed. In this, an intelligent fusion center (IFC) is utilized for signal spectrum sensing decisions. Support vector machines (SVM) are employed to learn and adapt to network dynamics and identify the PU spectrum usage. Trained SVM classifier provides 95.11% accuracy.

There are many kinds of literature that explored the spectrum sharing mechanism for IoT. There are many couples of works in the direction of resource management [8]. The security issues over the physical layer of cognitive radio are summarized [9]. The various threat types, detection mechanisms, and countermeasures are reported.

But the security aspect of the cognitive radio for spectrum sharing in IoT applications is not explored much. There are many physical layers attacks in the cognitive radio that disables the dynamic spectrum sharing [9–13] like Primary user emulation attack, data falsification jamming attack. Among those attacks, jamming attacks are very simple to make, which could be a prominent one in the IoT application with cognitive radio. The jamming attack with continuous jamming is easier to detect. But the intelligent jammer will randomize the jamming attack, which is very difficult to detect. Thus this research article is intended to solve such kind of ransom jamming attack. The detection of such jamming attacks is the first step to mitigate that attack. Under a random attack scenario detecting the attack time instant will be very much essential to alleviate the attack. So, this research article intended to detect the attack with the exact time instant of the attack.

1.2 Literature Survey/Related Work

Few kinds of literature deal with CR in IoT and jamming attack detection.

The jamming attack is one of the leading security issues for spectrum sharing in cognitive radio, especially in an IoT network. CR network’s security issue of proactive jamming and reactive jamming in spectrum sharing is addressed [14]. The channel assignment process in the presence of both proactive and reactive jamming attacks is carried out with a probabilistic-based channel assignment mechanism. The proposed mechanism tries to minimize the invalidity ratio of CR packet transmissions with a delay constraint. The statistical information of primary users’ activities, fading conditions and jamming attacks over idle channels are used as base data to achieve the lowest invalidity ratios with a quality-aware channel assignment algorithm.

Another jamming attack work for the cognitive radio network with an anti-jamming task with the help of a Markov decision process and deep reinforcement learning method is
proposed [15]. The mechanism used to learn a policy to maximize the rate of successful transmission. A Double Deep Q Network (Double DQN) model is used to detect the jammer. The Q network is trained using the Transformer encoder to estimate action-values pair from raw spectrum data.

Channel hopping is one of the schemes for anti-jamming in cognitive radio, which requires pre-sharing of access information and not taking care of the traffic loads’ variations. An anti-jamming scheme with the dynamic adjustment of the sending/receiving ratio to maximize the throughput is proposed [16]. The mechanism works based on Load Awareness with Anti-channel hopping for anti-jamming, which uses extended Langford pairing.

The coexistence of IoT and CRN makes it possible to deploy a broad range of solutions. The time-sensitive IoT applications suffer from communication security, especially when CR concepts are utilized in IoT, prone to more attacks. A security-aware routing protocol with jamming attacks is presented [17]. This mechanism assigns the most secure channel for every hop of communication by solving an optimization problem. An Ensemble-based Jamming Behaviour Detection and Identification (E-JBDI) technique is also presented to identify the behavior anomaly of jamming attacks. Results also show the proposed mechanism achieves accuracy and precision-recall rates of approximately 100%.

An active anti-jamming method based on frequency diverse array (FDA) radar phase center is presented [18]. This cognitive activity ensures the radar difficult for a jammer to detect or locate during the normal operation. The Bayesian filter is applied to realize cognitive beamforming to make anti-jamming.

The jamming issue in the Cognitive radio (CR) WiFi network with dynamic channel switching is presented [19]. This integration is called a Cognitive WiFi (CWF) network. A new LU-MAC protocol is proposed for the CWF network, which can work dynamically in licensed and unlicensed bands to increase jamming resistance. The protocol has a new control frame, Data Channel Switching (DCS) process and coordination modules, control channel switching process. Few algorithms are analyzed for the anti-jamming, namely 1. Intrusion Detection System (IDS), 2. Random Frequency Hopping Pattern (RFHP) 3. Statistical-based Reactive Channel Switching (SRCS).

In cognitive radio, any malicious user can observe communication and emits a false message to block communication due to shared frequency. Software-defined radio (SDR) technology makes jamming attacks very easy since the waveforms are defined in software. A jammer transmits radio signal, making communication reducing signal to noise ratio. Different jammer detection methods are analyzed [20]. Few network parameters are used to detect the jamming attack that can be 1. Packet delivery ratio (PDR), 2. Packet send ratio (PSR), 3. Bad packet ratio (BPR), 4. Signal to noise ratio (SNR). The communication parameter used in SDR like 1. Synchronization indicator, 2. Iteration, 3. Adaptive signal to jammer plus noise ratio (ASNJR) are the new detections for detecting the jamming attack for the software-defined radio hardware.

The Internet of Medical Things (IoMT) is the innovation that enables e-healthcare to make treatments more flexible and convenient. But security and privacy become the central issue of IoMT which is challenging to have highly computational cryptographic to solve because of the following characteristic of the IoMT devices 1. Limited computational capability, 2. Limited memory space, 3. Has energy constraints. A friendly jamming (Fri-jam) scheme is proposed for potential countermeasures to the security of IoMT [21]. This Fri-jam protects the confidential medical data collected by medical sensors from eavesdropping. It is also possible to integrate Fri-jam schemes communication technologies like 1. Beamforming, 2. Simultaneous Wireless Information and Power Transfer (SWIPT), 3. Full duplexity.
A rational attacker made for a specific transmission characteristics waveform has the highest impact on cognitive radio. The software-defined radio platform makes it possible to generate such a target attack. A honeynet-based defense for jamming is presented [22]. The honeynet learns the attacker’s strategy and adapts the preemptive process. It is proved that the proposed mechanism makes CRN successfully avoids jamming attacks and improves the packet delivery ratio.

The primary user emulation (PUE) attack is another type of attack in cognitive radio which disable accessing idle frequency spectrums. An adaptive Bayesian learning automaton algorithm (Multichannel Bayesian Learning Automata (MBLA)) is proposed to defend PUE [23]. The algorithm MBLA learning in non-stationary environments and selecting the optimal frequency channel. An uncoordinated frequency hopping (UFH) is utilized for sending data on different channels.

Vehicular ad-hoc networks (VANET) are deployed with safety-critical applications to be protected from jamming attacks. A jamming detection approach for wireless vehicular networks using unsupervised machine learning is presented [23]. The variations of the relative speed between the jammer and the receiver are used as a parameter for detecting the jamming attack. The mechanism can differentiate intentional and unintentional jamming with their unique characteristics.

Jammer localization finding an essential one because it helps for jamming-avoidance routing. It is easy to detect location in the presence of usage of omnidirectional antennas. It is challenging for directional jammers to use directional antennas. An Adaptive Jammer Localization Algorithm (AILA) method, which estimates the jammer’s antenna type and selects the appropriate algorithm based on the type of antenna, is presented [24]. Centroid Localization (CL), Virtual Force Iteration Localization (VFIL) algorithms are used for omnidirectional antennas. An Improved Gravitational Search Algorithm (IGSA) is used for a directional antenna. The comparison of the related work with the proposed mechanism is presented in Table 1.

From Table 1, it can be observed that the proposed mechanism is superior that the literature in the aspect of our proposed mechanism able to detect random discrete jamming, convergence time is less, not requires pre-sharing of access information and statistical information of primary users’ activities and jammer not required. Limitation of the proposed mechanism works well only if the jammer’s signal is strong enough comparing to the primary user is not a big issue since the jammer always attacks with a strong signal to destroy the legislative user signal.

Reactive jamming is most challenging to detect. Jammer detection and localization in a WiFi network are presented [25]. The mechanism uses the fact that the jammer increases the interference range and keeps the access point busy with more time. Recently deep learning models are used for wireless communication [26, 27] for signal classification and attack detection, which inspired to use the deep learning model to detect jamming attack detection in CIoT.

Thus, this article focuses on the detection of jamming attack in IoT cognitive radio network with the following contribution.

1. Simulation of Discrete random jamming signal using Phased Barrage Jammer model.
2. Discrete random jamming detection is challenging, which is formulated as a deep learning classification/ anomaly detection problem and solved by an autoencoder.
3. Time instant of jamming is detected, which will help to fine-tune spectrum detection algorithm and improve the spectral efficiency
| Literature | Problem solved | Methodology | Research gap |
|------------|----------------|-------------|--------------|
| Salameh et al. [14] | Channel assignment process in the presence of both proactive and reactive jamming | A probabilistic-based channel assignment mechanism | The statistical information of primary users’ activities, fading conditions and jamming attacks over idle channels are required |
| Xu et al. [15] | Maximize the rate of successful transmission | Markov decision process and deep reinforcement learning | Reinforcement learning takes some time to converge |
| Chao et al. [16] | Anti-jamming mechanism | Channel hopping | Requires pre-sharing of access information and not taking care of the traffic loads’ variations |
| Karunambiga et al. [19] | Anti-jamming mechanism- increase jamming resistance | Dynamic channel switching in Cognitive WiFi (CWF) network, LU-MAC protocol for dynamically switching between licensed and unlicensed bands | Increases jamming resistance, not fully solve the problem. Unlicensed bands able only provide less data rate |
| Bhunia et al. [22] | Defense for jamming and improves the packet delivery ratio | A honeynet-based defense | Honeynet need more time for learns the attacker’s strategy |
| Proposed work | 1. Detect discrete ransom jamming; able to give time instant of attack 2. Improves spectral efficiency 3. Convergence time is less 4. Not requires pre-sharing of access information 5. The statistical information of primary users’ activities and jammer not required | Autoencoder | Work well only if the jammer’s signal is strong enough comparing to primary user |
2 System Model

The system model used for the simulation of the proposed method is given in Fig. 1. The system model in figure one consists of a primary user network of cellular communication with licensed spectrum access. It is assumed that there is n number of primary users in the primary user network. The secondary network called cognitive radio IoT network has m number of nodes that sense data from the environment and communicate to the data collection center. This CRIoT network uses dynamic spectrum access using cognitive radio, which enables sharing the primary user spectrum which the primary user does not use. It is assumed that there is a spectrum sensing algorithm running on each CRIoT node that can detect the spectrum hole by using an energy detector. Since the energy detector is a low complex algorithm for the spectrum sensing, which is utilized in the CRIoT because those nodes are low cost and less computational capable. There is a jammer in the close vicinity of the CRIoT network that will be injecting random jamming signal to the CRIoT nodes in the same frequency set of the primary network with the aim to disable the CRIoT node not to utilize the spectrum hole point spectrums properly. Since the energy detector algorithm detects the spectrum hole through the received signal energy on a particular frequency and the jamming signal injects a high energy signal in the primary user frequency, the CRIoT node will wrongly interpret the jammer signal as the direct signal. It will not use the primary user frequency even though they are free without utilizing it at the given instant of the time. It is also assumed that the jammer signal impact on the primary user network is minimum since it is away from the jammer and the transmit power of those nodes is higher than the jammer signal.

The received signal at cognitive radio when primary alone user present

\[ y_{cp} = h_p x_p + n \]  

(1)

where \( h_p \) is the channel coefficient between the primary user and cognitive radio, \( x_p \) is the primary user transmit symbol, \( n \) is the additive white gaussian noise.

The received signal at cognitive radio when no primary user and jammer present

![Fig. 1 System model of CRoT network with jamming attack](image-url)
$y_{cp} = n$  

The received signal at cognitive radio when primary user and jammer present

$$y_{cp} = h_p x_p + h_J x_J + n$$  \hspace{1cm} (3)

where $h_J$ is the channel coefficient between the cognitive radio and jammer, $x_J$ is the jammer transmit symbol, $n$ is the additive white gaussian noise.

The received signal power of the primary user at cognitive radio is

$$P_{PRX} = \frac{P_p G_p c^2 G_c}{4\pi^2 f^2 R_{pc}^2}$$  \hspace{1cm} (4)

The received signal power of the jammer at cognitive radio is

$$P_{JRX} = \frac{P_J G_J c^2 G_c}{4\pi^2 f^2 R_{jc}^2}$$  \hspace{1cm} (5)

The impact of the jamming effect is based on the jamming to signal power ratio

$$JSPR = \frac{P_J G_J R_{pc}^2}{P_p G_p R_{jc}^2}$$  \hspace{1cm} (6)

where $P_J$ and $P_p$ are the transmit power of the jammer and primary user, respectively. $G_J$ and $G_p$ are the antenna gain of jammer and primary user $R_{pc}$ and $R_{jc}$ are the range or distance between the primary user and cognitive radio, the distance between the jammer and cognitive radio.

If the $JSPR$ is more significant than one, it has an impact on the primary user detection. Whenever $P_{JRX} \geq$ threshold, then the jammer signal will be treated as the primary user signal, then the primary user frequency occupied decision will be made wrongly. This wrong detection makes the cognitive user not use the primary user spectrum hole and reduces the spectrum utilization.

Figure 2 shows the mechanism of the jammer detection. The data obtained from the receiver is pre-processed by shaping the data in the 2D form, splitting the data into training a test set, windowing to have 256 samples at a time for the training. The formatted data

![Fig. 2 Block diagram of jammer detection using an autoencoder](image-url)
can be applied as raw data and as well as feature extraction. First, the Fourier transform is applied to the raw data then the amplitude-frequency spectrum is taken as a feature set used for training the autoencoder.

The feature set of the data for classification is calculated by first computing N point DFT for the received signal of the cognitive radio as follow

\[ Y(k) = \frac{N-1}{\sum_{n=0}^{N-1}} y_{cp}(n)e^{-i2\pi kn/N} \]

where \( Y(k) \) is the N point DFT of the received cognitive radio signal \( y_{cp}(n) \). The amplitude spectrum of \( Y(k) \) is taken as a feature set which is computed by taking the absolute value of \( Y(k) \).

Feature set \( f = ||y(k)|| \).

Autoencoder is configured as an anomaly detector to find the jamming attack. The outcome of the autoencoder is connected to the spectrum management unit. The spectrum management unit has the three essential processes required for utilize the unused spectrum 1. Spectrum sensing for find the free channel, 2. Free channel selection, 3. Channel handoff if the primary user occupies the channel currently being used. For all three key processes, the jamming attack detection outcome is a must to make the right decision.

Figure 3 shows autoencoder architecture. Table 2 shows the autoencoder architecture layer information. It can be observed that the encoder part has one 1D convolutional layer with 32 filters with a kernel size of 7, padding mechanism of the same and strides size 2 with activation function of relu. Followed by a drop-out layer is provided with a drop-out rate of 0.2. a second layer, another convolution layer, is provided with 16 filters with a kernel size of 7, padding mechanism of same, and strides size 2 with activation function

| Layer            | Filters | Kernel size | Padding | Strides | Activation |
|------------------|---------|-------------|---------|---------|------------|
| Input layers-Conv1D | 32      | 7           | Same    | 2       | Relu       |
| Dropout (rate = 0.2) |         |             |         |         |            |
| Conv1D           | 16      | 7           | Same    | 2       | Relu       |
| Conv1DTranspose  | 16      | 7           | Same    | 2       | Relu       |
| Dropout (rate = 0.2) |         |             |         |         |            |
| Conv1DTranspose  | 32      | 7           | Same    | 2       | Relu       |
| Conv1DTranspose  | 1       | 7           | Same    | 2       | Relu       |
of relu. Then the decoder section, the first layer, has the weight value of the transposed of the last encoder layer with the same configuration of the last layer in the encoder section, i.e., with 16 filters with kernel size 7, padding mechanism of same and strides size 2 with activation function of relu. Then a dropout layer is used with the rate of drop out 0.2. the second layer in the decoding section has a transpose weight value of the first layer of the encoder with the same configuration parameter of it, i.e., 1D convolutional layer with 32 filters with a kernel size of 7, padding mechanism of same and strides size 2 with activation function of relu. The decoder’s last layer has only one filter with kernel size 7, padding mechanism of same and strides size 2 with activation function of relu.

The algorithm of jammer detection using autoencoder is given below.

| Autoencoder anomaly detection based jammer detection for cognitive radio |
|---------------------------------------------------------------|
| **Training phase**:                                           |
| **Step 1**: configure the primary user to generate \( x_p \) QPSK signal at ransom instant of time on frequency \( f \) and propagate signal on the channel \( h_p \) with the transmit power of \( P_p \). |
| **Step 2**: Receive the signal at cognitive radio \( y_{cp} \) |
| \( y_{cp} = h_p x_p + n \) |
| **Step 3**: Split the data as training, validation and test set |
| **Step 4**: apply a windowing technique of rectangular window to generate 256 samples at a time to feed and train the autoencoder |
| **Step 5**: for training sample window = 1 to \( M \) propagate training samples and train the autoencoder to reproduce the exact same input signal and record the range of mean absolute difference values by computing as |
| a) compute encoder output \( o_{en} = y_{cp} W^T \) |
| b) compute decoder output \( Y_{pdec} = o_{en} V^T \) |
| c) calculate mean absolute error \( MAE_{train} = \frac{1}{M} \sum_{win=1}^{M} |Y_{pdec} - y_{cp}| \) |

**Jammer detection phase**:

**Step 1**: configure the jammer to generate a Phased Barrage Jammer signal \( x_j \) at ransom instant of time on frequency \( f \) and propagate signal on the channel \( h_j \) |

**Step 2**: Receive the signal at cognitive radio \( y_{cp} \) |
| \( y_{cp} = h_p x_p + h_j x_j + n \) |
**Step 3**: apply a windowing technique of rectangular window to generate 256 samples at a time to feed the autoencoder |
**Step 4**: compute mean absolute difference values as |
| a) compute encoder output \( o_{enj} = y_{cp} W^T \) |
| b) compute decoder output \( Y_{pdec} = o_{enj} V^T \) |
| c) calculate mean absolute error \( MAE_{test} = \frac{1}{M} \sum_{win=1}^{M} |Y_{pdec} - y_{cp}| \) |
**Step 5**: for \( i = 1 \) to \( M \) |
| a) compute the MAE difference between the training values \( MAE_{train} \) and \( MAE_{test} \) |
| \( jam_{score}(i) = |MAE_{train}(i) - MAE_{test}(i)| \) |
| b) if \( jam_{score}(i) > jam_{threshold} \) then jammer signal presents at the \( i^{th} \) instant of the window; else no jamming signal at \( i^{th} \) moment of window |

The step-by-step process behind the detection of jamming attacks is proved in the above algorithm named Autoencoder anomaly detection based jammer detection for cognitive radio.
The key principle mechanisms applied in the algorithm are as follows 1. the typical jammer signal is generated and used for the training of the autoencoder. 2. The autoencoder is operated as an anomaly detector. 3. After training in the detection mode operation of the autoencoder, the mean absolute error value is tracked. 4. If the mean absolute error value is greater than the value observed in the training phase, then it is declared that an anomaly is detected, i.e., a jamming attack is detected; else, it is declared there is no jamming attack.

3 Result and discussion

The entire proposed system is simulated in MATLAB and python. The MATLAB software is used for signal generation, i.e., data generation with PU, Jammer and cognitive radio. Then the generated data is used for the jammer detection by using an autoencoder as an anomaly detector.

Table 3 shows the simulation set of parameters used for the data generation in MATLAB. The Phased Barrage Jammer is used with ERP=10 with 400 samples per frame generation. The simulation is repeated ten times to realize 4000 samples of data. The PU signal is generated with QPSK modulation and ERP=10. The signal of PU is propagated through the transmit filter of Raised Cosine with Roll off Factor=0.3 and Output Samples Per Symbol=2. This PU signal and jammer signal are combinedly received at CR receiver and stored in the autoencoder training file. The autoencoder with the layer information given in Table 2 is trained using this data-generated file.

Figure 4 shows the received signal at a cognitive radio node with PU signal components and jammer signal components. The signal level with amplitude within 20 is the PU component. The signal level with more than 20 up to 130 is the jammer component. From Fig. 4, it is also evident the jammer signal is added randomly with the PU signal.

Figure 5 shows the QPSK modulated PU signal. It takes an amplitude of 25 units as maximum amplitude. This signal is generated at random instant of time such that we can observe some spectrum hole in the graph, the white line instants are the spectrum hole points.

Figure 6 shows the jammer signal alone. It takes from −50 to +120. This jammer signal is also generated randomly such that the detection of the random attack is difficult. Moreover, the random time instant of the jammer injection is recorded for the comparison purpose after finding the autoencoder’s jammer injection instant.

| S. no | Parameter                          | Value                                |
|-------|------------------------------------|--------------------------------------|
| 1     | Number of PU node                  | 1                                    |
| 2     | Number of CR node                  | 1                                    |
| 3     | Number of jammer node              | 1                                    |
| 4     | Jammer type                        | Phased barrage jammer                |
| 5     | Phased barrage jammer parameters   | ERP = 10, SamplesPerFrame = 400      |
| 6     | PU signal parameters               | QPSK modulation, ERP = 10            |
| 7     | Transmit filter for PU             | Raised cosine transmit filter, roll-off factor = 0.3 |
|       |                                    | Output samples per symbol = 2        |
Tracking the deep learning model’s training and validation losses will give an idea of whether the model is overfitted or under fitted. Moreover, it also gives an idea of whether the furthermore training is required or more training data is required. Figure 7 gives the training and validation losses of the designed autoencoder. The figure shows that the model is not overfitting or underfit, but there is a scope of improving the accuracy by a new data set or feature set.

The mean absolute error is used as a loss function in the autoencoder and the model is trained to minimize this loss value. Figure 8 shows the histogram of the training time mean absolute error value. It is evident from the graph that the error is taken from 0.105 to 0.140. This small variation indicates that the data set does not have much variance to make more weight changes. It is also evident that the error value of around 0.118 is taken much of the time.
Fig. 6 Jammer signal alone

Fig. 7 Training and validation losses

Fig. 8 Histogram of the training mean absolute error
The autoencoder is designed and trained to encode the PU signal in the encoder part and reprocess the PU signal exactly at the output of the decoder. In order to access the reproducing capability of the autoencoder after training the PU signal, the decoded PU signal and the original input PU signal which is used to train is plotted in a single graph as shown in Fig. 9. In Fig. 9, blue color signal indicate the input original PU signal which fed during the training phase and the orange color signal is the reconstructed signal at the autoencoder output by taking the compressed component of the signal from the outcome of the encoder part.

The error performance of the deep learning model for the test data set is required to access the model’s generalization capability. Figure 10 shows the test error performance of the autoencoder. The figure shows that for the then-new unseen data set during the test phase, the model mostly provides around 0.13, i.e., 13% error.

Detecting the jammer signal’s presence at the receiver of CR radio and the time when jammer enters into the system is essential. Under the proposed autoencoder, the detection of the jammer signal and the time instant on which it is detected is given. Figure 11 shows such detection of the jammer signal. The red color mark in the figure shows the jammer signal component detected by the autoencoder. The axis value shows the timestamp of the detected jammer component. From the figure, it is evident that the jammer signal component is detected after the sampling instant 256 onwards. It happens because the first
window of size data 256 needs to be processed by the autoencoder, then only it can recognize that the samples are from the jammer. It is the limitation of the proposed mechanism. The accuracy of detection of the jammer is evaluated and it is found around 89%.

In order to evaluate the performance of the proposed jamming detection mechanism, the spectral efficiency of the CRIoT network node after jammer detection with various jamming timing is calculated and plotted in Fig. 12 against the spectral efficiency of the same node with a jammer. The simulation parameter given in Table 3 is used for PU and jammer signal generation. QPSK modulation with an ERP value of 5 is used to generate the cognitive radio signal. The spectral efficiency is calculated over 10 s of transmission time. Figure 12 is drawn by considering the 10 s transmission time, and the performance is analyzed. Figure 12 proves that the proposed mechanism maintains spectral efficiency with negligible reduction comparing to that of jammer. Around 2 bits/Hz/s, spectral efficiency
gain is achieved by the proposed jammer detection algorithm for the 5 s jamming time comparing to that of the spectral efficiency with a jammer.

4 Conclusion

Jamming attack in cognitive IoT network disable the dynamic spectrum sharing, the detection of jamming attack predominantly random jamming is challenging to detect, which is solved using autoencoder deep learning model. The autoencoder is configured as an anomaly detector, and the jamming attack is detected using that. The proposed mechanism able to detect the jammer with the time instant of jamming. The accuracy of the prediction of the jammer is achieved with 89% accuracy. The spectral efficiency of the cognitive radio with jammer detection and without jammer detection is evaluated. It is found that for 5 s jamming, the spectral efficiency is improved as double while considering the 10 s transmission time. However, the proposed mechanism limitations are 1. Can detect more accurately only if the jammer’s signal power is strong than the primary user signal power. 2. The system is not incorporated the mitigating mechanism. These limitations will be addressed in future work. A robust system that will work for all signal power and automatically apply the appropriate mitigation mechanism will be incorporated in future work.

Author’s contributions No part of this manuscript is published in any other journal and this work is truly a combined work of all the authors.

Funding This research received no external funding.

Data Availability The authors declare that no data or material was taken illegally.

Code availability The authors declare that no exact code has been copied to carry out the research

Declarations

Conflict of interest The author declares no conflict of interest.

References

1. Moayedian, N. S., Salehi, S., & Khabbazian, M. (2020). Fair resource allocation in cooperative cognitive radio IoT networks. IEEE Access, 8, 191067–191079.
2. Ansere, J. A., Han, G., Wang, H., Choi, C., & Wu, C. (2019). A reliable energy efficient dynamic spectrum sensing for cognitive radio IoT networks. IEEE Internet of Things Journal, 6(4), 6748–6759.
3. Arat, F. & Demirci, S. (2019). Analysis of spectrum aware routing algorithms in CR based IoT devices. In: 2019 4th international conference on computer science and engineering (UBMK), pp. 751–756. IEEE.
4. Awin, F. A., Alginahi, Y. M., Abdel-Raheem, E., & Tepe, K. (2019). Technical issues on cognitive radio-based Internet of Things systems: A survey. IEEE Access, 7, 97887–97908.
5. Ponnusamy, V., & Malarvihhi, S. (2017). Hardware impairment detection and prewhitening on MIMO precoder for spectrum sharing. Wireless Personal Communications, 96(1), 1557–1576.
6. Vijayakumar, P., & Malarvihhi, S. (2017). Green spectrum sharing: Genetic algorithm based SDR implementation. Wireless Personal Communications, 94(4), 2303–2324.
7. Ahmed, R., Chen, Y., Hassan, B., & Du, L. (2021). CR-IoTNet: machine learning based joint spectrum sensing and allocation for cognitive radio enabled IoT cellular networks. Ad Hoc Networks, 112, 102390.
8. Liu, M., Song, T., & Gui, G. (2018). Deep cognitive perspective: Resource allocation for NOMA-based heterogeneous IoT with imperfect SIC. *IEEE Internet of Things Journal, 6*(2), 2885–2894.
9. Salahdine, F., & Kaabouch, N. (2020). Security threats, detection, and countermeasures for physical layer in cognitive radio networks: A survey. *Physical Communication, 39*, 101001.
10. Lin, S.-C., Wen, C.-Y., & Sethares, W. A. (2017). Two-tier device-based authentication protocol against PUEA attacks for IoT applications. *IEEE Transactions on Signal and Information Processing over Networks, 1*(4), 33–47.
11. Bhattacharjee, S., Sengupta, S., & Chatterjee, M. (2013). Vulnerabilities in cognitive radio networks: A survey. *Computer Communications, 36*(13), 1387–1398.
12. Haldorai, A., & Ramu, A. (2020). Security and channel noise management in cognitive radio networks. *Computers and Electrical Engineering, 87*, 106784.
13. Ponnusamy, V., Kottursamy, K., Karthick, T., Mukeshkrishnan, M. B., Malathi, D., & Ahanger, T. A. (2020). Primary user emulsion attack mitigation using neural network. *Computers and Electrical Engineering, 88*, 106849.
14. Salameh, H. A. B., Almajali, S., Ayyash, M., & Elgala, H. (2018). Spectrum assignment in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks. *IEEE Internet Things Journal, 5*(3), 1904–1913.
15. Xu, J., Lou, H., Zhang, W., & Sang, G. (2020). An intelligent anti-jamming scheme for cognitive radio based on deep reinforcement learning. *IEEE Access, 8*, 202563–202572.
16. Chao, C.-M., & Lee, W.-C. (2021). Load-aware anti-jamming channel hopping design for cognitive radio networks. *Computer Networks, 184*, 107681.
17. Salameh, H. B., Otoum, S., Aloqaily, M., Derbas, R., Al Ridhawi, I., & Jararweh, Y. (2020). Intelligent jamming-aware routing in multi-hop IoT-based opportunistic cognitive radio networks. *Ad Hoc Networks, 98*, 102035.
18. Ge, J., Xie, J., & Wang, B. (2021). A cognitive active anti-jamming method based on frequency diverse array radar phase center. *Digital Signal Processing, 109*, 102915.
19. Karunambiga, K., & Sundararambal, M. (2015). LU-MAC: licensed and unlicensed MAC protocol for cognitive WiFi network with jamming-resistant. *Procedia Computer Science, 47*, 424–433.
20. Bhujani, R., & Joshi, R. (2016). An integrated approach for jammer detection using software defined radio. *Procedia Computer Science, 79*, 809–816.
21. Li, X., Dai, H.-N., Wang, Q., Imran, M., Li, D., & Imran, M. A. (2020). Securing internet of medical things with friendly-jamming schemes. *Computer Communications, 160*, 431–442.
22. Bhunia, S., Miles, E., Sengupta, S., & Vázquez-Abad, F. (2018). CR-HoneyNet: A cognitive radio learning and decoy-based sustenance mechanism to avoid intelligent jammer. *IEEE Transactions on Cognitive Communications and Networking, 4*(3), 567–581.
23. Mahmoudi, M., Faez, K., & Ghasemi, A. (2020). Defense against primary user emulation attackers based on adaptive Bayesian learning automata in cognitive radio networks. *Ad Hoc Networks, 102*, 102147.
24. Wang, T., Wei, X., Fan, J., & Liang, T. (2018). Adaptive jammer localization in wireless networks. *Computer Networks, 141*, 17–30.
25. Cai, Y., Pelechrinis, K., Wang, X., Krishnamurthy, P., & Mo, Y. (2013). Joint reactive jammer detection and localization in an enterprise WiFi network. *Computer Networks, 57*(18), 3799–3811.
26. Clement, J. C., Indira, N., Vijayakumar, P., & Nandakumar, R. (2021). Deep learning-based modulation classification for 5G and beyond wireless systems. *Peer-to-Peer Networking and Applications, 14*(1), 319–332.
27. Erpek, T., Sagduyu, Y. E., & Shi, Y. (2018). Deep learning for launching and mitigating wireless jamming attacks. *IEEE Transactions on Cognitive Communications and Networking, 5*(1), 2–14.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
V. Nallarasan is a Part time (internal) research scholar in School of Computing, SRM Institute of Science and Technology, Tamilnadu, India. He is a Member of IEEE. His research interests includes internet of things, wireless networks, network/cyber security, machine learning.

Kottilingam Kottursamy is an Associate Professor in School of Computing, SRM Institute of Science and Technology, Tamilnadu, India. He has completed his PhD from Anna University, India. He is one of the Co-Principal Investigators of IITK-SPARC Project, in collaboration with UC Davis, USA. He is a former Member of NGNLab, Anna University, Chennai, India. His research interest includes Artificial Intelligence and Machine Learning, Next Generation Networks, Bio-Statistics and UAV Swarm Networks.