Signal Denoising and Detection for Uplink in LoRa Networks based on Bayesian-optimized Deep Neural Networks

Angesom Ataklity Tesfay, Sofiane Kharbech, Eric Pierre Simon, Laurent Clavier

Signal Denoising and Detection for Uplink in LoRa Networks based on Bayesian-optimized Deep Neural Networks

Angesom Ataklity Tesfay, Sofiane Kharbech, Eric Pierre Simon, and Laurent Clavier

***Abstract*—Long-range and low-power communications are suitable technologies for the Internet of things networks. The long-range implies a very low signal-to-noise ratio at the receiver. In addition, low power consumption requires reduced signaling, hence the use of less complex protocols, such as ALOHA, so reduced communication coordination. Therefore, the increase of objects using this technology will automatically lead to an increase in interference. In this paper, we propose a detector for Long Range (LoRa) networks based on an *autoencoder* for denoising and dealing with the interference, followed by a *convolutional neural network* for symbol detection. Simulation results demonstrate that the proposed approach outperforms both the convolutional neural network-based detector and the classical LoRa detector in the presence of interference from other LoRa users. The proposed detector shows around 3 dB gain for a target Symbol Error Rate (SER) of 10^{-4}.**

***Index Terms*—LoRa, IoT, deep learning, neural networks, autoencoder, Bayesian optimization.**

I. Introduction

Despite the high expectation of large-scale deployment of the Internet of Things (IoT) in the last decade, the number of devices on the field is far from what was expected. Indeed, low energy, low cost, and reliability are challenging to handle simultaneously. Low-Power Wide-Area Networks (LPWANs) address this issue and present an asymmetry that we can exploit. If devices are low power and low cost, access points manage a large number of devices; they can be more expensive and plugged in, able to handle a higher computational burden. It allows one to implement more complex decoding algorithms.

Additionally, the broad coverage of a single cell and the low power allocated to transmission make IoT networks operate mainly in low signal-to-noise ratio (SNR) values. LoRa (Long Range) uses a chirp spread spectrum (CSS) modulation scheme and works below the noise level, i.e., SNR < 0 dB. Thus, denoising techniques are worth investigating to extract useful information from the noisy signal. Besides, the ALOHA protocol used in LoRaWAN generates collisions and packet losses when two or more devices transmit using the same spreading factor (SF) and frequency band at the same time [1].

M. Tesfay [1,2], S. Kharbech[2,3], E.-P. Simon[1], and L. Clavier[1,2] are with [1]Univ. Lille, CNRS, Centrale Lille, UMR 8520 - IEMN, 59000 Lille, France (e-mail: [1]firstname.name@univ-lille.fr), [2]IMT Nord Europe, Centre for Digital Systems, 59000 Lille, France (email: [2]firstname.name@imt-nord-europe.fr), and [3]Univ. Tunis El Manar, Laboratory Sys'Com-ENIT (LR-99-ES21), Tunis 1002, Tunisia (email: [3]firstname.name@enit.utm.tn).
This work has been supported by IRCICA USR CNRS 3380.

In this paper, we propose a deep learning-based detector structure for LoRa symbol detection in the uplink that copes with challenging conditions at the receiver. The structure is based on a denoising autoencoder [2], [3] and a convolutional neural network. The contributions are (i) the implementation of an autoencoder (AE) for denoising LoRa signals as well as coping with the collisions corresponding to interference coming from other LoRa transmitters using the same SF and frequency band, (ii) a convolutional neural network (CNN) to detect the symbols, and (iii) the set up of a Bayesian-optimized complete decoding scheme.

In previous works, few improvements to the original LoRa detector can be found. If optimal in Gaussian noise, it does not handle interference properly. Some specific works have been proposed to evaluate or improve the capture effect [4], but most of the time with a single interferer and without modifying the receiver detection method. In [5], a convolutional neural network-based LoRa demodulator is proposed. This is achieved by considering time-domain LoRa symbols with impairments, including additive white Gaussian noise, carrier frequency, and time offset. The results show an improvement when using deep learning over typical non-coherent detection. However, in this work, the impact of interference from the other LoRa user is not considered. In [6], a neural network-based approach is proposed, but without preprocessing step. In this paper, we consider as input for the proposed detector the time-frequency representation of the received signal, i.e., its spectrogram, to efficiently take advantage of autoencoders' denoising capabilities. In [6], the input of the CNN is a two-dimensional image of a line plot. However, autoencoders are not suited for denoising such kind of input since the noise in [6] corrupts the shape of the line plot rather than acting stochastically on the whole image. By using the spectrogram as input, the noise is spread all over the image, resulting in a noisy image for which the autoencoder is able to perform noise removal. Besides denoising, the autoencoder is also able to deal with interference from other LoRa users if present. If autoencoders have been used previously to denoise signals in receivers [7]–[9], as far as we know, they were not employed yet for LoRa systems or signal detection.

Simulation results show that our proposed detector outperforms the classical LoRa detector [10], the coherent LoRa detector in [11], and the deep learning-based detectors of [5] and [6] in the presence of interference.

II. System Model

Each symbol in LoRa carries SF bits, resulting in $M = 2^{\text{SF}}$ different symbols. SF ranges from 7 to 12. The symbol duration is $T_s = MT$, where $T = \frac{1}{B}$ and B is the signal bandwidth [12]. Each LoRa symbol is represented by a chirp that contains a linear frequency change over the time interval T_s. It is generated from a raw chirp $s(t)$ which has an instantaneous frequency of $\frac{B}{T_s}t$, resulting in the base-band expression:

$$s(t) = \exp\left(2\pi\jmath\frac{B}{2T_s}t^2\right) \qquad t \in \left[-\frac{T_s}{2}, \frac{T_s}{2}\right]. \tag{1}$$

Let Q be the total number of LoRa symbols in a packet. User j transmits at time qT_s $(q = 0, \cdots, Q-1)$ the symbol $m_q^{(j)} \in \{0, \cdots, M-1\}$. Information is encoded by cyclic shifting the raw chirp by $\delta_q^{(j)} = m_q^{(j)}T$, as shown in Fig. 1. We have:

$$s_q^{(j)}(t) = \begin{cases} \exp\left(2\pi\jmath\left[\frac{Bt^2}{2T_s} + \frac{m_q^{(j)}}{T_s}t\right]\right), t \in \left[-\frac{T_s}{2}, \frac{T_s}{2} - \delta_q^{(j)}\right[, \\ \exp\left(2\pi\jmath\left[\frac{Bt^2}{2T_s} + \left(\frac{m_q^{(j)}}{T_s} - B\right)t\right]\right), t \in \left[\frac{T_s}{2} - \delta_q^{(j)}, \frac{T_s}{2}\right]. \end{cases}$$

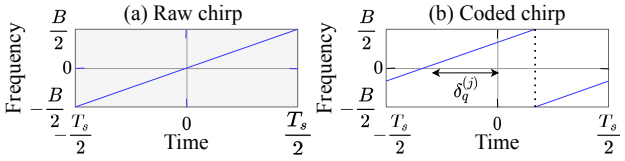


Fig. 1: (a) Raw chirp (b) Coded chirp associated with $m_q^{(j)}$.

User j transmits $s^{(j)}(t) = \sum_{q=0}^{Q-1} s_q^{(j)}(t - qT_s)$. A preamble composed of raw chirps is added to form the LoRa packet.

We assume that the gateway is at the center of a circle of radius r_{max} with a user-free guard-zone of radius r_{min}. The number N_i of interfering users with the same SF is randomly chosen from a Poisson distribution with parameter λ. Their 2D coordinates are uniformly distributed in the ring. The communication between nodes is uncoordinated and asynchronous. Synchronization at the receiver is based on the correlation between the received signal and the preamble, and we assume it is perfect. The received signal associated with symbol q of user j sampled at $t = nT$, $n = -\frac{M}{2}, \cdots, \frac{M}{2} - 1$ is:

$$r_q[n] = h^{(j)}\ s_q^{(j)}[n] + \sum_{i \in \mathcal{I}} h^{(i)} s_{q,\text{interf}}^{(j,i)}[n] + w_q[n], \tag{2}$$

where $s_q^{(j)}[n] = s_q^{(j)}(nT)$, $\mathcal{I}$ is the set of interfering users ($|\mathcal{I}| = N_i$), and $w_q[n] \sim \mathcal{CN}(0, \sigma^2)$ is a complex Gaussian noise. The channel coefficients of user i is denoted $h^{(i)}$.

III. The Classical Detector

The classical detection for LoRa utilizes the non-coherent Frequency Shift Keying (FSK) detection technique [10]. First multiplies the samples of the received signal by the conjugate of the raw chirp, yielding $y_q[n] = r_q[n]s^*[n]$. A Fast Fourier Transform (FFT) is then applied:

$$Y_q[k] = \sum_{n=-M/2}^{M/2} y_q[n]\ e^{-2\pi\jmath\frac{nk}{M}},\ k = 0, \ldots, M-1. \tag{3}$$

The symbol $m_q^{(j)}$ is estimated by taking the frequency index where the modulus of (3) is maximum. It cannot handle an interferer with higher power.

IV. Proposed Deep Learning-based Detector

The proposed deep learning-based detector architecture is illustrated in Fig. 2 and decomposed in two steps: an autoencoder and a CNN based detector.

A. Input signal

Instead of the signal itself, the input of our denoiser is a grayscale spectrogram of the received signal (cf. Fig. 6a and 6b). The spectrogram is obtained as follows: a window (Hamming) of the signal is taken, and the Fourier Transform of this window is calculated. The magnitude of the FFT will represent one column of the spectrogram. Then the window is shifted, and a second line is created using the same process. The length of the window (and of the FFT) and the length of the shifts are parameters to be chosen that will define the size of the spectrogram. The spectrogram is defined by the length l of the window used for calculating the Short-Time Fourier Transform (we use a Hamming window), m the number of overlapping samples between two consecutive windows, and k the number of samples of the Fourier transform. Having fixed l, the minimum resolution would be given by $m = 0$ and $k = l$.

To improve the receiver performance, we over-sample the spectrogram both in time and frequency. For SF = 7 in result section V-A, we use a 128×128 image size obtained with $l = 32$, $m = l - 1$ and $k = 128$. It is to be noted that we center the first window on the first signal value $r_q[-M/2]$, which requires to add $l/2$ samples. Taking benefit of the cyclic property of $r_q[n]$ we add the $l - 1$ last samples, i.e. $r_q[i] = r_q[i + 2^{\text{SF}}]$ for $i = -M/2 - l/2, \cdots, -M/2 - 1$. Similarly, we add the first $l/2$ samples at the end of the signal so that the last window is centered on the last sample.

However, it is not necessary to take a $N \times N$ window with $N = 2^{\text{SF}}$. We will see in the complexity analysis (section V-B) that fixing an image size with $N = 128$ is sufficient. The image does not even need to be square, but the best compromise between accuracy and complexity is an open question.

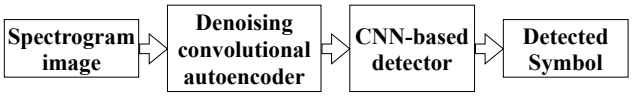


Fig. 2: Architecture of the detector.

B. Denoising step: The Convolutional Autoencoder

An autoencoder is a neural network with two parts: encoding and decoding. The encoding part aims at getting a compressed low dimensional representation of the input; the decoding part reconstructs the input. The autoencoder is trained with noisy input and a noise-free target results in a model that learns the essential features and denoises the

input image. As shown in Fig. 3, the encoder is a series of convolutional layers, each followed by a downsampling (pooling) layer. The decoder is a series of convolutional layers, each followed by an upsampling layer. As the decoder must do the reverse operation, the whole architecture is symmetrical [13]. The number of convolutional layers, the number of filters and their sizes for each layer are determined through Bayesian optimization [14]. In short, Bayesian optimization is a typical derivative-free optimization method used for black-box objective functions. A black-box function is a function in which we can only observe its output based on a given input, i.e., with no information about what is inside. Within the context of our paper, the input is the set of hyperparameters of the considered machine learning (ML) model (based on the approach, it can be the CNN or the Convolutional Autoencoder), and the output is the ML model accuracy. Thus, the black-box is the function that maps a given set of hyperparameters (i.e., the input) to the corresponding ML model accuracy (i.e., the output).
Bayesian optimization relies on the following steps. (i) Evaluate the objective function based on the (initial) input samples. (ii) Based on the samples and their corresponding outputs, perform a Gaussian Process Regression (GPR). In fact, the result is a statistical model of the black-box objective function (in other words., a substitute or surrogate). In this step, the Bayes rule is involved. (iii) Calculate the acquisition function, a metric function to decide which set of hyperparameters is a potential lead to the optimum, so it will be employed for the next searching iteration. (iv) Identify the next to-evaluate input, then iterate (starting from step (ii) to update the GPR model based on the enriched observations) until achieving some stopping criteria.

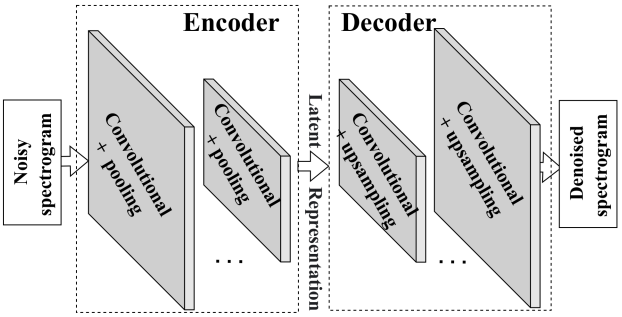


Fig. 3: Convolutional Autoencoder Network architecture.

C. Detection step: The Convolutional Neural Network

As shown in Fig. 4, the CNN architecture consists of a series of convolutional layers, each followed by a pooling layer and a sequence of fully connected layers. The number of convolutional and fully connected layers, the number of filters and their sizes, and the size of each fully connected layer are determined through Bayesian optimization. The activation function is the ReLU function, and batch normalization is applied. We use the softmax function for the output classification layer. The output of the overall detection system is the index of the transmitted symbol.

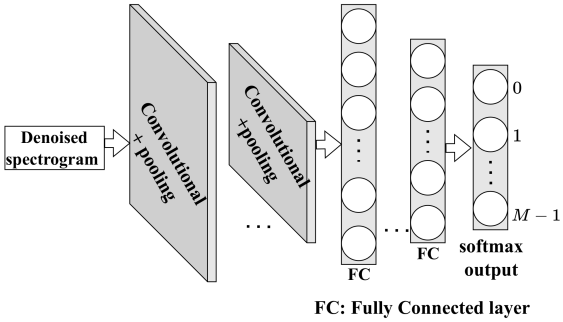


Fig. 4: Convolutional Neural Network architecture.

The adaptive moment estimation (ADAM) is used to optimize the AE and CNN structures' learnable parameters, such as the weights and biases.

V. Simulation Results

The considered LoRa transmission setting is bandwidth $B = 250$ kHz, SF $= 7$ resulting in $M = 128$. The values for r_{max} and r_{min} are 1 km and 20 m, respectively. The signal amplitude decays with distance d according to $d^{-\gamma/2}$, where γ is the path loss exponent, $\gamma = 4$ is considered here. The number of interfering users N_i in a given time interval is obtained from a Poisson distribution as explained in section II, with $\lambda = 0.25$, 0.7, and 1. The 128×128 received signals' spectrogram (see section IV-A is considered as the input of the AE. As previously mentioned, Bayesian optimization is performed to set the hyperparameters of both the AE and CNN: the number of convolutional layers (nCL), fully connected layers (nFC), filters (nf); the size of the filters (sf) and fully connected layers (sFC). The Bayesian optimization step is done independently for each value of λ. Moreover, *a priori* experimental trials showed that running that step for SNR $= -8$ dB is enough to have a network structure that could be generalized for the other considered SNR values. Tables I and II list the optimized structure for both AE and CNN parts, where the subscripts indicate the number of the corresponding layer. The hyperparameter values are given for the encoder, and the decoder part of the AE is symmetric to the encoder structure. Roughly, the number of kernels of the AE and their sizes expands with the severity of the interference. In contrast, the hyperparameters of the CNN part are almost uncorrelated with the interference parameter λ. In fact, since the AE is trained to provide noise-and-interference free spectrograms, this makes the CNN much less sensitive to the interference effect. Note that the size of the last FC layer is set to M as described in Fig. 4. In order to perform the denoising step independently of the symbol decoding step, the denoising autoencoder and the detector are trained sequentially. The AE is trained for 30 epochs, with a mini-batch size of 100 samples, and the mean squared error (MSE) function is used as the loss function. The CNN network is trained to minimize the cross-entropy loss for 60 epochs with a mini-batch size of 300 samples. To optimize the learnable parameters for both AE and CNN structures, we employed the ADAM algorithm with an initial learning rate of 0.001. The symbol error rate (SER)

TABLE I: Bayesian-based tuned hyperparameters of the convolutional autoencoder for different λ.

λ	nCL	nf_1	nf_2	nf_3	nf_4	sf_1	sf_2	sf_3	sf_4
0.25	4	15	23	30	35	8	5	4	16
0.7	4	94	18	58	89	5	14	16	20
1	4	80	97	51	99	12	10	18	19

TABLE II: Bayesian-based tuned hyperparameters of the CNN for different λ.

λ	nCL	nf_1	nf_2	sf_1	sf_2	nFC	sFC_1	sFC_2
0.25	2	15	80	3	8	2	915	128
0.7	2	83	96	6	12	2	879	128
1	2	45	98	3	12	2	459	128

as a function of SNR is used to compare the performance of the proposed detector with the classical LoRa detector, the coherent LoRa detector, and the CNN-based detectors in [5] and [6].

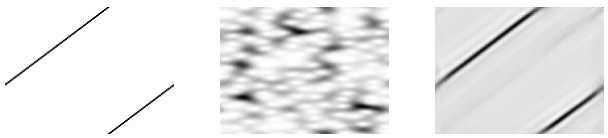
A. Performance evaluation

Training and testing are performed for each (SNR, λ) pair. It should be noted that typical LoRa detectors are able to estimate the SNR. The average number of potentially interfering devices is represented by λ. As the gateway knows its environment, including the number of users using the same spreading factor, an estimation of λ could also be implemented. We also note that in the absence of interference, the proposed detector performs very close to the coherent LoRa and classical LoRa (noncoherent) detectors. However, in such a case, $\lambda = 0$, switching to the coherent or classical receiver seems the best option.

For illustration, we first show in Figs. 5 and 6 the effect of the denoising on a given symbol for two different values of SNR. In Fig 5 and 6, the term noisy represents the presence of noise plus interference in the signal.

(a) Noise-free (b) Noisy (c) Denoised

Fig. 5: Spectrogram of (a) a noise-free LoRa symbol of index 116, (b) before and (c) after the AE, SNR $= -8$ dB, $\lambda = 0.25$, SF $= 7$.



(a) Noise-free (b) Noisy (c) Denoised

Fig. 6: Spectrogram of (a) a noise-free LoRa symbol of index 116, (b) before and (c) after the AE, SNR $= -12$ dB, $\lambda = 0.25$, SF $= 7$.

Figure 7 shows the simulation results for $\lambda = 0.25$, 0.7, and 1. The performance curves illustrate that the proposed AE-CNN detector outperforms the CNN-based detectors in [5], [6], classical LoRa detector, and coherent LoRa detector for all λ.

In the lowest number of interfering users ($\lambda = 0.25$), the proposed AE-CNN detector is more efficient than the CNN-based detectors, with at least 2 dB gain for a target SER of 10^{-4} (cf. Figs. 7a). In the case of a higher number of interfering users ($\lambda = 0.7$ and $\lambda = 1$), the proposed AE-CNN detector attains 3 dB gain compared to the CNN-based detectors (cf. 7b and Fig. 7c). Furthermore, the classical LoRa detector suffers a higher performance loss in the presence of interference.

Figure 8 investigates the impact of time and frequency synchronization errors on the performance of the proposed detector. SD denotes the standard deviation of the errors modeled as zero-mean Gaussian random variables. First, the robustness to time synchronization errors was tackled by considering SD $= T_s/(2M)$. The performance is degraded but remains in the same order of magnitude as the perfect synchronization case. Then frequency synchronization errors were considered with SD $= B/(2M)$, leading to the same conclusions. Note that both SD values correspond to half the time and frequency resolutions, which is reasonably considered poor synchronization [15].

B. Computational Complexity

As a series of convolutional layers, and based on [16], the complexity order of the proposed detection scheme is estimated to be $\mathcal{O}(N^4) + \mathcal{O}(N^2 M)$. The complexity of the spectrogram calculation is neglected in front of $\mathcal{O}(N^4)$. If a maximum resolution ($N = 2^{\text{SF}}$) is chosen, the resulting exponential complexity can be limiting for high SF (12 in particular). However, we focus on the uplink scenario, and the detector is the access point, which can be more expensive and reduce energy constraints so that an implementation could be envisioned.

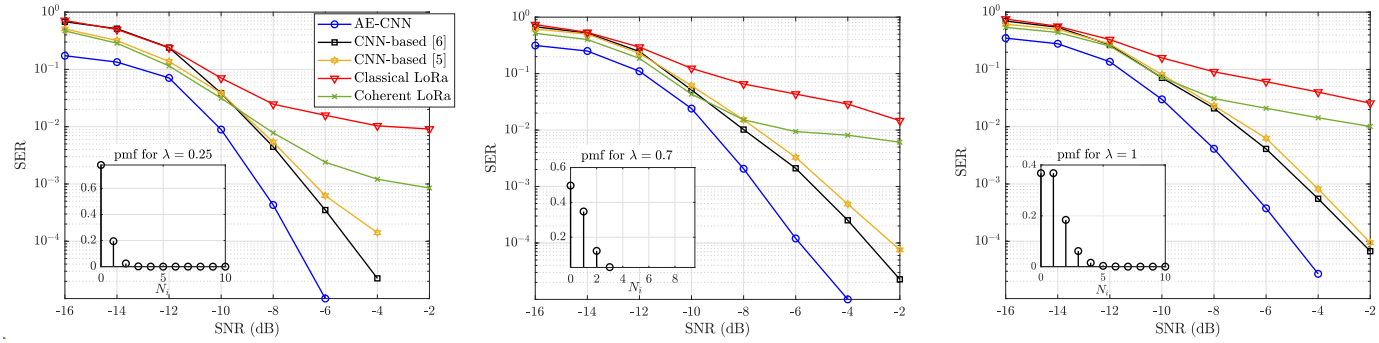
Besides, the size of the spectrogram has not to be conditioned on the signal length M. In fact, we can significantly reduce N to limit the complexity. In table III we show the performance with SF $= 9$ and SF $= 12$ for a given λ and SNR when we change N from 2^{SF} to 256 and 128. The performance loss is very limited while the complexity is significantly reduced, as illustrated by the duration needed to decode one hundred symbols (on a computer using Matlab, so not an optimized implementation), see table IV.

TABLE III: Comparison of SER when different SF and image sizes are considered for $\lambda = 0.7$ and SNR$= -10$ dB.

	SF $= 9$	SF $= 12$
$N = 2^{\text{SF}}$	5.07×10^{-4}	6.38×10^{-6}
$N = 256$	5.26×10^{-4}	7.57×10^{-6}
$N = 128$	5.33×10^{-4}	7.62×10^{-6}

TABLE IV: Average computational time (in second) to decode a 100 symbols packet length, for $\lambda = 0.7$, SNR$= -10$ dB, and when different image sizes are considered.

	SF $= 7$	SF $= 9$	SF $= 12$
$N = 2^{\text{SF}}$	0.186	1.80	21.9
$N = 128$	0.186	0.254	0.791



(a) $\lambda = 0.25$ (b) $\lambda = 0.7$ (c) $\lambda = 1$

Fig. 7: Symbol error rate as a function of the SNR for different detection approaches: classical detector, coherent detector in [11], CNN-based [5], [6], and the proposed AE-CNN detector when $B = 250$ kHz, SF $= 7$. The plot inside shows the probability mass function (pmf) related to the number of interfering users.

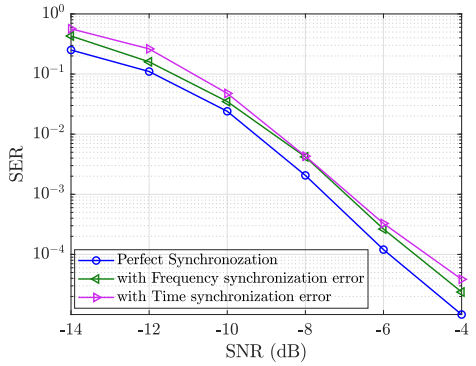


Fig. 8: SER as a function of the SNR for the proposed AE-CNN detector with time (SD $= \frac{T_s}{2M}$) and frequency (SD $= \frac{B}{2M}$) synchronization errors, $\lambda = 0.7$.

VI. Conclusion

This paper proposes a new detector for LoRa-like networks, which combines a denoising autoencoder and a CNN-based detector. The main idea is to use a denoising autoencoder to alleviate the impact of the noise and interference. The proposed detector can decode the user of interest's signals in the presence of multiple interfering users, which transmit simultaneously using the same transmission setting (frequency channel and SF). The results show that the proposed AE-CNN detector outperforms the CNN-based, the coherent, and the classical LoRa detectors when there is interference coming from other users. The proposed detector shows 3 dB gain for a target SER of 10^{-4}. Additionally, the classical LoRa detector suffers a significant performance loss when interfering users increase. Deep learning-based approaches, such as denoising autoencoder and CNN, appear then to be a potential choice for addressing noise and interference in LoRa networks. As machine learning-based decoding approaches are known for their capacity to learn transmission mismatches, a good expansion to the proposed detection scheme would be to consider synchronization errors in the training process (i.e., considering features with synchronization errors) in order to make the receiver even more robust to those errors.

References

[1] Q. M. Qadir, "Analysis of the reliability of LoRa," *IEEE Communications Letters*, pp. 1–1, 2020.

[2] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 01 2008, pp. 1096–1103.

[3] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[4] C. Pham, A. Bounceur, L. Clavier, U. Noreen, and M. Ehsan, "Investigating and Experimenting Interference Mitigation by Capture Effect in LoRa Networks," in *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, ser. ICFNDS '19. New York, NY, USA: Association for Computing Machinery, 2019.

[5] K. Dakic, B. Al Homssi, A. Al-Hourani, and M. Lech, "LoRa Signal Demodulation Using Deep Learning, a Time-Domain Approach," in *2021 IEEE 93rd Vehicular Technology Conference*, 2021, pp. 1–6.

[6] A. A. Tesfay, E. P. Simon, S. Kharbech, and L. Clavier, "Deep Learning-Based signal detection for uplink in LoRa-Like networks," in *2021 IEEE 32nd PIMRC*, Helsinki, Finland, Sep. 2021.

[7] T. Wada, T. Toma, M. Dawodi, and J. Baktash, "A Denoising Autoencoder based wireless channel transfer function estimator for OFDM communication system," in *ICAIIC*, 2019, pp. 530–533.

[8] E. Almazrouei, G. Gianini, N. Almoosa, and E. Damiani, "A deep learning approach to radio signal denoising," in *2019 IEEE WCNCW*, 2019, pp. 1–8.

[9] C. Mio and G. Gianini, "Signal reconstruction by means of embedding, clustering and autoencoder ensembles," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[10] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "LoRa Digital Receiver Analysis and Implementation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1498–1502.

[11] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Investigating theoretical performance and demodulation techniques for lora," in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–6.

[12] Semtech, "AN1200.22: LoRa Modulation Basics," Semtech Corporation, Tech. Rep., 2015.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[14] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[15] T. Ameloot, H. Rogier, M. Moeneclaey, and P. Van Torre, "LoRa Signal Synchronization and Detection at Extremely Low Signal-to-Noise Ratios," *IEEE Internet of Things Journal*, 2021.

[16] K. He and J. Sun, "Convolutional Neural Networks at constrained time cost," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5353–5360.