NELoRa: Towards Ultra-low SNR LoRa Communication with Neural-enhanced Demodulation

Chenning Li[1], Hanqing Guo[1], Shuai Tong[2], Xiao Zeng[1], Zhichao Cao[1], Mi Zhang[1], Qiben Yan[1], Li Xiao[1], Jiliang Wang[2], Yunhao Liu[2]
[1]Michigan State University [2]Tsinghua University

ABSTRACT

Low-Power Wide-Area Networks (LPWANs) are an emerging Internet-of-Things (IoT) paradigm marked by low-power and long-distance communication. Among them, LoRa is widely deployed for its unique characteristics and open-source technology. By adopting the Chirp Spread Spectrum (CSS) modulation, LoRa enables low signal-to-noise ratio (SNR) communication. However, the standard demodulation method does not fully exploit the properties of chirp signals, thus yields a sub-optimal SNR threshold under which the decoding fails. Consequently, the communication range and energy consumption have to be compromised for robust transmission.

This paper presents NELoRa, a neural-enhanced LoRa demodulation method, exploiting the feature abstraction ability of deep learning to support ultra-low SNR LoRa communication. Taking the spectrogram of both amplitude and phase as input, we first design a mask-enabled Deep Neural Network (DNN) filter that extracts multi-dimension features to capture clean chirp symbols. Second, we develop a spectrogram-based DNN decoder to decode these chirp symbols accurately. Finally, we propose a generic packet demodulation system by incorporating a method that generates high-quality chirp symbols from received signals. We implement and evaluate NELoRa on both indoor and campus-scale outdoor testbeds. The results show that NELoRa achieves 1.84-2.35 dB SNR gains and extends the battery life up to 272% (~0.38-1.51 years) in average for various LoRa configurations.

CCS CONCEPTS

• **Networks** → **Network protocol design**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

IoT, LPWAN, LoRa, Machine Learning for Wireless Systems

ACM Reference Format:
Chenning Li[1], Hanqing Guo[1], Shuai Tong[2], Xiao Zeng[1], Zhichao Cao[1], Mi Zhang[1], Qiben Yan[1],, Li Xiao[1], Jiliang Wang[2], Yunhao Liu[2]. 2021. NELoRa: Towards Ultra-low SNR LoRa Communication with Neural-enhanced Demodulation. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys'21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3485730.3485928

SenSys'21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11…$15.00
https://doi.org/10.1145/3485730.3485928

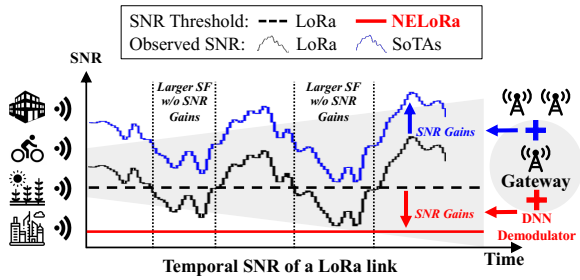


Figure 1: SNR gains can benefit energy efficiency by keeping SF small. State-of-the-arts (SoTAs) use multiple gateways/nodes to enhance the observed SNR. We design a DNN demodulator (NELoRa) which lowers the SNR threshold of the chirp symbol decoding at a single gateway.

1 INTRODUCTION

Recent years have witnessed the emergence of Low-Power Wide-Area Networks (LPWANs) as a promising mechanism to connect billions of low-cost Internet of Things (IoT) devices for wide-area data collection (e.g., smart-industry, smart-city, smart-agriculture) [29, 32]. Long Range (LoRa) [1], SIGFOX [5], and NB-IoT [40] are the three commercialized wireless technologies that facilitate the establishment of LPWANs. Among them, LoRa is the only open-source one and works on unlicensed frequency bands. By modulating data via Chirp Spread Spectrum (CSS), LoRa allows sensor nodes to send data at low data rates to gateways several or even tens of miles away. Unfortunately, recent studies [8, 10, 12, 14, 15, 22, 27, 28, 31, 50] show that the communication range of LoRa is far from the expectation in complex real-world environments (e.g., urban areas, campus). The blockage attenuation could severely degrade the Signal-to-Noise Ratio (SNR) of LoRa packets, causing decoding failures even at a sub-kilometer distance. Consequently, a LoRa node has to adapt its configuration with more energy consumption to compensate for the SNR degradation, reducing its battery life drastically.

In LoRa, Spreading Factor[1] (SF) and Bandwidth (BW) are two key configurable knobs that balance the range and energy consumption of LoRa communication [44, 45]. Given a specific LoRa configuration (i.e., SF, BW), the standard LoRa demodulation method, *dechirp*, determines an SNR threshold above which chirp symbols can be decoded. A larger SF enables a lower SNR threshold, which results in a longer communication range and larger energy consumption under the same BW. To minimize battery drain for packet transmission, LoRa employs a rate adaption strategy that uses the observed historical SNR to choose the smallest feasible SF [10]. Intuitively, if we can obtain extra SNR gains to enlarge the gap between the observed

[1]The spreading factor denotes the number of bits that can be encoded per chirp symbol, determining the data rate of LoRa's CSS modulation.

Table 1: Comparison between NELoRa and recent studies regarding primary feature, performance, and deployment cost. GW, HD, and MDTS stand for the gateway, hardware diversity, and multi-dimension temporal-spatial features.

	Feature	SNR Gain	GW/Node
Charm [10]	Energy	1-3dB	2-8/1
OPR [2]	Frame	1.5-2.5dB	2-6/1
Chime [14]	Freq.	2.4-3.4dB	4-6/1
Choir [12]	HD	N/A	1/36
NELoRa	MDTS	1.8-2.4dB	1/1

SNR and the SNR threshold determined by a LoRa configuration, the communication range will be enlarged, and the upper layer protocol will have more spaces to extend the battery lifetime [2, 12, 14].

Figure 1 illustrates an example of how the extra SNR gains could help improve LoRa nodes' energy efficiency. Specifically, the black curve represents the observed SNR of a LoRa link, fluctuating at the SNR threshold of the currently applied SF indicated by the black dashed line. When the observed SNR is lower than the SNR threshold, a larger SF must be applied for successful decoding. As a result, more energy is consumed during this period. However, if the extra SNR gains can be obtained through either increasing the observed SNR to the blue curve (e.g., blue up-arrow) or reducing the SNR threshold to the red line (e.g., red down-arrow), the larger SF is no longer needed so that the energy efficiency of LoRa nodes can be significantly improved.

Status Quo and their Limitations. Status quo approaches [2, 10, 12, 14] obtain extra SNR gains by enhancing the observed SNR over a weak LoRa link as the blue curve depicted in Figure 1. Such SNR gains are obtained by leveraging information collected from multiple LoRa nodes or gateways. For instance, Choir [12] leverages up to 36 co-located LoRa nodes to boost the received signal strength; Charm [10] utilizes the spatial diversity of 2 to 8 gateways to decode weak chirp symbols through coherent combining; OPR [2] explores the disjoint link-layer bit errors across 2 to 6 gateways to recover the corrupted packets; and Chime [14] uses a heart-beat packet and three gateways to estimate wireless channel state to select the optimal frequency for packet transmissions.

As summarized in Table 1, although these approaches have achieved impressive SNR gains, such gains are obtained costly if the LoRa nodes and gateways are not densely deployed. The root cause of the limitations shared across the status quo approaches described above is that they are all designed based on dechirp, which decodes a chirp symbol by *only* relying on its energy in the spectrum [12]. Such a design choice, though simple, is *coarse-grained*: it ignores fine-grained information embedded inside the chirps, which can be helpful in chirp symbol decoding.

Overview of the Proposed Approach. The limitation of dechirp motivates us to rethink the design of the LoRa demodulation method. To this end, we present NELoRa[2], a neural-enhanced demodulation method that achieves ultra-low SNR LoRa communication with a single gateway. The key idea of NELoRa is to use Deep Neural Networks (DNN) to extract the *fine-grained* information embedded inside the chirps for decoding. Compared to the *single-dimension* energy information used in dechirp, the extracted fine-grained information contains robust and consistent *multi-dimension* patterns across time, frequency, phase, and energy information of the chirps. By doing this, NELoRa breaks the SNR threshold of dechirp and obtains extra SNR gains by lowering the SNR threshold depicted as the red line in Figure 1. As a result, NELoRa can enlarge the LoRa communication range and reduce the energy consumption at a single gateway.

First, NELoRa incorporates a dual-channel spectrogram to create a multi-dimension feature space, in which extra information beyond energy can be extracted for decoding chirp symbols. The dual-channel spectrogram contains not only the amplitude but also the phase. Since amplitude and phase are orthogonal regarding different noise, diverse high-level features extracted from the dual-channel spectrogram provide the foundation to decode chirp symbols at ultra-low SNR levels (§3.2.1).

Second, NELoRa incorporates a dual-DNN design. The first DNN acting as a noise filter recovers clean chirp symbols by masking their noisy input spectrogram. The second DNN acts as an adaptive decoder, which classifies the recovered chirp symbols. Given the finite coding space of LoRa, training and testing datasets share the same chirp symbols except for the distorted noises. Hence, the well-known "bad" overfitting[3] of DNN can be turned into a useful characteristic for chirp symbol decoding [52]. We further compress our dual-DNN design in terms of latency and parameter size to efficiently run on a resource-constrained gateway (§3.2.2).

Third, NELoRa incorporates a chirp-level data synthesis scheme to enhance its generalization capability for diverse deployment environments. In comparison with data collected from a specific environment, our chirp-level data synthesis scheme can generate chirp symbols with a wide range of noise given a LoRa configuration (§3.2.3).

Implementation and Evaluation Results. We have implemented NELoRa on a USRP N210 Software Defined Radio (SDR) combined with a back-end host and evaluated its performance with commodity LoRa nodes in both indoor and outdoor deployments. Our results show that NELoRa can achieve 1.8-2.35 dB extra SNR gains across a wide range of LoRa configurations and extend the LoRa node battery lifetime by up to 272% (~0.38-1.51 years).

In summary, our work makes three major contributions:

- To the best of our knowledge, NELoRa represents the first neural-enhanced LoRa demodulation method with the minimum deployment cost. Furthermore, it consistently outperforms the standard method under a wide range of LoRa configurations.
- We have incorporated two novel techniques, including the dual-channel spectrogram for multi-dimension feature space construction and the dual-DNN design for noise removal and chirp symbol decoding. These techniques represent unique contributions that altogether push the SoTA of LoRa systems forward.
- We have implemented a prototype of NELoRa via COTS devices and evaluated its performance in both indoor and outdoor deployments. The results show that NELoRa can achieve 1.84-2.35 dB SNR gains and 0.38-1.51 years battery life improvement.

[2]The datasets and source codes are available at *https://github.com/hanqingguo/NELoRa-Sensys*.

[3]Overfitting refers to a model that precisely models the foreground information in training data.

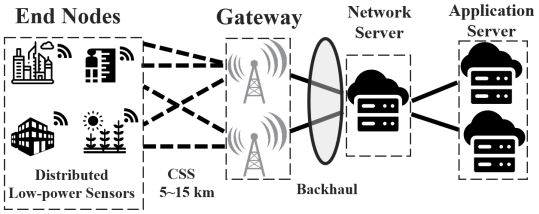


Figure 2: Illustration of LoRaWAN architecture.

2 UNDERSTANDING THE PROBLEM

2.1 LoRaWAN Architecture

As illustrated in Figure 2, a LoRaWAN consists of end nodes, gateways, a network server, and an application server. The collected sensory data (e.g., temperature, humidity) transmitted from the distributed end nodes is relayed by several gateways to the network server. In LoRa's communication stack, its physical layer enables long-distance communication via CSS modulation at the end nodes and dechirp (§2.2) at the gateways. We summarize the deployment issues and protocol design concerns at the end node and gateway sides as follows:

- **End nodes** are widely distributed in a large area and powered by batteries or harvesting energy from solar power and ambient wireless signals [17, 38]. Since energy is precious at the end node, LoRaWAN makes the up-layer protocols as simple as possible. For example, the commonly used class A mode [1] adopts a simple ALOHA media access protocol to avoid the energy consumption on carrier sense. Additionally, end nodes are operating in infrastructure mode. They directly communicate with a gateway without a multi-hop relay among themselves. Hence, the demodulation task is rarely performed on an end node.
- **LoRa gateways** are deployed with tethered power supplies. Thus the energy consumption is no longer a problem at the gateways [44, 45]. The onboard micro-control unit (MCU) (e.g., STM32), however, is computationally limited. Therefore, a low-cost computing platform (e.g., Raspberry PI, Arduino) is also physically connected to provide extra computation resources to execute the tasks (e.g., remote programming). Besides, from the view of network and application servers, the second-level delay at the gateway can also be tolerated due to the low duty cycle of LoRa traffic flows.

The design of NELoRa fits LoRaWAN architecture well. At the end node side, there is no additional cost incurred, and end nodes benefit from our SNR gains, resulting in a longer communication range and battery life. At the gateway side, by leveraging the gateway's tolerance on power consumption and its extra compute resources, NELoRa adopts the deep learning techniques for weak chirp symbol decoding.

2.2 Standard Modulation and Demodulation

LoRa uses CSS modulation [3]. Given the pre-configured BW, CSS first defines a base up-chirp whose frequency increases linearly at the rate of k over time from $-\frac{BW}{2}$ to $\frac{BW}{2}$, denoted as $C(t) = e^{j2\pi(-\frac{BW}{2}+\frac{kt}{2})t}$. The data bits are then encoded by shifting the initial frequency of a base up-chirp to f_s, rendering a chirp symbol

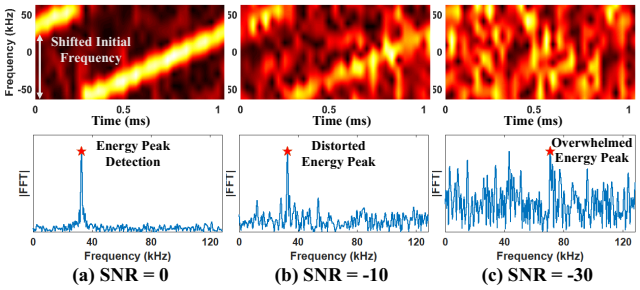


Figure 3: In dechirp, the energy peak of a chirp symbol's spectrum is distorted or overwhelmed as the SNR decreases.

as $y_e = C(t)e^{j2\pi f_s t}$, shown in Figure 3a (top). Propagating through the wireless channel, the received chirp symbol y_r at the gateway can be formulated as follows:

$$y_r[n] = hy_e[n] + w[n] \ (n = 0, 1, \cdots, N-1) \tag{1}$$

where $y_r[n]$ and h are the n^{th} sample of the chirp symbol with total $N = 2^{\text{SF}}$ samples and the amplitude of the received chirp symbol. w is the channel noise following the compound Gaussian distribution [46] in the I-Q space, namely $\Re(w) \sim \mathcal{N}(0, \sigma^2)$ and $\Im(w) \sim \mathcal{N}(0, \sigma^2)$.

A LoRa receiver adopts dechirp to decode the initial frequency f_s of a received chirp symbol by first multiplying the chirp symbol with a time-aligned base down-chirp, indicated as C^{-1}, the conjugate of the base up-chirp. With Fast Fourier Transform (FFT), Equation (2) demonstrates the energy of the frequency component $|X[m]|$ at bin m:

$$|X[m]| = |\mathcal{F}(y_r \cdot C^{-1})| = |\sum_{n=0}^{N-1} e^{-j2\pi\frac{nm}{N}}(hy_eC^{-1} + \hat{w})| \tag{2}$$

$X[m]$ consists of two parts. One is $X_c[m]$ indicating the energy from chirp symbol y_e. The other is noise energy $X_n[m]$ (i.e., $\hat{w}$) that still follows the compound Gaussian distribution [46] with parameter σ. Equation (3) shows $|X_c[f_s]|$ can be estimated as the product of the amplitude h and the total sample number N.

$$|X_c[f_s]| = \lim_{m/T \to f_s} |\sum_{n=0}^{N-1} he^{j2\pi(-\frac{nm}{N}+f_s\frac{nT}{N})}| = h \times N \tag{3}$$

Hence, as shown in Figure 3a (bottom), the energy of the chirp symbol can be accumulated and form an energy peak at the frequency f_s in the spectrum [46]. Consequently, in dechirp, we determines f_s by finding the frequency bin with the maximum energy.

However, Equation (4) depicts the maximum noise energy $|X_n[m]|$, which follows the Rayleigh distribution with $\mathcal{N}(0, \sigma^2)$ [44].

$$max|X_n[m]| \approx \sqrt{2\sigma^2 N \times H_{N-1}} \tag{4}$$

where H_{N-1} is the value of $N-1$ term harmonic series [41] as $\sum_{n=1}^{N-1}\frac{1}{n}$. Consequently, as shown in Figure 3b and 3c, when SNR is decreased gradually, the energy peak can be distorted or even overwhelmed by the noise energy.

To decode a received chirp symbol, the energy peak $|X_c[f_s]|$ should be higher than the maximum noise energy $|X_n[m]|$ in the spectrum. To ensure that the energy peak is not overwhelmed by the noise energy ($|X_c[f_s]| > max|X_n[m]|$), Equation (5) provides

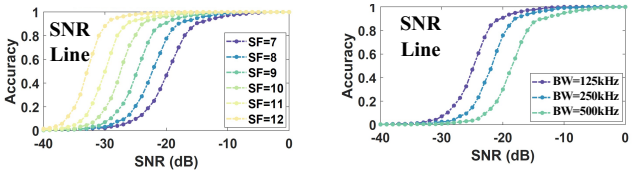


Figure 4: The SNR threshold of dechirp under different LoRa configurations across different SFs (left) and BWs (right).

the SNR threshold, under which chirp symbols cannot be decoded by dechirp:

$$SNR_{thres} = 10lg(\frac{h^2}{2\sigma^2}) = 10lg(\frac{H_{N-1}}{N}) \quad (5)$$

Furthermore, we use our synthesis chirp symbol dataset collected from our indoor testbed (§5) to validate the existence of the SNR threshold in Equation (5). Figure 4 illustrates the decoding accuracy of dechirp as SNR decreases. Specifically, it shows the SNR threshold under different configurations across SFs and BWs. With higher SF or smaller BW, the SNR threshold is getting lower. For example, when BW=125kHz and SF=12, we achieve over 90% accuracy when SNR is larger than -30 dB. Although the decoding accuracy of dechirp can achieve 90% when the SNR is higher than -13 dB across all the experimental settings, we argue that the derived SNR threshold is *sub-optimal* since the energy feature only reflects a part of the chirps, which motivates the design of NELoRa.

3 NELORA OVERVIEW

3.1 NELoRa Architecture

Figure 5 illustrates the overall architecture of NELoRa. NELoRa consists of three stages to realize reliable symbol generation and neural-enhanced demodulation. In the *Packet Identification* stage, a LoRa packet is first detected from raw signal samples via the *Chirp Enhance* and *Preamble Detection* modules. The detected packet is then putted into the *DNN Input Generation* stage. The *Offset Recovery* module exploits the redundant chirp symbols in packet preamble to compensate offsets in frequency and time domains to generate the time-aligned and offset-free chirp symbols in packet payload. Each extracted chirp symbol is then transformed by the *Symbol Transform* module into a dual-channel spectrogram. The final stage is *DNN-based Demodulation*. Given the dual-channel spectrogram, the *Mask-enabled Filter* module alleviates the channel noise to obtain a masked spectrogram, which is decoded by the *Spectrogram-based Decoder* module to generate the packet.

3.2 Key Design Choices

The design of NELoRa involves three critical design choices. We provide an overview of these design choices in this section before presenting details of NELoRa in the next section.

3.2.1 *Feature Space Selection.*

Key Issue: A DNN consists of multiple layers to capture different kinds of features from its input [51]. A multi-dimensional input is preferred to enlarge the feature spaces. Many wireless sensing systems [6, 25] demonstrate the effectiveness of feeding a DNN with a 2D spectrogram via Short-time Fourier Transform (STFT).

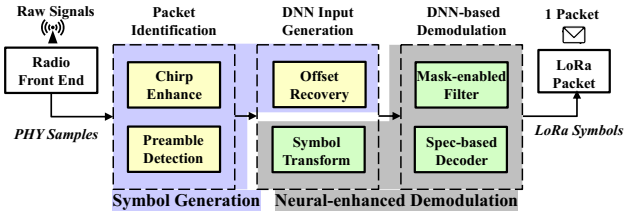


Figure 5: NELoRa's architecture integrates symbol generation (purple) and neural-enhanced demodulation (gray).

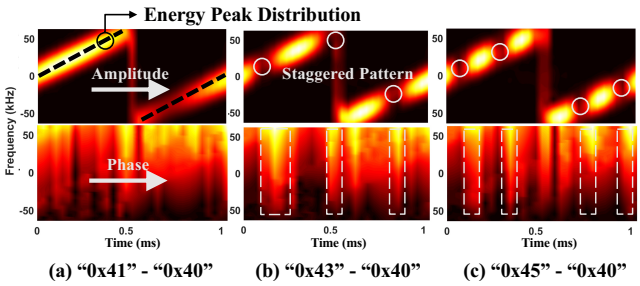
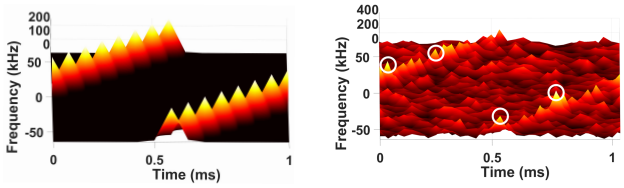


Figure 6: The dual-channel spectrogram difference (top: amplitude; bottom: phase) between a pair of chirp symbols. (a) (0x41, 0x40); (b) (0x43, 0x40); (c) (0x45, 0x40).

However, compared to the energy feature extracted from a chirp symbol's 1D spectrum, whether its 2D spectrogram can provide a richer feature space that benefits LoRa demodulation at low SNR levels is questionable.

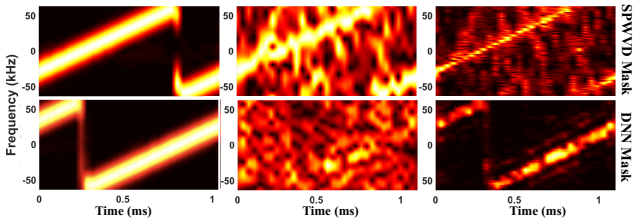
Our Approach: To address this issue, given a chirp symbol, we first divide it into a sequence of short chirp segments with equal length. Then we compute the spectrum of each short chirp segment separately as dechirp does to generate an amplitude spectrogram. In addition, we extract the phase of each short chirp segment's spectrum, which leads to a **dual-channel spectrogram** in § 4.2. To demonstrate the feature space of our dual-channel spectrogram, we collect four SF-7 chirp symbols representing four different but very close data bits (e.g., 0x40, 0x41, 0x43, 0x45) at a high SNR level. The dual-channel spectrogram creates the desired feature space only if the spectrograms of the four chirp symbols are distinct enough. Then, we take 0x40 chirp symbol as a reference to compute the dual-channel spectrogram differences with others. The results are shown in Figure 6. The spectrograms of amplitude and phase are on top and bottom, separately.

For a chirp symbol, the spectrum energy peaks derived by continuous short chirp segments form a linearly increasing **energy peak distribution** in its amplitude spectrogram. The initial frequency of the energy peak distribution is determined by the initial frequency of the chirp symbol, which corresponds to the data bits it represents. As shown at the top of Figure 6, although the encoded initial frequencies among the four chirps are close, we can clearly observe the energy peak distributions from the amplitude spectrogram differences. This indicates energy peak distribution is a useful feature dimension. Additionally, as shown in Figure 6b and Figure 6c, we can see peaks (e.g., bright areas) and valleys (e.g., dark areas) alternatively appear in both amplitude (e.g., circle) and phase (e.g.,



(a) SF=7, SNR=35 (b) SF=7, SNR=-10

Figure 7: The linearly increasing energy peaks of the short chirp segments at different SNR levels. (a) high SNR level; (b) under noise floor.



(a) GT Spectrogram (b) Raw Spectrogram (c) Masked Spectrogram

Figure 8: Given (a) ground-truth spectrogram at high SNR levels and (b) raw spectrogram at low SNR levels, (c) masked spectrogram is generated by SPWVD (top) and our DNN noise filter (bottom).

dashed rectangle) spectrograms. Specifically, the patterns observed by amplitude and phase are correlated, but different chirp symbols exhibit diverse patterns. Hence, the **staggered pattern** is another feature dimension of the dual-channel spectrogram to distinguish different chirp symbols.

When SNR is getting low, however, the dual-channel spectrogram will be polluted by noise. To illustrate this, we collect an SF-7 chirp symbol to calculate its amplitude spectrogram under different SNR levels. As shown in Figure 7a, when SNR is 35 dB, we can see the spectrum energy peaks of all short chirp segments. When SNR drops to -10 dB, Figure 7b shows only several short chirp segments' energy peaks (e.g., white circles) can be explicitly observed compared to surrounding noise energy.

Facing the seriously polluted dual-channel spectrogram, a DNN can succeed in recognizing chirp symbols due to the noise-resilient patterns obtained from both amplitude and phase spectrograms. Specifically, the energy peak distribution exhibits a linear pattern, which can still be observed with several explicit energy peaks in Figure 7b. Moreover, the staggered pattern exists in both amplitude and phase spectrograms. Since the amplitude and phase of a short chirp segment's spectrum are affected by the noise independently, the staggered pattern has the potential to tolerate specific noise. A well-designed DNN is good at learning these patterns. Although random noises may be much stronger than chirp symbols, it is hard to simultaneously form similar patterns in multi-dimensional feature space to mislead the DNN. Hence, we feed the dual-channel spectrogram of a chirp symbol to our DNN.

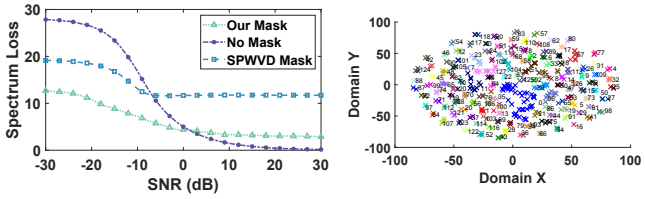
3.2.2 *Feature Learning Effectiveness.*

Key Issue: The primary goal of NELoRa is to accurately decode different chirp symbols at ultra-low SNR levels. To achieve this, NELoRa must learn the unique patterns of each chirp symbol from its dual-channel spectrogram even if intense noise exists. However, existing spectrogram-based DNN models [6, 25, 53, 54] cannot be directly adopted since their black-box feature extraction process becomes less efficient in our classification task. Specifically, there are 1024 different chirp symbols when SF is 10. Such a large amount of classification categories of chirp symbol decoding make these DNN models hard to learn any useful pattern under ultra-low SNR levels. Moreover, without any prior knowledge about the correct pattern of each chirp symbol, the pattern features extracted by the DNN models are easily misled by the patterns of strong channel noises if only back-propagating the information whether a chirp symbol is successfully classified. Hence, a tailored DNN model is required to achieve high feature learning effectiveness.

Our Approach: To address this issue, we design a dual-DNN model, which consists of a noise filter and a chirp symbol decoder for noise reduction and chirp symbol decoding, respectively. Taking the dual-channel spectrogram of a chirp symbol as input, the noise filter is a DNN that outputs a masked spectrogram that mimics the ideal patterns of the chirp symbol. Analogous to the wildly-used Smoothed Pseudo Wigner-Ville Distribution (SPWVD) [4, 7, 39], our noise filter aims to preserve principal chirp symbol patterns in a raw spectrogram instead of understanding the noise distribution for noise cancellation [53, 54]. The reason is that the types of noise distribution are infinite; but given a LoRa configuration, the types of chirp symbols are finite. For each chirp symbol, we can generate its ground-truth (GT) dual-channel spectrogram at high SNR levels as labels (e.g., Figure 8a) to train the DNN together with the raw spectrogram (e.g., Figure 8b) at different low SNR levels. Then, we fully exploit the DNN's over-fitting property to learn the unique patterns, which enables efficient pattern masking under ultra-low SNR levels. Figure 8c shows that our DNN noise filter and SPWVD can derive the masked spectrogram, similar to the GT one, but our DNN-masked spectrum looks better in terms of pixel similarity.

To quantify the effectiveness of our DNN noise filter against SPWVD [4, 7, 39], we further measure the pixel-wise spectrum loss defined as the average spectrum energy variance between the masked spectrogram and the GT one. The less the spectrum loss is, the more spectrogram patterns are retained. We use the spectrum loss of the raw spectrogram without any masking as the baseline. We collect all types of SF-7 chirp symbols at high SNR levels and generate the chirp symbols at different low SNR levels in the range of [-30, 0) dB by injecting noise. The results are shown in Figure 9a. When SNR is less than 0 dB, both masking methods can filter out the noise compared with no masking. Additionally, the spectrum loss of our DNN noise filter shows a similar increasing trend with that of SPWVD, but is always lower. The spectrum loss of our DNN noise filter is about 10 at -20 dB SNR or higher, while SPWVD can achieve the same spectrum loss only if SNR is larger than -10 dB. This verifies that our DNN noise filter can achieve more effective pattern preservation than SPWVD under low SNR levels.

However, we can see that when SNR is higher than 10 dB, the spectrum loss of the raw spectrogram (i.e., no masking) is less than that of our DNN noise filter. This indicates our DNN noise filter induces a certain level of spectrum energy loss which is independent of the SNR level. Hence, we cannot direct adopt dechirp to



(a) Masking Spectrum Loss (b) Symbol Clusters via DNN

Figure 9: The effectiveness of our DNN noise filter and decoder. (a) the comparison of different masking techniques; (b) SF-7 chirp symbol clusters in the feature space.

decode the chirp symbol converted by a masked spectrogram. To compensate for the spectrum energy loss of our DNN noise filter, we design the other DNN, which takes the masked spectrogram of a chirp symbol as input and outputs the data bits represented by the chirp symbol. Our spectrogram-based DNN decoder explores the multi-dimension feature space created by our dual-channel spectrogram to over-fit the pattern difference among different chirp symbols, which can tolerate the spectrum energy loss to achieve accurate classification. As shown in Figure 9b, given 8,000 SF-7 chirp symbols across SNR from -20 dB to 0 dB, our spectrogram-based DNN decoder can distinguish all $2^7 = 128$ symbol clusters in our high-level pattern feature space via t-SNE [33, 56].

3.2.3 DNN Model Generalization and Compression.

Key Issue: In practice, considering the retraining cost and computation resource limitation while deploying NELoRa, we must provide a one-fits-all DNN model to avoid extra retraining overhead in a new environment and compress the DNN model to fit the limited computation resource at a gateway. Two key factors can incur a biased training dataset, by using which we will fail to train a one-fits-all DNN model. One is the inaccurate symbol timing alignment or the frequency offset between a LoRa end node and a gateway, which can cause a random shift in the spectrum of a chirp symbol. The random spectrum shift will mislead our DNN model and make it hardly matches the masked spectrogram with the correct data bits it represents. Hence, the chirp symbols in both training and testing processes should be fully time-aligned and contain no hardware-dependent frequency offset. The other key factor is the infinite patterns of random Gaussian noises. To make our DNN model can generate a masked spectrogram with minimum spectrum energy loss under various SNR levels, the training dataset should contain the labeled chirp symbols under all kinds of SNR levels corresponding to the diverse noise levels in different environments. However, facing the long communication range of LoRa, it is exhausted to collect such a training dataset. To compress a DNN model, we face the tradeoff between computation efficiency and classification accuracy. And it is not trivial to significantly reduce the execution latency while keeping a stable classification performance as the original DNN does.

Our Approach: To address this issue, we design a white-box method (§ 4.1) that utilizes multiple base up-chirps in the preamble of LoRa packets to extract "clean" chirp symbols, which are **time-aligned** and **offset-free**, from a LoRa packet even under ultra-low SNR-level. Moreover, we design a data augmentation method to derive a synthesis training dataset. Specifically, after collecting a small set of hardware-independent chirp symbols at narrow SNR levels, we add various kinds of random Gaussian noises with controlled amplitudes to the I and Q traces of these chirp symbols [44, 45]. In this way, we can obtain a high-quality synthesis training dataset that covers all SNR levels. Additionally, we adopt structured pruning [26] and lightweight layers to compress our DNN model to achieve efficient running time on LoRa gateways.

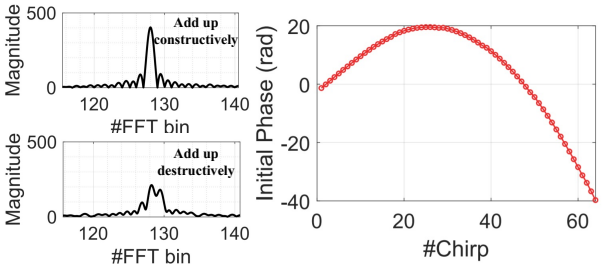
4 DESIGN DETAILS

4.1 High-quality Chirp Symbol Generation

Packet Identification: To reap the benefits of the DNN model in decoding chirp symbols at ultra-low SNR level, NELoRa must efficiently detect incoming LoRa packets, then divide the payload of each packet into some chirp symbols, which are further fed to our DNN demodulator. The default packet detection method utilizes the preamble of a LoRa packet, which consists of multiple continuous base up-chirps. If we apply dechirp on the preamble, several continuous energy peaks appear at FFT bin 0 of the multiple base up-chirps' spectrum. In practice, a gateway continuously applies dechirp on recorded symbol-length signals (called *window signal*). If a LoRa preamble appears, a window signal contains a chirp symbol (called *window chirp*) which may not be exactly time-aligned with the base up-chirps in the LoRa preamble. Considering the multiple continuous base up-chirps in a LoRa preamble, we will observe several identical window chirps. If the energy peaks of several window signals appear at the same FFT bin, a LoRa packet is detected. Then, we align the chirp symbols of the packet by moving the observed FFT bin to bin 0. With the base down-chirps in the packet's SFD (Start Frame Delimiter), we can remove carrier frequency offsets (CFO) [44] to generate high-quality chirp symbols in the packet. However, the dechirp based chirp symbol generation is limited by the SNR threshold of dechirp, so that we still cannot be directly adopted by NELoRa, which intends to achieve a lower SNR threshold.

To tolerate a lower SNR threshold than dechirp's, our basic idea is that instead of using the energy peak of a window chirp, we sum up multiple continuous window chirps to form an enhanced window chirp, in which the window chirps are added up coherently, but the random noise is not. We apply dechirp on the enhanced window chirp to obtain an ideally accumulated energy peak, surpassing the randomly increased noise energy. In theory, when we sum up eight window chirps coherently, the resulted SNR gains will be 9dB. We define a threshold δ. If the energy peak of the enhanced window chirp is at least δ higher than the average noise energy, a LoRa packet is detected.

Offset Recovery: Putting this idea into practice is still challenging due to the existence of CFO and sampling frequency offsets (SFO), which introduce phase shifts onto the window chirps that accumulate over time. Therefore, different window chirps may have different initial phases. To take advantage of coherently overlapping, we cannot directly add up these window chirps. As shown in Figure 10a, we superpose two window chirps and apply dechirp for packet detection. When manually turning the phases of these window chirps and making them add up coherently, the energy peak is much higher than the noise energy, allowing us to detect a LoRa



(a) Chirp Overlapping (b) SF = 12

Figure 10: Packet detection with an enhanced window chirp. (a) Two window chirps are added up coherently or incoherently by tuning their initial phases; (b) Initial phases of continuous window chirps.

packet at the SNR below the dechirp's SNR threshold. Otherwise, window chirps will be added up incoherently, resulting in signal distortion and thus degrading the SNR gains.

NELoRa leverages an optimal phase searching algorithm for accurately measuring and compensating the phase offsets over multiple window chirps. Supposing the frequency bias of a LoRa node's oscillator is κ ppm compared to the oscillator of a gateway, the CFO and SFO are $F_{RF} \times \kappa$ MHz and $F_S \times \kappa$ MHz, respectively, where F_{RF} is the carrier frequency and F_S is the sampling frequency. Both the CFO and SFO influence the phase of the window chirps, but in totally different ways. The CFO introduces continuous phase shifts that accumulate over time, thus the initial phase of the i^{th} window chirp is shifted by

$$\phi_c = i \cdot T \cdot 2\pi \cdot CFO \quad rad \tag{6}$$

where T is the chirp symbol duration. The SFO induces a length difference between the window chirps and the base down-chirp used in dechirp. For example, the actually received window chirp is shorter (or longer) than a base down-chirp by $\tau = T/(1+\kappa)$. This leads to a time offset between each received window chirp and the base down-chirp, accumulating over time. Therefore, at the i^{th} window chirp, the phase shift introduced by the SFO is

$$\phi_s = i \cdot T \cdot 2\pi \cdot (i \cdot \frac{BW}{T}\tau) \quad rad. \tag{7}$$

Putting ϕ_c and ϕ_s together, the phase shift for the i^{th} window chirp can be finally derived as

$$\phi(\kappa, i) = T \cdot 2\pi \cdot (\frac{BW}{1+\kappa} i^2 + F_{RF} \cdot \kappa \cdot i) \quad rad \tag{8}$$

which is a quadratic function related to both the frequency bias and the window chirp index. We verify the correctness of the $\phi(\kappa, i)$ by making a commodity LoRa node transmit a packet with a long preamble. As shown in Figure 10b, the initial phase of each window chirp perfectly fits Equation (8).

Given the frequency bias of a LoRa node is within a limited range under all operating conditions (i.e., $|\kappa| < \Delta$), we can search for κ between $-\Delta$ and Δ to compensate the phase shifting via:

$$\kappa = \arg \max_{-\Delta<\kappa<\Delta} \max |\mathcal{F}[\sum_{i=0}^{N-1} (C_i(t) e^{j\phi(\kappa,i)}) \cdot C^{-1}]| \tag{9}$$

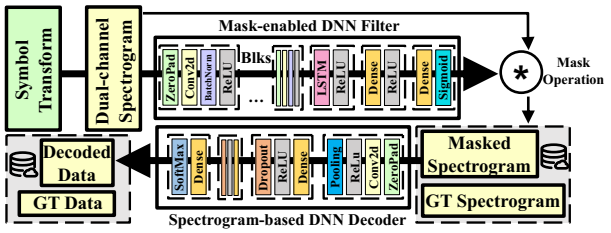


Figure 11: Architecture of our dual-DNN model.

where N is the number of base up-chirps in a LoRa preamble and $C_i(t)$ represents the i^{th} window chirp. We apply stochastic gradient-descent algorithms to speed up the searching process with randomly chosen initial κ. We align the chirp symbols of a LoRa packet by moving the FFT bin where the energy peak of its enhanced window chirp appears to bin 0. Then, we apply κ to remove both CFO and SFO to generate high-quality chirp symbols.

4.2 Neural-enhanced Demodulation

Symbol Transform: To decode the encoded data bits from an extracted chirp symbol, our DNN model takes the dual-channel spectrogram of the chirp symbol as input. First, we take STFT on a chirp symbol $\mathbf{x(n)}$. Then we concatenate the real and imaginary parts to retain the amplitude and phase of the $STFT(\mathbf{x(n)})$ as:

$$STFT(\mathbf{x(n)}, m, \omega) = \sum_{n=-\infty}^{\infty} \mathbf{x(n)} W[n-m] e^{-j\omega n} : \mathbb{C}^n \mapsto \mathbb{R}^{t \times f}$$
$$\mathbf{z} = [\Re(STFT(\mathbf{x(n)})), \Im(STFT(\mathbf{x(n)}))]_{t \times f} \tag{10}$$

where W is the $Hann$ window whose size is m, and the chirp symbol $\mathbf{x(n)} \in \mathbb{C}^n$ is translated into feature $z \in \mathbb{R}^{2 \times t \times f}$ with t sampling points and f frequency bins.

DNN-based Demodulation: Given M chirp symbols in a LoRa packet, our objective is to learn a mapping function $G : X \rightarrow Y^{code}$ from the designed dual-channel spectrogram $X = \{z_i\}_{i=1}^{M}$ to the ground truth encoded data bits $Y^{code} = \{y_i^{code}\}_{i=1}^{M}$.

As shown in Figure 11, our dual-DNN model includes two modules, the noise filter F and the spectrogram-based decoder D for noise reduction and chirp symbol decoding, respectively. The first module aims to preserve the primary spectrogram features of a chirp symbol by masking the raw dual-channel spectrogram z as $F(z) \cdot z$. In a conceptual sense, the noise filter F is more like an end-to-end shortcut connection in the ResNet block [18] by transforming the shortcut from layers into ends. It contains multiple blocks of CNN and one LSTM to fully exploit the spatial and temporal features of the input, followed by two dense layers to output a well-matched mask $F(z)$. Moreover, a four-layer CNN-based decoder is designed to fully capture the spatial energy peak distribution and temporal staggered pattern in the masked spectrogram.

To evaluate the impact for each layer of the DNN noise filter, we illustrate the intermediate mask in Figure 12. Taking a dual-channel spectrogram as input, the 8-channel output from the 8-block CNN module shows a distorted energy peak distribution initially, in which it filters the random noise by exploiting the spectrogram spatially. Figure 12c, the output from the LSTM and Dense layer, presents a relatively clear energy peak distribution

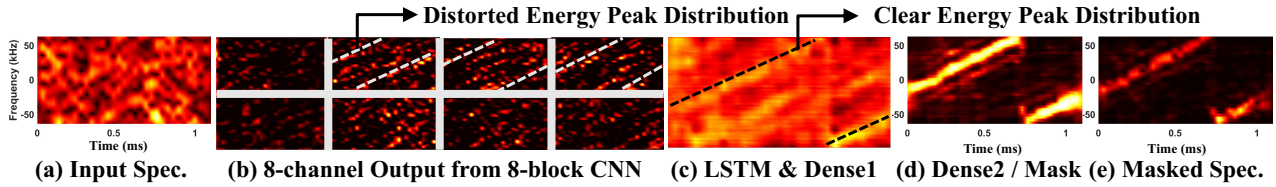


Figure 12: Visualizing intermediate outputs across CNN, LSTM, and Dense layers for chirp symbol at -20 dB SNR.

with the increasing frequency as time passes. We cannot derive a reliable mask until we further compress the intermediate output with another Dense layer, rendering a well-matched mask $F(z)$ in Figure 12d. Finally, by multiplying the mask with the input, we deliver the masked spectrogram $F(z) * z$ in Figure 12e.

To train the DNN noise filter, for each value of the encoded data bits, we collect the corresponding chirp symbols at high SNR-level as the GT spectrogram $Y^{spec} = \{y_i^{spec}\}_{i=1}^{M}$. To enhance the learning processing in the training stage, we design two loss functions corresponding to each DNN for back-propagation. Equation 11 demonstrates that our training goal aims for optimal noise filter F^* and decoder D^*, rendering the least average loss $\mathbb{E}_M$ with M symbols for each batch on the training dataset.

$$F^*, D^* = \arg\min_{F,D} \mathbb{E}_M(\lambda \cdot \text{MSE}(F(z) * z, y^{spec}) + \gamma \cdot \text{CrossEntropy}(D(F(z) * z), y^{code})) \quad (11)$$

where MSE denotes the mean squared error between the masked spectrogram and the GT one while we adopt the cross-entropy loss as the decoding loss, which are weighed by λ and γ, respectively.

Data Augmentation: We improve the generalization of our DNN model by training it with millions of synthesis LoRa chirp symbols, which cover different SNR levels with diverse random noise patterns. Specifically, we collect each type of chirp symbol at high SNR on an indoor testbed. Then, to achieve fine-grained SNR control, we add various Gaussian white noises with controlled amplitude on the collected I and Q traces [44, 45] to generate new chirp symbols.

DNN Model Compression: Though NELoRa can efficiently run on a PC equipped with GPU, the runtime cost increases when deploying on a LoRa gateway with limited resources [55]. Thus, we adopt the structured pruning [13] to compress the original model for efficient running. Specifically, we calculated the L1-norm of weights in each filter of CNN and dense layer and preserved those with the largest L1-norm. Besides, we also replace the LSTM layer with the GRU layer, which is a more computation-efficient version of RNN, while achieving similar performance when the input sequence is not too long. The compression model shows equivalent performance with the initial NELoRa and reduces the inference runtime in our evaluations.

5 IMPLEMENTATION AND EVALUATION METHODOLOGY

Implementation: We have implemented NELoRa and evaluated its performance with commercial LoRa nodes. Figure 13 illustrates the system prototype of NELoRa. Specifically, we use the USRP

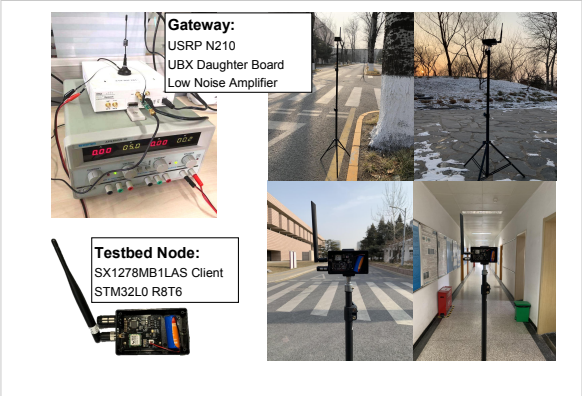


Figure 13: USRP N210 based gateway and commodity SX1278 client radio based LoRa node.

N210 software-defined radio (SDR) platform for capturing over-the-air LoRa signals, operating on a UBX daughter board at the 470MHz bands. The captured signal samples are then delivered to a back-end host for pre-processing and demodulation. Note that demodulation method of NELoRa are hardware-independent, so they can be implemented on any other commercial LoRa gateways as long as the signal samples can be obtained. On the transmitter side, we use SX1278 client radio based commodity LoRa nodes for transmitting LoRa packets.

Chirp Symbol Dataset: The LoRa signals are collected for our training dataset and evaluation of NELoRa. In addition, we configure the LoRa nodes to transmit random payloads at different LoRa configurations (e.g., SFs and BWs) periodically at various locations in an indoor environment. Specifically, we first collect approximately 3,000 LoRa packets at the high SNR (>30 dB), including 4 SFs (e..g, 7, 8, 9, 10) and 3 BWs (e..g, 125K, 250K, 500K). Then, we use our data augmentation method to render 15 million chirp symbols covering -40 dB to 15 dB.

DNN Model Training and Testing: We can traverse all possible SF and BW configurations to detect the applied ones of an incoming LoRa packet at the packet identification. Hence, we train an individual DNN model for each configuration based on the chirp symbols with the corresponding configuration. We further split the dataset into training and test sets. One containing 80% chirp symbols is used for the DNN model training. The test set includes the rest 20% chirp symbols.

System Parameter Settings: In NELoRa implementation, several system parameters are set as follows. A LoRa preamble contains eight base up-chirps. According to the datasheet of SX1278 client

radio, the frequency bias κ is not larger than 50ppm. Hence, we set the searching range Δ as 50ppm. In different environments, we first measure the average level of the noise energy as M, then we empirically set the packet detection threshold δ as $3M$. When we calculate the dual-channel spectrogram of a chirp symbol, give the SF setting, the window size m is set as 2^{SF-1}. Note that we keep the same sampling rate (i.e., 1 MS/s) for a fair comparison between NELoRa and dechirp. Since a higher sampling rate can optimize the packet reception rate and symbol error rate for dechirp, it also benefits NELoRa by improving the resolution of the spectrogram fed into our DNN model, while increasing the running time.

Baseline and Evaluation Metrics: We compare NELoRa with dechirp, the standard demodulation method of LoRa widely used as the baseline by existing studies [12, 42, 44], denoted as the **baseline** below. For a comprehensive comparison, we utilize three metrics to evaluate the performance of NELoRa:
(1) **Symbol Error Rate (SER)** is defined as the accuracy of chirp symbol decoding on the test dataset, which measures the resilience to the channel noise.
(2) **SNR Gains** is defined as the SNR gap between NELoRa and dechirp at 10% SER on the SER-SNR curves. When SER is higher than 10%, a LoRa packet containing tens of symbols is hard to be correctly decoded even with coding redundancy. Thus, this metric measures the SNR benefit NELoRa can provide.
(3) **Battery Life Gain (BLG)** is defined as the extended LoRa node lifetime of NELoRa over dechirp calculated based on LoRaWAN battery models [10, 14]. This metric measures the energy efficiency of NELoRa. We assume an ideal error correction mechanism to map the expected SER to the appropriate Forward Error Correction (FEC) code length. NELoRa provides a better SNR sensitivity at a gateway. Thus it can provide robust communication with much shorter FEC codes (e.g., a smaller SF configuration) compared with the standard LoRa demodulation method. This leads to a significant decrease in the packet transmission time, which affects the battery life. We estimate the battery life by assuming a LoRa node is powered by two AA batteries and sending 40-byte packets six times per hour.

6 EVALUATION

In this section, we evaluate the performance of NELoRa to answer the following questions.

- **Q1 (§6.1)**: *How much does NELoRa improve the demodulation performance than dechirp under various LoRa configurations?*
- **Q2 (§6.2)**: *How effective is each key technique incorporated in the design of NELoRa?*
- **Q3 (§6.3)**: *Is NELoRa robust to different environments for low-cost deployment?*
- **Q4 (§6.4)**: *What is the storage overhead and running time of NELoRa?*
- **Q5 (§6.5)**: *What is the performance of NELoRa in outdoor environment?*

6.1 Overall Performance

Setup: We evaluate the performance of NELoRa with various LoRa configurations, including 4 SFs (e.g., 7, 8, 9, 10) and 3 BWs (e.g., 125K, 250K, 500K).

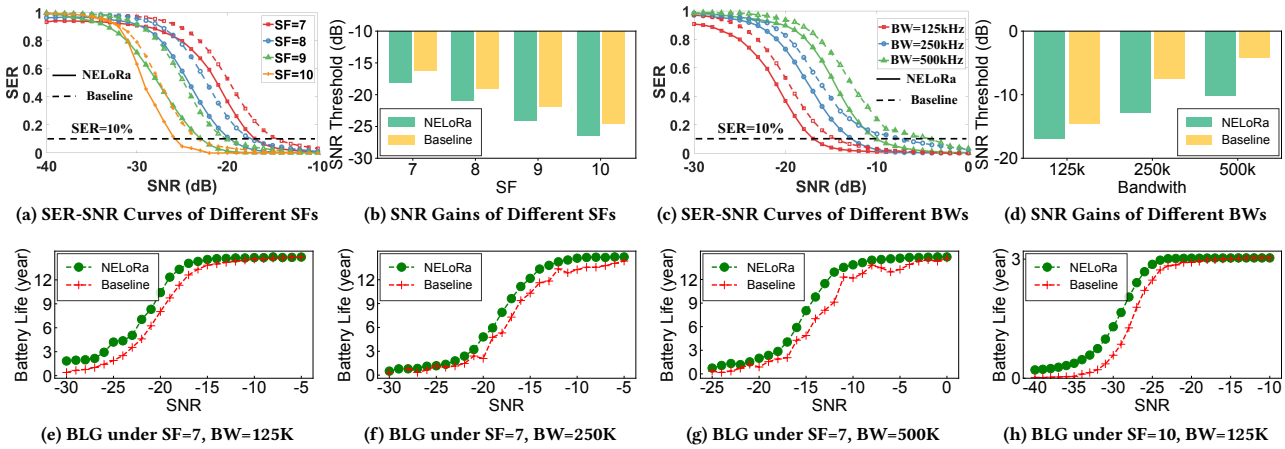
Results: The results are shown in Figure 14. Given the BW is 125K, Figure 14a and b show the impact of different SFs on SER and SNR threshold, respectively. We can observe that NELoRa (e.g., solid line) has obtained consistently lower SER than dechirp (e.g., dashed line) for SFs from 7 to 10 across all SNR levels. For different SFs, the SNR gain is ranging from 1.84 dB (e.g., SF=8) to 2.35 dB (e.g., SF=7) The SNR threshold of NELoRa at an SF can almost catch up with dechirp at a higher SF. For example, the SNR threshold of NELoRa at SF=7 is close to dechirp's at SF=8. Then, we study the trend of SER and SNR threshold as BW changes when SF is set to 7. As shown in Figure 14c and d, it demonstrates similar SNR threshold improvement and SNR gains under different BWs. In all evaluated LoRa configurations, the largest SNR gain is 5.94 dB under SF=7 and BW=500K. Additionally, we can also find the SER of NELoRa cannot reach 100% even at ultra-low SNR, such as the SF=10 at SNR=-40 dB. The results verify the efficiency of our DNN demodulator in ultra-low SNR. And multi-dimensional pattern features are successfully abstracted during the model training process with millions of chirp symbols. Our DNN model can be further refined as more diverse chirp symbols are used for training.

We further evaluate the battery life for NELoRa and the baseline (i.e., dechirp) at different SNR levels and LoRa configurations, as shown in Figure 14e-h. And we can see all the green lines for NELoRa are higher than the red ones, resulting a consistent BLG. In Figure 14e-g, we can see different BWs for SF=7 have a comparable battery life when the SNR is higher than -10 dB. Additionally, when the SNR is getting lower, NELoRa outperforms the baseline consistently, especially can provide a higher BLG for a smaller BW at a ultra-low SNR-level. As the SF increases to 10, Figure 14h shows, the battery life is less than three years, much smaller than that using SF=7 for high energy consumption using a longer chirp symbol. However, NELoRa achieves a higher BLG with SF=10 than SF=7. Statistically, the average BLG under SF=7, BW=125K to 500K are 1.39, 1.42 and 1.51 years for SNR$\in$[-25, -10] dB. Compared to the baseline, NELoRa extends the median battery life by 27%, 33% and 76%, respectively. Moreover, the median BLG under SF=10 can reach 0.38 years (SNR$\in$[-40,-18] dB), equivalent to 272% of the baseline in battery life. The BLG is increasing as the SNR decreases. The reason is that with a lower SNR level, a higher SF is needed in dechirp, but our SNR gains removes the more energy consuming SF change.

6.2 Component-wise Analysis

Setup: We evaluate the efficiency of NELoRa's sequential DNN noise filter and decoder on LoRa demodulation. With chirp symbols (SF=7, BW=125K), we train and test our DNN model by disabling certain modules separately. For example, we only run the DNN noise filter to mask the raw dual-channel spectrogram and then convert it back to the chirp symbol, on which we adopt dechirp for decoding, named as the **w/o Decoder**. Besides, we demodulate the raw dual-channel spectrogram directly via our DNN decoder without the DNN noise filter, labeled as **w/o Filter**. We compare NELoRa with **w/o Decoder**, **w/o Filter** and the baseline (i.e., dechirp).

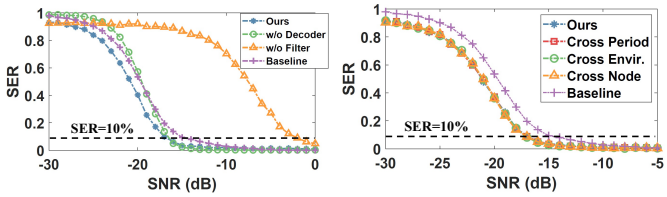
Results: Illustrated in Figure 15a, NELoRa can lower the SNR threshold from -14 dB of the baseline to -17 dB, when the SER is 10%. Hence, NELoRa brings a 3 dB SNR gain. However, if we use dechirp to replace our DNN decoder, the SER is similar with



(a) SER-SNR Curves of Different SFs (b) SNR Gains of Different SFs (c) SER-SNR Curves of Different BWs (d) SNR Gains of Different BWs

(e) BLG under SF=7, BW=125K (f) BLG under SF=7, BW=250K (g) BLG under SF=7, BW=500K (h) BLG under SF=10, BW=125K

Figure 14: Overall performance of NELoRa under different LoRa configurations. (a) and (c) illustrate the SER-SNR curves under different SFs and BWs. The solid line and dashed line represent the performance of NELoRa and dechirp, respectively. (b) and (d) illustrate the SNR gains under different SFs and BWs. (e)-(h) illustrate the BLG under different SNR and LoRa configurations.



(a) SER Across Components (b) SER Across Domains

Figure 15: (a) Ablation study for the efficiency of the DNN noise filter and decoder; (b) the DNN model can be generalized across different periods, locations, and nodes.

NELoRa when the SNR is higher than -16 dB, but getting worse as the SNR decreases. The reason for this gap is that the DNN noise filter can reduce the noise and induce the spectrum energy loss on the recovered chirp symbol, which is getting larger as long as the SNR is decreasing. The spectrum energy loss will fail dechirp accordingly. This verifies the efficiency of the DNN decoder working on the masked spectrogram. Additionally, if we remove our DNN noise filter and directly apply the DNN decoder on a raw dual-channel spectrogram, the performance is the worst, as the yellow line shows. The SER begins to increase when the SNR is around 0 dB. The reason is that the masked spectrogram used to train the DNN decoder usually has a high SNR. Therefore, the DNN decoder cannot explore the spectrogram patterns distorted by the noise at low SNR levels. This also verifies the efficiency of the DNN noise filter. Hence, only combining the DNN noise filter and decoder together can achieve SNR gains compared to the baseline.

6.3 Cross Domain Transfer-ability Analysis

Setup: According to different settings (e.g., LoRa node, location, period), we split our dataset into several sub-datasets. The setting combinations are different among different sub-datasets. We use a sub-dataset to generate synthesis training data and the others to test the demodulation performance of the pre-trained DNN model without further model re-training. For example, the synthesis training sub-dataset (SF=7, BW=125K) is collected at the node A from room $\hat{A}$ at 7:00pm. Then we will test the pre-trained DNN model on the sub-dataset transmitted by node B at the same location and time (labeled as *Cross Node*), node A at the same location with different periods (labeled as *Cross Period*), and node A at room $\hat{B}$ with the same period (labeled as *Cross Envir*). We also use the DNN model trained by all training datasets and dechirp to evaluate the same testing sub-datasets for comparison.

Results: As shown in Figure 15b, we can see the SER-SNR curves of NELoRa, Cross Period, Cross Envir. and Cross Node are overlapping with each other, demonstrating the same trend, which renders the extra SNR gain of 3 dB at the SER=10% compared to dechirp. This verifies the efficiency of our training data augmentation and high-quality chirp symbol generation to train a unified DNN model. Since all possible masked dual-channel spectrogram inputs have been seen by NELoRa in the training stage, which is agnostic to periods, nodes, or environments. With the finite coding space of CSS modulation in LoRa, we can maximize the benefits of the over-fitting property of DNN at the testing stage.

6.4 Storage Overhead and Running Time

Setup: By running our original DNN model on a local PC with 1080Ti GPU and the compressed version on a single-board computer Raspberry Pi 4, we evaluate its storage overhead and running time of each DNN module. Both the original and compressed DNN models can achieve similar SNR gains on our testing dataset. The running time measures the time used to process 16 chirp symbols and is computed by running 160 times and taking the average.

Results: Table 2 shows the storage overheads of NELoRa's DNN modules under different SFs. We can see that model compression

Table 2: NELoRa's storage overhead before and after compression (BW = 125K, unit: MB)

Module	SF = 7	SF = 8	SF = 9	SF = 10
DNN Filter	23/4.6	36.7/8.24	64.2/15.4	119/29.7
DNN Decoder	9.1/2.7	36.2/7.4	144/42.7	578.8/123.2

Table 3: Time consumption of NELoRa for demodulating a packet with 16 chirp symbols on various platforms using different DNN model (e.g., PC / Raspi / Raspi Compressed)

Config.	Transform	Filter(ms)	Decoder(ms)
SF=7	1.60/21.7/-	5.12/13740/1902	0.47/173/121
SF=8	1.6/38.1/-	6.97/24179/3391	0.48/243/188
SF=9	2.06/47.4/-	9.79/45016/5823	0.46/1803/871
SF=10	1.89/124.2/-	29.8/120032/11237	0.55/10368/4218

reduces the model size by around 70% to 80%. We can also see that the model size of NELoRa increases as SF goes up. This is because the larger SF has longer chirp symbols, resulting in larger models and more parameters to handle the large size input.

The running time of NELoRa are shown in Table 3. Similar to the storage overhead, the running time increases as SF goes up. Also, when NELoRa is running on a PC equipped with GPU, incurring 7.19ms when SF=7 for demodulating 16 chirp symbols. The time consumption is much larger when running on Raspi than on GPU since Raspi is a resource-constrained platform. Specifically, it incurs 13934.7ms under SF=7. After model compression, it incurs 2044.7ms, achieving 6.8× speedup. In terms of larger SF (e.g., SF=10), the total processing latency of 16 chirp symbols is up to 7 seconds. In the future, we can further compress our DNN model to boost the running efficiency.

6.5 Campus-scale Testbed Experiments

Setup: Figure 16 illustrates the deployment of our campus-scale ($1700m \times 1200m$) testbed, covering various landcover types (e.g., trees, buildings, roads, and pond). We deploy the LoRa nodes at six locations. Location 1 is the closest, and location 5 is the farthest one. We set SF and BW as seven and 125K. Each LoRa node transmits 15 packets, and each contains 188 chirp symbols. We run our LoRa packet detection method to capture these packets and generate chirp symbols as the input of our DNN demodulator. We first test pre-trained NELoRa directly. Since the pre-trained NELoRa only witness the artifact Gaussian white noises in the environment of the indoor testbed, we further divide these new chirp symbols into training and testing ones. Then, we use the training chirp symbols to fine-tune our DNN model to achieve higher performance regarding our campus environment.

Results: First of all, all 90 packets transmitted in the six locations are detected, which verifies the efficiency of our packet detection method. Then, Figure 17 illustrates that the SER increases as the distance increases between the gateway and the LoRa node. Although location 2 is near to the gateway, it has a high SER due to a low SNR level incurred by the blockage of buildings. Compared with the baseline, the original NELoRa decreases SER by 5.51% to

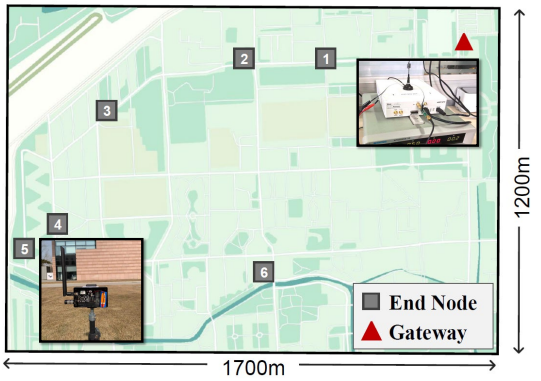
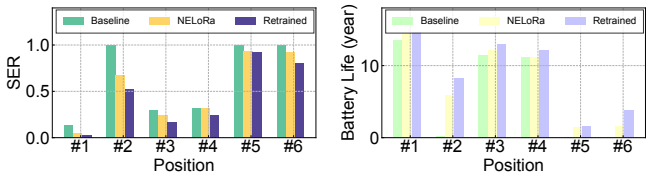


Figure 16: The illustration of our outdoor testbed and the topology of the LoRa nodes and NELoRa gateway.



(a) SER at different locations **(b) BLG at different locations**

Figure 17: SER and BLG performance at six different locations on our campus-scale testbed.

31.9%, and the re-trained NELoRa further reduces the SER by 7.72% to 46.9%. We can see the re-trained model is a little better than the original one by capturing more unseen noise patterns (e.g., multi-path noise) and reducing the spectrum energy loss of the DNN noise filter. We can further improve NELoRa by updating in an online and distributed manner (e.g., federated learning [24, 34, 36]) as more packets are collected across multiple gateways.

We estimate the battery life of the operating LoRa node at each location. Figure 17b shows the battery life can be extended by 1.32 to 5.73 years with the original NELoRa. The maximum BLG can reach 8.06 years by using the re-training NELoRa at location 2. The SER of location 2, 5, and 6 reach 100% using the dechirp. The gateway cannot demodulate any data from those locations even the LoRa nodes drain out their battery. With NELoRa, the SER is lowered, and the battery life is increased significantly.

When we increase the SF (e.g., 8-10), with an increasing SNR threshold, the SER of dechirp will be reduced at the same position. Thus, the SNR gains of NELoRa is lower than that under SF=7. However, when we enlarge our deployment range, we can observe consistent SNR gains as we have observed in our indoor testbed.

7 RELATED WORK

LoRa Range and Energy Enhancement: Inspired by using multiple antennas (MIMO) to improve SNR in Wi-Fi and cellular communication, recent studies [2, 10, 12] bring the diversity gains of distributed MIMO to LoRa. At the transmitter side, Choir [12] leverages up to 36 co-located LoRa nodes to boost the received signal strength. At the receiver side, Charm [10] coordinates multiple

gateways to decode weak signals undecodable at any individual gateway by detecting the combined energy peak in the spectrum. Additionally, Chime [14] eliminates the multi-path interference by frequency selection to capture extra SNR gains for LoRa transmissions. All require multiple pairs of transceivers and are built on dechirp demodulation method. Nephalai [30] further proposes to demodulate compressed PHY samples in the cloud with the sparse approximation. In contrast, NELoRa utilizes the deep learning to obtain extra SNR gains for a single pair of transceivers. Thus, it can supplement existing works and be further enhanced with the diversity gains of distributed MIMO.

LoRa Throughput and Deployment Study: FTrack [49], mLoRa [48], CoLoRa [45], NScale [44], and SCLoRa [21] propose various methods to resolve the LoRa packet collision, improving LoRa's throughput. Besides, NetScatter [19] designs a distributed CSS coding for hundreds of strictly synchronized concurrent transmissions. Some other studies [8, 22, 27, 28, 50] deploy real-world LoRaWAN to study the deployment, measurement, and localization problems. In contrast, NELoRa develops a neural-enhanced demodulator to enlarge a LoRa node's communication range and optimize energy usage, parallel to studies in this category.

Deep Learning based Wireless Communication: Instead of physical model-driven approaches, data-driven deep learning techniques have been used to optimize wireless communication systems [25], such as the coding [16, 23, 35, 43] and decoding [11, 37, 47] mechanisms. And the former learn the coding structure of signals and decode the data bits, including polar codes [16], convolutional codes [23], turbo codes, hamming [43], and LDPC codes [35]. In contrast, the latter covers the decoding of DQPSK signals [11]. However, existing deep learning methods cannot be directly adopted in LoRa since they require the signals are above the noise floor. Therefore, we develop a new learning component to combat various noises and enable ultra-low SNR communication under the noise floor. Additionally, we utilize augmented learning to generalize our DNN model with low overhead.

For LoRa communication, only a few works have proposed deep learning based approaches recently. Specifically, DeepLoRa [31] utilizes Bi-LSTM DNN to develop a land-cover aware path loss model and reduces the estimation error to less than 4 dB, which is 2× smaller than state-of-the-art [9] for link estimation. DeepSense [6] further explores the deep learning augmented random access in the coexistence of LPWANs, even below the noise floor (e.g., -10dB). In comparison with DeepLoRa and DeepSense, NELoRa targets a different task, LoRa demodulation. We first show a general DNN demodulator can achieve lower SNR threshold than dechirp does with only affordable computation overhead at the gateway side.

Moreover, TinySDR [20] delivers FPGA-based hardware, which can boost the research on AI-augmented LoRa networks by supporting deep AI algorithms on-board. Given the strong feature learning ability of DNN, we believe the AI-augmentation brings a new direction to design a more efficient communication stack and sensing technique in LoRa networking.

8 DISCUSSION AND OPEN ISSUES

DNN Model Optimization: The maximum SF we evaluated is 10 and a commodity LoRa node can support two larger SFs, namely 11 and 12. With the increasing SF, the chirp symbol length and types exponentially increase. As a result, we must use much more computation resources (i.e., multi-GPU cluster) to train the DNN model. According to the consistent SNR gains observed when SF increases from 7 to 10, similar SNR gains can be expected when SF is 11 or 12 as long as the DNN model is trainable. To reduce the model complexity and converge time, we can further optimize the structure of our DNN model and loss function design.

End-to-end Neural-enhanced LoRa Demodulation: NELoRa focuses on decoding LoRa chirp symbols with a DNN model. And it requires extra modules for packet detection and chirp symbol generation via signal processing, which can also be substituted by DNN. For example, DeepSense [6] delivers the neural-enhanced carrier sense for efficient channel access in the coexistence of LPWANs, even under the noise floor. Given the LoRa CSS modulation, a holistic neural-enhanced demodulation method is still an open issue in our future work. Besides, as AI chipsets and hardware-enabled platforms [20], most edge devices will be equipped with machine intelligence to support the deployment of NELoRa.

Online DNN Model Adaptation: NELoRa's performance can be lowered by the unseen noise patterns in a new environment, shown in Figure 17. Given widely deployed LoRa nodes, gateways and massive collected data, a promising optimization is to update NELoRa in an online manner.

9 CONCLUSION

In this paper, we present the design, implementation, and evaluation of NELoRa, a neural-enhanced LoRa demodulation system that breaks the SNR threshold of the standard dechirp approach. The SNR gains enable longer communication distance and battery lifetime in LoRa. NELoRa consists of several essential techniques. First, we develop a new dual-channel spectrogram to maximum the temporal-spatial feature space in our DNN model. Second, we design a novel DNN model containing a mask-enabled DNN noise filter and a spectrogram-based DNN decoder to fit the boundary among different LoRa symbols in the feature space. With the finite coding space of LoRa, the training and testing datasets have the identical distribution of LoRa chirp symbols in the feature space. The DNN's over-fitting is explored to optimize our model training. Finally, we generate massive synthesis data, covering various noise to train a general DNN model. To further make NELoRa practical, we compress our DNN model and develop an end-to-end LoRa demodulation method by combining the methods of LoRa packet detection and hardware offset cancellation. We implement NELoRa and conduct extensive experiments to evaluate its performance. The results show that NELoRa obtains 1.84-2.35 dB SNR gains and a largest 272% (~0.38-1.51 years) battery lifetime gain in average across different configurations and environments.

10 ACKNOWLEDGEMENT

We sincerely thank the anonymous reviewers and our shepherd for their valuable feedback. This work was supported by NSF Awards CNS-1909177, CNS-1617627, CNS-1814551, CNS-1950171, CNS-1949753, CNS-1617412, CCF-2007159 and PFI:BIC-1632051.

REFERENCES

[1] LoRa Alliance. Retrieved by Nov 19th 2020. A technical overview of LoRa and LoRaWAN. In *https://lora-alliance.org/resource-hub/what-lorawanr*.

[2] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. 2020. A cloud-optimized link layer for low-power wide-area networks. In *Proceedings of ACM MobiSys*.

[3] Albert Berni and WO Gregg. 1973. On the utility of chirp modulation for digital signaling. *IEEE Transactions on Communications* (1973).

[4] Chao Cai, Zhe Chen, Henglin Pu, Liyuan Ye, Menglan Hu, and Jun Luo. 2020. AcuTe: Acoustic Thermometer Empowered by a Single Smartphone. In *Proceedings of ACM SenSys*.

[5] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi. 2016. Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications* (2016).

[6] Justin Chan, Anran Wang, Arvind Krishnamurthy, and Shyamnath Gollakota. 2019. DeepSense: Enabling Carrier Sense in Low-Power Wide Area Networks Using Deep Learning. *arXiv:1904.10607 [cs]* (2019).

[7] TACM Claasen and Wolfgang Mecklenbräuker. 1980. The Wigner distribution—A tool for time-frequency signal analysis—Part II: Discrete time signals. *Philips Research* (1980).

[8] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. 2019. Automated estimation of link quality for LoRa: a remote sensing approach. In *Proceedings of ACM/IEEE IPSN*.

[9] S. Demetri, M. Zúñiga, G. P. Picco, F. Kuipers, L. Bruzzone, and T. Telkamp. 2019. Automated Estimation of Link Quality for LoRa: A Remote Sensing Approach. In *Proceedings of ACM/IEEE IPSN*.

[10] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Charm: Exploiting Geographical Diversity through Coherent Combining in Low-Power Wide-Area Networks. In *Proceedings of ACM/IEEE IPSN*.

[11] Sebastian Dörner, Sebastian Cammerer, Jakob Hoydis, and Stephan Ten Brink. 2017. Deep learning based communication over the air. *IEEE Journal of Selected Topics in Signal Processing* (2017).

[12] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering Low-Power Wide Area Networks in Urban Settings. In *Proceedings of ACM SIGCOMM*.

[13] Biyi Fang, Xiao Zeng, and Mi Zhang. 2018. NestDNN: Resource-Aware Multi-Tenant On-Device Deep Learning for Continuous Mobile Vision. In *Proceedings of ACM MobiCom*.

[14] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. 2020. Frequency Configuration for Low-Power Wide-Area Networks in a Heartbeat. In *Proceedings of USENIX NSDI*.

[15] Akshay Gadre, Fan Yi, Anthony Rowe, Bob Iannucci, and Swarun Kumar. 2020. Quick (and Dirty) Aggregate Queries on Low-Power WANs. In *Proceedings of ACM/IEEE IPSN*.

[16] Tobias Gruber, Sebastian Cammerer, Jakob Hoydis, and Stephan ten Brink. 2017. On deep learning-based channel decoding. In *Proceedings of IEEE Conference on Information Sciences and Systems (CISS)*.

[17] Xiuzhen Guo, Longfei Shangguan, Yuan He, Jia Zhang, Haotian Jiang, Awais Ahmad Siddiqi, and Yunhao Liu. 2020. Aloba: rethinking ON-OFF keying modulation for ambient LoRa backscatter. In *Proceedings of ACM SenSys*. ACM.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE CVPR*.

[19] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. 2019. NetScatter: Enabling Large-Scale Backscatter Networks. In *Proceedings of USENIX NSDI*.

[20] Mehrdad Hessar, Ali Najafi, Vikram Iyer, and Shyamnath Gollakota. 2020. TinySDR: Low-Power SDR Platform for Over-the-Air Programmable IoT Testbeds. In *Proceedings of USENIX NSDI*.

[21] Bin Hu, Zhimeng Yin, Shuai Wang, Shuai Wang, Zhuqing Xu, and Tian He. 2020. SCLoRa: Leveraging Multi-Dimensionality in Decoding Collided LoRa Transmissions. In *Proceedings of IEEE ICNP*.

[22] Oana Iova, Amy Murphy, Gian Pietro Picco, Lorenzo Ghiro, Davide Molteni, Federico Ossi, and Francesca Cagnacci. 2017. LoRa from the city to the mountains: Exploration of hardware and environmental factors. In *Proceedings of EWSN*.

[23] Hyeji Kim, Yihan Jiang, Ranvir Rana, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. 2018. Communication algorithms via deep learning. *arXiv preprint arXiv:1805.09317* (2018).

[24] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient Federated Learning via Guided Participant Selection. In *Proceedings of USENIX OSDI*.

[25] Chenning Li, Zhichao Cao, and Yunhao Liu. 2020. Deep AI Enabled Ubiquitous Wireless Sensing: A Survey. *ACM Computing Surveys (CSUR)* (2020).

[26] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* (2016).

[27] Jansen C Liando, Amalinda Gamage, Agustinus W Tengourtius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks* (2019).

[28] Yuxiang Lin, Wei Dong, Yi Gao, and Tao Gu. 2020. SateLoc: A Virtual Fingerprinting Approach to Outdoor LoRa Localization using Satellite Images. In *Proceedings of ACM/IEEE IPSN*.

[29] Daibo Liu, Zhichao Cao, Mengshu Hou, Huigui Rong, and Hongbo Jiang. 2020. Pushing the limits of transmission concurrency for low power wireless networks. *ACM Transactions on Sensor Networks* (2020).

[30] Jun Liu, Weitao Xu, Sanjay Jha, and Wen Hu. 2020. Nephalai: towards LPWAN C-RAN with physical layer compression. In *Proceedings of ACM MobiCom*.

[31] Li Liu, Yuguang Yao, Zhichao Cao, and Mi Zhang. 2021. DeepLoRa: Learning Accurate Path Loss Model for Long Distance Links in LPWAN. In *Proceedings of IEEE INFOCOM*.

[32] Qiang Ma, Zhichao Cao, Wei Gong, and Xiaolong Zheng. 2021. BOND: Exploring Hidden Bottleneck Nodes in Large-scale Wireless Sensor Networks. *ACM Transactions on Sensor Networks* (2021).

[33] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* (2008).

[34] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR*.

[35] Eliya Nachmani, Elad Marciano, Loren Lugosch, Warren J Gross, David Burshtein, and Yair Be'ery. 2018. Deep learning methods for improved decoding of linear codes. *IEEE Journal of Selected Topics in Signal Processing* (2018).

[36] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. 2021. ClusterFL: A Similarity-Aware Federated Learning System for Human Activity Recognition. In *Proceedings of ACM MobiSys*.

[37] Timothy J O'Shea and Jakob Hoydis. 2017. An introduction to machine learning communications systems. *arXiv preprint arXiv:1702.00832* (2017).

[38] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. 2018. PLoRa: a passive long-range data network from ambient LoRa transmissions. In *Proceedings of ACM SIGCOMM*.

[39] Edmundo Pereira de Souza Neto, Marc-Antoine Custaud, Jean Frutoso, Laurence Somody, Claude Gharib, and Jacques-Olivier Fortrat. 2001. Smoothed pseudo Wigner–Ville distribution as an alternative to Fourier transform in rats. *Autonomic Neuroscience* (2001).

[40] ABI Research. Retrieved by Nov 19th 2020. NB-IoT and LTE-M Issues to Boost LoRa and Sigfox Near and Long-term Lead in LPWA Network Connections. In *https://tinyurl.com/2026- cellular-iot*.

[41] Dilip Roy. 2004. Discrete rayleigh distribution. *IEEE Transactions on Reliability* (2004).

[42] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. 2021. Concurrent interference cancellation: decoding multi-packet collisions in LoRa. In *Proceedings of ACM SIGCOMM*.

[43] LG Tallini and P Cull. 1995. Neural nets for decoding error-correcting codes. In *Proceedings of IEEE Technical applications conference and workshops*.

[44] Shuai Tong, Jiliang Wang, and Yunhao Liu. 2020. Combating packet collisions using non-stationary signal scaling in LPWANs. In *Proceedings of ACM MobiSys*.

[45] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. 2020. CoLoRa: Enabling Multi-Packet Reception in LoRa. In *Proceedings of IEEE INFOCOM*.

[46] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.

[47] Tianqi Wang, Chao-Kai Wen, Hanqing Wang, Feifei Gao, Tao Jiang, and Shi Jin. 2017. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications* (2017).

[48] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. 2019. mLoRa: A Multi-Packet Reception Protocol in LoRa networks. In *Proceedings of IEEE ICNP*.

[49] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: parallel decoding for LoRa transmissions. In *Proceedings of ACM SenSys*.

[50] Yuguang Yao, Zijun Ma, and Zhichao Cao. 2019. LoSee: Long-Range Shared Bike Communication System Based on LoRaWAN Protocol.. In *Proceedings of EWSN*.

[51] Hyunho Yeo, Chan Ju Chong, Youngmok Jung, and Juncheol Ye. 2020. NEMO: enabling neural-enhanced video streaming on commodity mobile devices. In *Proceedings of ACM MobiCom*.

[52] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. 2018. Neural Adaptive Content-aware Internet Video Delivery. In *Proceedings of USENIX OSDI*.

[53] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing* (2017).

[54] Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2018. FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. *IEEE Transactions on Image Processing* (2018).

[55] Mi Zhang, Faen Zhang, Nicholas D Lane, Yuanchao Shu, Xiao Zeng, Biyi Fang, Shen Yan, and Hui Xu. 2020. Deep Learning in the Era of Edge Computing: Challenges and Opportunities. *Fog Computing: Theory and Practice* (2020).

[56] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. 2017. Learning Sleep Stages from Radio Signals - A Conditional Adversarial Architecture. In *Proceedings on ICML*.