The Journal of Supercomputing (2023) 79:17079–17097
https://doi.org/10.1007/s11227-023-05355-4

LoRa DL: a deep learning model for enhancing the data transmission over LoRa using autoencoder

B. Shilpa[1] · Puranam Revanth Kumar[1] · Rajesh Kumar Jha[1]

Accepted: 26 April 2023 / Published online: 5 May 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Low Power Wide Area Networks (LPWANs) have become a popular option for modern wireless communication technologies. Long Range (LoRa) protocol was designed for LPWAN, which offers long-distance communication, low-power consumption, and simultaneous transmissions. Long-distance communication necessitates an extremely low signal-to-noise ratio at the receiver. Additionally, low power consumption necessitates less signaling, which leads to the usage of simpler protocols like ALOHA and less coordinated communication. Therefore, as the number of devices equipped with this technology grows, its performance will degrade naturally as a result of scalability and interference difficulties. Deep learning holds great promise for resolving these problems through data-driven approaches and enhances the efficiency of LoRaWAN in usage of limited spectrum resources. In this work, we present the design of an end-to-end communication system as an autoencoder architecture utilizing deep learning. This adaptable architecture is able to efficiently capture channel impairments while simultaneously optimizing the operations of the transmitter and receiver together. The autoencoder is designed with multiple system model elements, including a sender net that simulates a LoRa transmitter and modulates data, a channel net that models channel impairments, and a receiver net that acts as a LoRa receiver to demodulate and retrieve the original data. The proposed autoencoder model is trained and evaluated with LoRa samples generated by simulation, and it is shown to be a high performer by comparing with the traditional LoRaWAN in terms of Bit Error Rate (BER) and Packet Success Rate (PSR) measurements.

Keywords Autoencoder · Deep learning · Internet of Things · LoRa · Wireless Communication

✉ B. Shilpa
b.shilpa@ifheindia.org

[1] Department of Electronics and Communication Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad, India

1 Introduction

The most popular connection choice for Internet of Things (IoT) applications needing extended range, great energy efficiency, and low cost has emerged as Low Power Wide Area Networks (LPWAN). The goal of LPWAN protocols is to connect a large number of battery-operated devices, such as sensors used in applications that can tolerate delays and have modest throughput requirements per device [1]. A significant number of LPWAN technologies uses unlicensed frequencies, which presents an extra obstacle on account of the increased interference levels. Effective data transmission is essential for long-range communications and improved quality of service, which increases reliability. Consequently, reducing the need for signal retransmission, which entails spectrum congestion. Spectrum efficacy is especially critical in the unlicensed Industrial, Scientific, and Medical (ISM) band, because this spectrum is open to the public.

The goal of every communication system is to accurately recreate the message sent by the transmitter at the other end of the channel. The transmitted signal may get distorted as a result of the channel's noise characteristics, and the receiver may not be able to reconstruct the message exactly. A reliable communication system ought to be capable to overcome these channel-induced corruptions and faithfully reconstruct the message at the recipient. Wireless communication systems have traditionally been built using accurate system models and complicated mathematical theories. The complex relationship between spectrum data and communication architecture of new generation wireless communication technologies is difficult to represent using traditional modeling approaches. Machine Learning (ML) and Deep Learning (DL) technologies are grown rapidly in recent years, with applications extending into practically every sector and academic domain. The nonlinearity resistance of DL algorithms has made them a topic of intense study for implementation in wireless communication systems [2]. DL has emerged as a practical approach to address the challenges associated with the wireless communication system's need for increased data rate, speed, reliability, and security. Utilizing multiple DL classification approaches [3–5] enables its usage for modulation recognition. Equalization and sampling synchronization with DL has been investigated in [6, 7].

Multiple functional blocks are utilized to construct links in traditional mathematically derived models of communication systems. Every functional block is optimized separately to get maximum efficiency. As an alternative, the DL-based autoencoder architecture offers a comprehensive answer, one that is wholly data-driven and focused on learning from end-to-end in order to achieve optimal performance. This research focused on the end-to-end design of autoencoder-based communication systems. The idea of training a wireless communications network to act as an autoencoder was first proposed in [8]. An autoencoder is one kind of a Deep Neural Network (DNN) that features both an encoder and a decoder. It is the responsibility of the encoder to discover an underlying representation of the input data, and the decoder's task is to recover the original data from the encoding. Instead of optimizing the transmitter and receiver independently, this is accomplished by designing the two major components as an autoencoder. The authors in [9] presented a training method for end-to-end systems without knowing about the channel by considering perturbation stochastic approximations. In [10], another method based on transmitter output perturbations is suggested. [11, 12] provide techniques for utilizing neural networks to approximate the channel distribution and then use that as a substitute channel for backpropagation.

Long Range (LoRa) is a widely used LPWAN modulation technology that supports numerous IoT applications, including smart metering, smart agriculture, smart buildings, and logistics monitoring [13]. LoRa is a low-power, long-range communication standard implemented at the physical layer. LoRa's ability to provide compromises between power consumption, communication range, and data rate enables sustainable IoT. The physical layer of LoRaWAN allows for the orthogonalization of transmissions in terms of channel frequency, bandwidth, coding rate, transmission power, and spreading factor, enabling the deployment of LoRa nodes in high density. However, it also faces several challenges in terms of scalability, capacity, interference and so on. Most LoRa-based IoT applications nowadays are used to create massive network maps consisting of millions of nodes. This creates a problem for LoRa network densification. Evaluating an application on such a massive scale necessitates the use of advanced simulation and modeling techniques.

The following is a list of the contributions made in this work:

- Designing a DL-based autoencoder for end-to end communication of LoRaWAN.
- Implementing effective channel model and training mechanism.
- Performance evaluation of the model with simulation data.

The remainder of the article is organized as follows: The second section discusses the fundamentals of LoRa technology and how DL can be implemented in LPWANs. The autoencoder architecture and system model are presented in Sect. 3. Section 4 goes into detail about implementation, including two-phase training. The results of the simulation are presented in Sect. 5, and the article is concluded in Sect. 6.

2 LoRa and deep learning

LoRa is a physical layer acquired and developed by Semtech corporation, with higher-level properties for implementing LPWANs. LoRa's base adjustment is Chirp Spread Spectrum (CSS). It has similar low-power characteristics as Frequency Shift Keying, but apparently improves contact range. Multiple IoT nodes in LoRa network can be connected with a network server via gateways using a virtual channel. There are six virtual channels called Spreading Factors (SFs), ranging from 7 to 12 are used for communication between LoRa nodes and gateway. LoRa data rate varies between 300 bits and 50,000 bits per second relative to the selection of SF and bandwidth. A LoRa receiver can decode signals that are 19.5 decibels below the noise floor, allowing for extremely extended communication ranges. The possible communication range of LoRa is 15 km in rural and 5 km in urban areas [14]. Depending on the network environment and the number of LoRa nodes served by a LoRa gateway, the parameters of the network can be optimized to achieve better efficiency. The primary parameters of LoRa that can be optimized for better results are SF, bandwidth, coding rate, carrier frequency, and transmission power.

In addition to the LoRa radio protocol, the LoRa Alliance has developed the LoRaWAN protocol stack and network architecture [15]. LoRaWAN is a MAC layer protocol with open access which operates on regional ISM band frequencies of 433 MHz–868 MHz (EU), 915 MHz (AUS and US), and 923 MHz (ASIA). The LoRaWAN MAC layer determines the network architecture that in this respect is a star of stars. The LoRaWAN network architecture depicted in Fig. 1 consists of three functional devices, including LoRa nodes, gateways, and a network server. There are three classes of LoRa nodes (A, B, and C), with output varying according to the data rates and latencies made accessible by the network and, therefore, resulting in longer battery life. Gateways function as transparent bridging between LoRa nodes and the network server. Typically, communications to the network server are implemented through secured TCP/IP lines, application data are encoded from LoRa nodes, and gateways do not impose higher-level protocols because they are only pass-through tools. Before arriving at the application point, packets are inspected at gateways and any duplicates are discarded. The network server is the place where application real getting intent is performed.

LoRa communication technology has several advantages over other communication technologies, including its long-range capabilities, low power consumption, and low cost. However, as the demand for IoT applications and devices increases, LoRa communication faces several challenges that can affect its performance and scalability. Scalability is a significant challenge for LoRa communication, as the number of IoT devices and applications continues to grow. With an increasing number of devices transmitting data, the network can become congested, resulting in reduced data rates and increased latency. Additionally, LoRa communication is susceptible to interference from other devices operating in the same frequency band, which can further impact network performance. To address these challenges, the usage of DL has become increasingly important to maintain network performance.

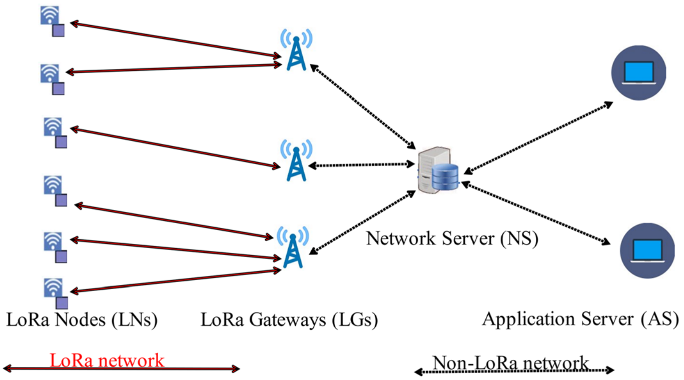


Fig. 1 LoRaWAN network architecture

The performance of LoRa systems can be highly dependent on the channel conditions, such as distance, location, and signal strength. The proposed autoencoder can be used to model the LoRaWAN channel accurately and learn the optimal mapping between the transmitted signal and the received signal under different channel conditions. This can help improve the accuracy of signal detection and interference mitigation algorithms. Additionally, DL-based LoRa autoencoder can also reduce the energy consumption and transmission time of the LoRaWAN by compressing the sensory data need to be transmitted [16].

2.1 Deep learning in LPWANs

IoT and Smart City implementations are producing massive volumes of sensory data that must be analyzed. Research in this area focuses on finding ways to apply DL to these different fields. DNNs are high-rated efficient DL algorithms. They are distinguished from traditional neural networks by their reliance on nonlinear processing neurons for feature extraction. They conceal a multitude of layers, each of which contributes to the accurate modeling of more intricate functions. DNNs often handle nonlinear data properties, which are helpful in big data use cases. DNNs are used to recognize the radio's one-of-a-kind hardware impairments and patterns. This is accomplished by studying a substantial quantity of the raw data that is released by the radios. Typically, IoT networks have a high density of devices, making it ideal for utilizing DL approaches in these networks to improve network performance, authentication, security, and service accessibility.

DL is used in multiple ways to address the challenges of LoRaWAN. One of the primary challenges with LoRaWAN communication is the need for efficient and reliable data transmission. A DL-based approach [17] is used to predict the optimal transmission parameters, such as the SF and data rate, based on the characteristics of the environment and the data traffic. This can help to minimize collisions and reduce energy consumption, resulting in improved performance and longer battery life for LoRaWAN devices. DL is also used for interference detection and mitigation in LoRaWAN communication by designing a model to analyze the received signal and detect patterns that indicate interference [18]. In addition to these applications, DL also used for data compression, feature extraction, and anomaly detection in LoRaWAN communication [19]. These techniques can help to reduce the amount of data that needs to be transmitted, improve data transmission rates, and detect abnormal events or behaviors in the network.

Various DL algorithms are employed in LoRaWAN; however, the autoencoder principle from DL is rarely used. Autoencoder is a DL technique that can be used for unsupervised feature learning and data compression. A compressed version of the input data is created by the encoder network, and the decoder network then restores the original data. Recent works have utilized the autoencoder concept for LoRaWAN localization [20] and demodulation [21] applications. According to our knowledge, this is the first attempt to design an end-to-end communication system for LoRaWAN using a DL-based autoencoder.

3 System model

The proposed model aims to provide end-to-end communication in a LoRa network through the use of an autoencoder. Figure 2 depicts a block diagram of the system. Sensor Nodes (SNs) connected to LoRa nodes (LNs), LoRa Gateway (LG), and Network Server (NS) make up the system. The SNs gather sensory data from IoT devices. The LN is responsible for compressing and transmitting sensory data to the LG. The LG receives and transmits the same to the NS. The transmitted data is decompressed at the NS and forwarded to the AS. Autoencoders are utilized in the design of communication system to simultaneously optimize sender and receiver modules, which is a wholly novel approach for constructing communication systems. The system employs the DNN model to compress sensory data. Encoder and decoder of DNN perform compression and decompression operations at LN and NS, respectively. The system employs sender net as an encoder for compression of sensory data and channel net to simulate LoRa channel characteristics. At the NS, the receiver net decompresses and retrieves the original data.

3.1 Design of autoencoder

Autoencoders are a type of neural network that is programmed to learn a compressed version of raw data and then use that information to reconstruct the input at the output. Developing strong representations for the message at each layer is necessary for the network to ensure reliable transmission. The proposed autoencoder model architecture is presented in Fig. 3. Consider the data to be transmitted from the sender to be $s \in S$. To send data over a noisy channel, modern communication systems transform the data s into a representation $x \in X$. At the destination, a corrupted form of x, denoted by y, is received. The receiver makes an effort to recreate s from y as accurately as feasible. A transmitter is modeled as a function that takes input data s and generates an intermediate representation x using the formula $x = f(s)$. The channel that corrupts x is denoted by $y = h(x)$. Here, the input x is transformed into the desired output y using the stochastic gradient function $h(x)$. Lastly, the receiver can be thought of as another function that uses $\hat{s} = g(y)$ to figure out the best way to reconstruct $\hat{s}$ from y.

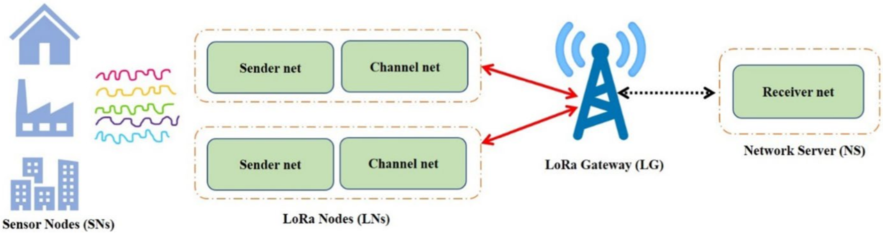


Fig. 2 Illustration of the system model

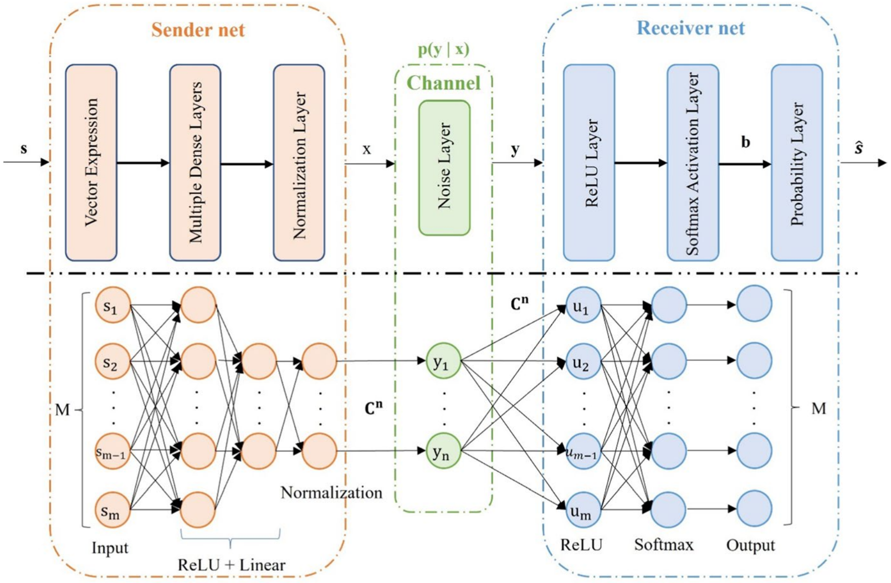


Fig. 3 Illustration of proposed model architecture

3.1.1 Sender net

The sender net is composed of a feed-forward neural network that has dense layers, which is then followed by a layer that is used for normalization. Every dense layer in a neural network is a deep-connected layer, indicating that each neuron in it receives input from all neurons in the preceding layer. Matrix–vector multiplication is carried out in the background by the dense layer. Back propagation can be used to learn and update the values of the real parameters that are represented as variables in the matrix.

LoRa employs CSS modulation, in which a frequency-modulated pulse is used to spread data across the channel's bandwidth. LoRa supports $M = 2^{\mathrm{SF}}$ distinct symbols, each of which contains SF bits. The duration of a symbol is denoted by Ts = MT, where $T = 1/B$ and B represents the bandwidth of the signal. The SF range in LoRa is $\{7, 8, 9, 10, 11, 12\}$ and available bandwidths are 125,250, and 500 kHz. A chirp is used to represent each LoRa symbol, and this chirp has a linear frequency shift that occurs over a period of time denoted by Ts. As input, the sender network receives the symbols from LoRa nodes.

An L-layered feed-forward neural network performs an iterative mapping $f(s_0;\theta)$: $R^{N_0} \rightarrow R^{N_L}$ to transform a vector of inputs $s_0 \in R^{N_0}$ to an output vector $\mathrm{x}_L \in R^{N_L}$ by multiple steps:

$$s_l = f_l\left(s_{l-1};\theta_l\right), \quad l = 1, \ldots, L \tag{1}$$

where $f_l\left(s_{l-1};\theta_l\right)\,:\,R^{N_{l-1}}\rightarrow R^{N_l}$ is the mapping performed at lth layer. This mapping relies on set of parameters θ_l along with an output vector s_{l-1} from the previous layer. The mapping f_l is a function of random variables as it is stochastic. The collection of network parameters is represented with $\theta=\left\{\theta_1,\ldots,\theta_l\right\}$. The lth layer is fully connected layer with

$$f_l\left(s_{l-1};\theta_l\right)=\sigma\left(W_l s_{l-1}+b_l\right) \tag{2}$$

where activation function is given by $W_l\in R^{N_l\times N_{l-1}}$, $b_l\in R^{N_l}$,, and $\sigma\,(\cdot)$. The parameters set of this layer is $\theta_l=\{W_l,b_l\}$. A fresh random mapping is created by all layers with stochastic mappings on every invocation. Normalization is used at the output of the last dense layer to alter the data such that it satisfies the average power restriction or the amplitude constraint imposed by hardware. Finally, the compressed version of input data denoted by x is transmitted via Channel net to the gateway.

3.1.2 Channel net

Once the transmitted samples are given to the LoRa channel, the next operation to be performed is the propagation of the signal through the channel. This involves the signal being affected by path loss, shadowing, multi-path, and noise, which can lead to attenuation, distortion, and interference. The channel also introduces timing errors due to the delay and dispersion of the signal.

The Channel net is designed to simulate the effects of the LoRa channel on the transmitted data. The Channel net designed of a single layer with a Gaussian noise function, where the standard deviation of the noise is determined by the path loss and shadowing effects, and the amplitude and phase of the noise are determined by the fading effect. Each in-phase and quadrature-phase (IQ) sample component is amplified by a genuine Gaussian random variable N in the standard channel known as Additive White Gaussian Noise (AWGN). To simulate the channel effects, a stochastic channel model can be used, which incorporates the LoRa channel features. The most common propagation models used for LoRaWAN systems are the log-normal shadowing model and the Rayleigh fading model. The parameters used to simulate the channel includes the distance between the transmitter and receiver, the frequency of the signal, the gain of the antennas, and the standard deviation of the shadowing and fading models.

The stochastic channel model for LoRa is designed as shown in Fig. 4 with the following components:

1. Upsampling and pulse shaping: In order to accurately reflect the proportion of complex samples to symbols, we set sampling factor $\beta\geq 1$. To begin, we upsample the input symbol vector $x\in C^n$ by appending $\beta-1$ zeros at the end of each symbol. Then, passed through a discrete normalized root-raised cosine filter $g_{\mathrm{rrc}}(t)$. Upsampled and filtered output $g_{\mathrm{rrc}}\in C^{\left(N_{\mathrm{msg}}+L-1\right)}$ contains complex-valued samples of length $N_{\mathrm{msg}}+L-1$.

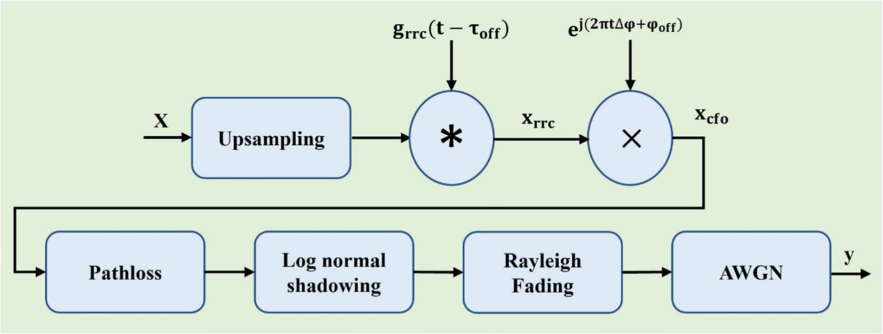


Fig. 4 Representation of channel modeling process

2. Sample offset time: Due to asynchronization of sender and receiver, the sample time has a random offset, indicated by τ_{off}. Since it is anticipated that the receiver would operate directly on IQ-samples received without resorting to any conventional synchronization methods, this timing offset may be modeled during the pulse-shaping stage. The time-shifted filter $g_{\text{rrc}}\left(t-\tau_{\text{off}}\right)$ is used to calculate the convolution of the upsampled input.
3. Phase offset and carrier frequency offset: The carrier frequency offset occurs when the oscillator frequencies of the sender f_{tx} and receiver f_{rx} in radio hardware are slightly out of sync. We describe this as a time-varying rotation impact on the complicated IQ samples.

$$x_{\text{cfo},}k = x_{\text{rrc}}, ke^{j(2\pi t\Delta\varphi+\varphi_{\text{off}})} \tag{3}$$

where f_s is a sampling frequency, $\Delta\varphi = \frac{f_{\text{cfo}}}{f_s}$ is the common-phase offset, and φ_{off} is an optional unknown phase offset.

4. Path loss model: The path loss model is used to simulate the attenuation of the signal as it travels through the channel. The Friis transmission equation [22] is used to model the path loss, which is given by:

$$P_r = P_t * G_t * G_r * \left(\frac{\lambda}{4\pi d}\right)^2 \tag{4}$$

   where P_r is the received power, P_t is the transmitted power, G_t and G_r are the gains of the transmitting and receiving antennas, λ is the wavelength of the signal, and d is the distance between the transmitter and receiver.

5. Shadowing model: The shadowing model is used to simulate the attenuation of the signal due to obstacles in the environment. The log-normal distribution is considered to model the shadowing, which is given by:

$$\mathrm{PL} = \mathrm{PL}\left(d_0\right) + 10\gamma \log 10\left(\frac{d}{d_0}\right) + X \tag{5}$$

where PL is the path loss, d_0 is the reference distance, γ is the path loss exponent, and X is a random variable representing the log-normal shadowing with a standard deviation σ.

6. Fading model: The fading model is used to simulate the variation in the signal amplitude and phase due to multi-path propagation. The Rayleigh distribution is used to model the fading, which is given by:

$$h = \left(\frac{1}{\sqrt{2}}\right) * (a + ib) \tag{6}$$

where a and b are independent, identically distributed Gaussian random variables with zero mean and unit variance.

7. Additive White Gaussian: To simulate the effects of noise and interference in the channel model, we have considered AWGN. Therefore, the output of the channel is described as having the form $y \in C^{N_{\mathrm{msg}}+L-1}$.

$$y = x + n_k \tag{7}$$

where $n_k \sim \mathrm{CN}\left(0, \sigma^2\right)$.

The stochastic channel model takes the transmitted signal samples as input and applies the above-mentioned channel effects to produce the received signal samples. These received signal samples are then fed into the receiver net of the autoencoder model for decoding and demodulation.

3.1.3 Receiver net

Demodulation of LoRa signals relies on the orthogonality of the underlying basis signals, as well as the cross-correlation qualities that exist between the 2^{SF} different potential LoRa basis signals. The LoRa receiver's correlator would provide 2^{SF} outputs according to the received signal's correlation with all basis signals. The default approach for recognizing LoRa symbols is to choose the index of the LoRa base signal that has the maximum correlation value with the signal that has been received.

As shown in Fig. 3, the receiver net is designed as DNN architecture consisting multiple fully connected layers with its final layer activated using a "softmax" function. The modulus of the de-chirped received samples will serve as the input to the receiver net, and there will be C^n input nodes. The output is comprised of the bits that make up the symbol that is being conveyed, which results in output nodes equal to M. The ReLU activation is employed in the hidden layers of the network. In the output layer, the sigmoid function is used to transform the output values into a range from [0, 1]. To avoid overfitting, batch normalization is included in the hidden layers. The complex value is transformed into its real value in the receiver network by

Table 1 Summary of LoRa network Parameters

Parameters	Symbols	Value
Transmission frequency	f	868 MHz
Spreading factor	SF	7,8,9,10,11,12
Bandwidth	BW	125 kHz
Coding rate	CR	4/5
Transmission power	P_t	16dBm
Transmission radius	R	1000 m
Node distribution	–	Random
Traffic Model	–	Poisson distribution
Propagation Model	–	Log-normal shadowing and fading
Pay load	PL	100bytes
Path loss exponent	γ	4

Table 2 Autoencoder network architectural parameters

Sender net	Parameters	Output dimension
Input	0	1
Embedding	65,536	256
Dense (ReLU)	65,792	256
Dense (ReLU)	2056	8
Normalization	0	8
Channel net		
Previous layer input	0	56
Dense (ReLU)	14,592	256
Noise layer		
Receiver net		
Input	0	56
Dense (ReLU)	14,592	256
Dense (ReLU)	65,792	256
Dense (softmax)	65,792	256

adding the real and imaginary parts of the channel's output. The output of receiver $b\epsilon(0,\ 1)^M$ is a vector of probabilities that allocates probabilities to each of the potential messages.

4 Implementation

The proposed autoencoder is implemented for LoRaWAN communication by training the autoencoder with IQ data collected from LoRa nodes. Table 1 shows the LoRaWAN attributes that were considered for system design and data gathering.

Table 2 lists the architectural parameters of an autoencoder network. The Stochastic Gradient Descent (SGD) method is implemented for training the autoencoder. As the communication issue has been formulated as a classification challenge, it is appropriate to employ the cross-entropy loss function

$$L_{\text{loss}} = -\log\left(b_s\right) \tag{8}$$

where b_s stands for the sth component of b. In light of the fact that we are working with an autoencoder, which requires that the output and the input be the same throughout training, we have a set number of M distinct training labels. Usage of randomized channel functions as a regularization prevents the NN from overfitting, as the receiver never sees the same training sample again. Training using the same set of M labels over and over again yields an unlimited amount of labeled training data. The autoencoder was separated into a transmitter (TX) and a receiver (RX) after training, with each half responsible for establishing a different mapping TX: $M \rightarrow C^n$ and RX: $C^n \rightarrow M$.

4.1 Training of autoencoder

The main benefit of representing the whole communication network from end to end as an autoencoder is that it makes the training process easier. The backpropagation technique [23] is used by all cutting-edge DL programs to efficiently calculate gradients across all layers. This technique can only be used with models of stochastic channels, which provide a mathematically differentiable transfer function for the channel throughout all training data. To address the mentioned issue, a two-stage training technique related to transfer learning [24] is employed. The overall training process [25] is depicted in Fig. 5. To begin, an autoencoder is trained with the use of a stochastic channel model that was built to precisely describe the LoRa channel

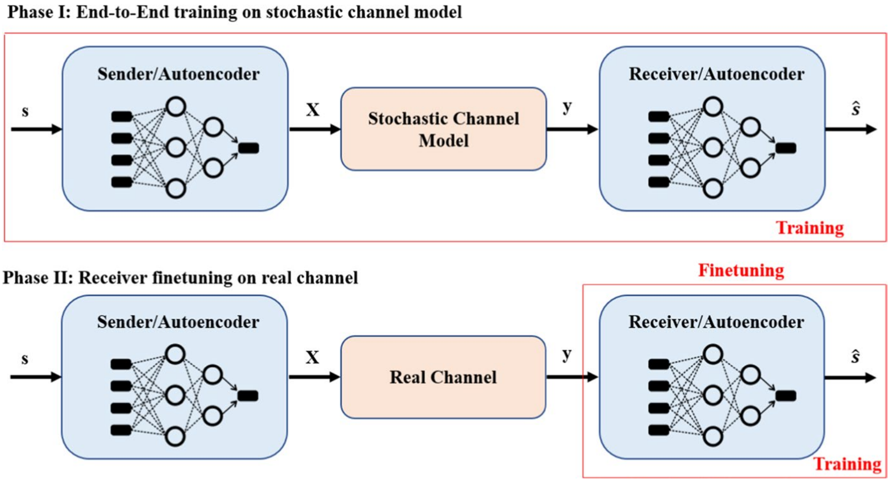


Fig. 5 Representation of two-phase training technique

behavior (Phase I). The pseudocode for stochastic channel modeling is given in algorithm 1. The initial performance will be dependent on the accuracy of the model with trained transmitter and receiver deployment for over-the-air communication. Inconsistencies between the simulated and actual communication channels are a common source of system failure.

We address this mismatch by performing (Phase II) by fine-tuning the receiver component of the autoencoder. The sender will send several messages across the real LoRa channel, and the receiver will record the corresponding IQ samples. After gathering enough data from these samples and message indices, a guided receiver tuning is performed. Transmitters in effective models of stochastic channels acquire representations 'x' of the message 's', which are resilient to distortions introduced by the channel. In spite of any differences in channel behavior, the acquired representations are sufficient for training a receiver that reliably recovers the transmitted messages.

```
Algorithm 1: Channel modelling
Input: x, SNR, $\alpha$, $\sigma_s$, $\gamma$, $\sigma_n$, $\sigma_K$
Output: y
Initialize: No. of samples $N_s = [none] * len(\mathrm{x})$
            channel samples $y = [none] * len(N_s)$
for $i$ in range $N_s$
    path loss $p_l = (\alpha * d(i))^{-\gamma}$
    sample path loss $S_{p_l} = N_s(i) * p_l$
    shadowing X = $N(0, \sigma_s)$
    sample shadowing $S_X = (S_{p_l} * 10)^{\frac{X}{10}}$
    fading $K = N(0, \sigma_K)$
    sample fading $S_K = S_X * K$
    AWGN = $N(0, \sigma_n)$
    $S_{AWGN} = S_K + AWGN$
    $y(i) = S_{AWGN}$
end for
return y
```

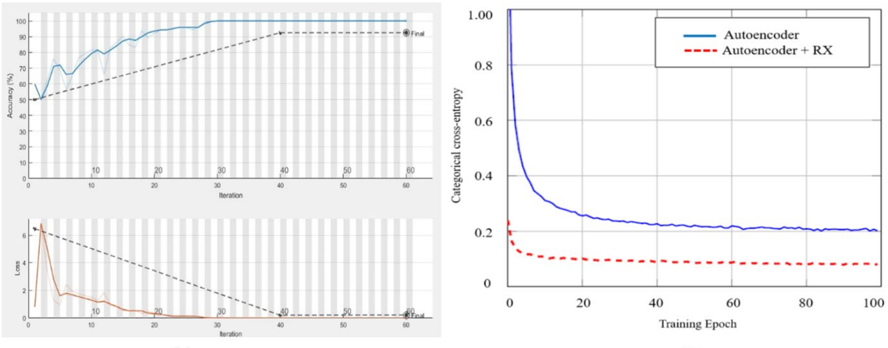
*Note: Here, x = input samples, SNR = signal-to-noise-ratio, α = path loss coefficient, γ = path loss exponent, σ_s = standard deviation of shadowing, σ_K = standard deviation of fading, σ_n = standard deviation of AWGN

5 Results and discussion

To train the autoencoder, we created a labeled dataset by simulation. This dataset contains LoRa IQ symbols with SF ranges from 7 to 12, a bandwidth of 125 kHz, a carrier frequency of 868.1 MHz, a coding rate of 4/5, and a number of various channel impairments, as described in Sect. 3. In the first step of the process, an open-source LoRa simulator is used to generate a set of LoRa symbols [26]. After that the data is processed via a radio channel emulator, which causes various impairments to be introduced into them. These impairments include AWGN noise, carrier offset, time offset, pathloss, and shadowing. The output of the channel emulator is recorded as in-phase and quadrature information, which consists of two sinusoids that are $\pi/2$ radians out of phase with one another. LoRa demodulation methods are then used on the channel's output to acquire the demodulated samples, which are then used to train the receiver net in the autoencoder. As mentioned in the training process, the receiver is trained with stochastic channel output samples and finetuned with the real channel samples.

In the first step of our process, we generated 10,000 LoRa IQ samples of data transmissions with SF values in the range of 7 to 12. The produced dataset is partitioned into three distinct subsets: training, validation, and test sets. The autoencoder is first trained with the help of the training data set, after which it is validated and tested for its performance in terms of BER and PSR. Later, the dataset is divided into segments based on each SF, and the autoencoder is tested using a set of data for each particular SF. We have also used a dataset with varying numbers of transmissions to test the autoencoder's robustness against interference, which grows in proportion to the number of transmissions. Therefore, the autoencoder is subjected to testing for a variety of SFs and interference levels as well.

The stochastic channel model is used to train the autoencoder employing SGD and Adam optimizer [27], with a learning rate of 0.001. We trained through 60 iterations of 10,000 arbitrary messages with a batch size of 50, 100, 500, and 1000 rising every 20 epochs. The validation accuracy rapidly surpasses 91.5%, while the validation loss decreases gradually as shown in Fig. 6a. To fine-tune the training data, we have used sequences for which the first receiver achieved a BER performance in range of 10^{-1} and 10^{-4}. To increase performance in the low SNR region, we used fine-tuning of sequences with which the original receiver's performance was less than around 10^{-1}. Additionally, it brought for a considerable error floor in the region with a high SNR. Due to the fact that fine-tuning with extremely noisy sequences obscures crucial data structure, the receiver loses its capacity to adapt to high SNR. Finally, we found that fine-tuning using sequences for which the BER was less than 10^{-4} led to a small but significant improvement. Fine-tuning was performed over



(a) (b)

Fig. 6 **a** Performance of autoencoder **b** Autoencoder training loss with and without fine-tuning of receiver net (RX) the course of one epoch per batch size, with batch sizes of 1000, 5000, and 10,000, respectively. The training is performed with a limited number of iterations to avoid the neural network being overtrained. A comparison of the autoencoder's performance with and without fine-tuning is depicted in Fig. 6b.

5.1 Performance metrics

The quality of service of a LoRa network depends on several parameters and metrics, including coverage, data rate, latency, packet success rate, bit error rate, scalability, and Interference. We evaluated the efficacy of a LoRaWAN network with the proposed autoencoder using the following three metrics.

5.1.1 Bit error rate

The bit error rate (BER) is defined as the percentage of lost data relative to the overall data. We have evaluated the performance of LoRa DL autoencoder with conventional LoRa in terms of BER. This evaluation consists of multiple tests corresponding to each SF. The BER is evaluated with respect to SNR for each individual SF. The BER curves of proposed model along with traditional LoRa are presented in Fig. 7. In conventional LoRa, the BER increases as the SNR decreases. The proposed autoencoder model provides substantially lower BER than traditional LoRa at the same SNR. The improvement in BER is more prominent at lower SNR values. In terms of SF, the BER will typically decrease as the SF value increases, with the autoencoder providing almost the same level of performance across the board for all SF values.

5.1.2 Collision rate

The collision rate is the percentage of packets that collide with other packets and are not successfully received by the receiver device out of the total number of packets

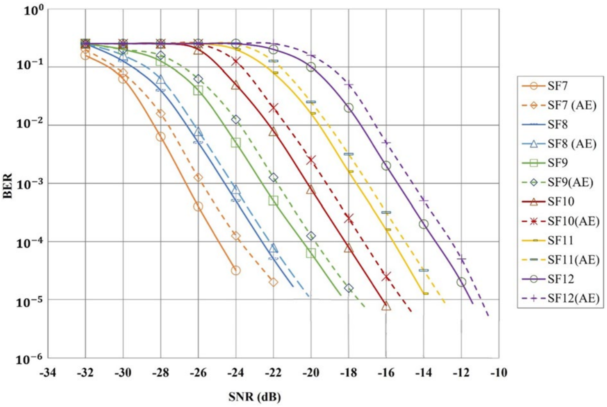


Fig. 7 BER analysis of proposed model in comparison with conventional LoRaWAN

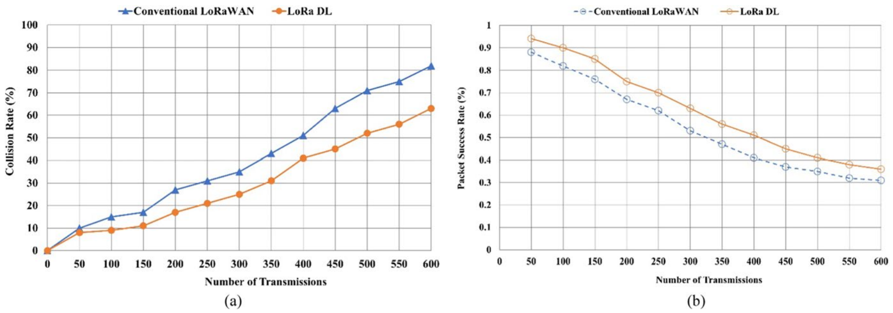


Fig. 8 Comparison of collision rate **a** and Packet success rate **b** of proposed model with conventional LoRaWAN

transmitted. The network throughput will be massively impacted by collision rate. As the number of transmissions increases, the collision rate will increase due to the increased likelihood of packets colliding with each other. The relationship between the number of transmissions and collision rate for proposed autoencoder and conventional LoRa are given in Fig. 8a. The conventional LoRa network shown collision rate of 82.5% for around 600 number of transmissions and the proposed autoencoder lowered it by almost 20% by achieving 62%.

5.1.3 Packet Success Rate

PSR is the ratio of reliably received data packets to the total number of data packets sent. In order to characterize the reliability and coverage of LoRa network, PSR values can be utilized. We have calculated the PSR of autoencoder by varying the number of transmissions. The comparative results are presented in Fig. 8b. The autoencoder shown 5% of improvement in PSR compared to traditional LoRa.

5.2 Testing scenarios

In order to evaluate the autoencoder in various test scenarios, the model's performance is examined for data with different spreading factors ranging from 7 to 12. Initially, the autoencoder is trained and tested with the dataset of samples taken from LoRa nodes which used all types of SFs. In this test scenario, the autoencoder was subjected to further testing using multiple set of input samples containing only data pertaining to each SF individually. The performance of the autoencoder in relation to the various input sets was calculated in terms of categorical cross-entropy loss of the model and the findings are presented in Fig. 9a. Based on the results, it can be concluded that the proposed autoencoder exhibits superior performance with higher SFs as compared to the lower ones, albeit the difference is relatively minor. Nonetheless, it can be ascertained that the autoencoder delivers remarkable performance for all SFs overall.

In a second scenario, the autoencoder is evaluated for different levels of interference. The autoencoder is tested with different datasets by increasing the number

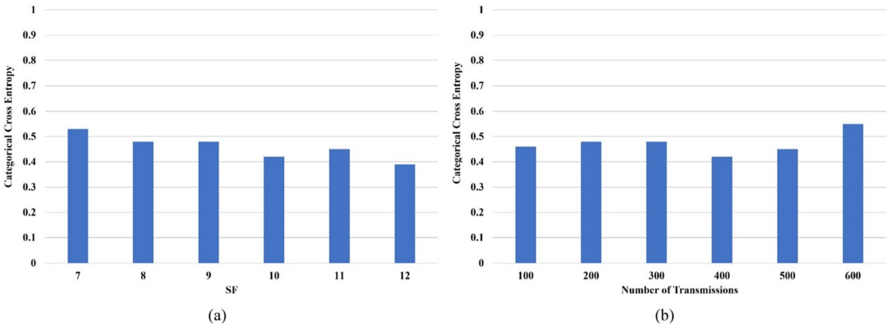


Fig. 9 Performance of proposed model for **a** different spreading factors **b** different levels of interference

of transmissions. The loss value of autoencoder for various levels of interference is presented in Fig. 9b. In a traditional LoRaWAN network, the number of collisions will rise as the number of transmissions in the network increases, which will result in a degradation of the network's performance. In the same manner, the performance of autoencoder degraded slightly with an increase in the number of transmissions, but not by an unacceptable amount.

6 Conclusion and future scope

In this article, we have proposed a novel approach to improve the performance of LPWANs by integrating the intelligence of DL. Specifically, we have demonstrated the feasibility of building an end-to-end communication system for LoRa using DNNs. We have developed a framework for designing, training and evaluating an autoencoder with different channel impairments to emulate practical LoRa communication channel conditions. Our results indicate that the proposed model performs efficiently in comparison with classic LoRa system. The proposed model achieved 5% of improvement in PSR and reduced 20% of collision rate. In terms of BER also the proposed model shown moderate improvement. We anticipate that performance will be enhanced further by adjusting hyperparameters and altering the NN topologies. Future work will involve rigorous training and testing to optimize different network parameters and improve overall performance. Implementing the proposed concept in real-time with hardware is also an interesting direction for future research.

Author contributions BS conceptualization, methodology, software, implementation, writing original draft preparation PRK implementation, investigation, writing original draft preparation RKJ visualization, writing-review, supervision. All authors have read and agreed to the published version of the manuscript.

Funding No funding was received for conducting this study.

Data availability The datasets analyzed during the current study are not publicly available due to individual privacy but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no competing interest.

References

1. Centenaro M, Vangelista L, Zanella A, Zorzi M (2016) Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. IEEE Wirel Commun 23(5):60–67. https://doi.org/10.1109/MWC.2016.7721743
2. Erpek TJT, Sagduyu YE, Shi Y, Clancy TC (2020) DL for Wireless communications. 2020. ArXiv. /abs/2005.06068. https://doi.org/10.48550/arXiv.2005.06068
3. Azzouz EE, Nandi AK (1996) Modulation recognition using artificial neural networks. In: Automatic Modulation Recognition of Communication Signals, pp. 132–176. https://doi.org/10.1007/978-1-4757-2469-1_5.
4. Ramjee S, Ju S, Yang D, Liu X, Gamal AE, Eldar YC (2019) Fast DL for automatic modulation classification. arXiv preprint arXiv:1901.05850. https://doi.org/10.48550/arXiv.1901.05850
5. West NE, O'Shea T (2017) Deep Architectures for modulation recognition. In: 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). IEEE, pp. 1–6. https://doi.org/10.1109/DySPAN.2017.7920754.
6. A Klautau, N González-Prelcic, A Mezghani, Heath R (2018) Detection and channel equalization with DL for Low resolution MIMO systems. In: 52nd Asilomar Conference on Signals, Systems, and Computers, pp. 1836–1840. https://doi.org/10.1109/ACSSC.2018.8645551
7. Xu W, Zhong Z, Be'ery Y, You X, Zhang C (2018) Joint neural network equalizer and decoder. In: 15th International Symposium on Wireless Communication Systems (ISWCS), pp. 1–5. https://doi.org/10.1109/ISWCS.2018.8491056.
8. O'Shea T, Hoydis J (2017) An introduction to DL for the physical layer. IEEE Trans Cognit Commun Netw 3(4):563–575. https://doi.org/10.1109/TCCN.2017.2758370
9. O'Shea TJ, Hoydis J An introduction to machine learning communications systems. arXiv preprint ArXiv:1702:00832. https://arxiv.org/pdf/1702.00832v1.pdf.
10. Aoudia FA, Hoydis J (2019) Model-free training of end-to-end communication systems. IEEE J Sel Areas Commun 37(11):2503–2516. https://doi.org/10.1109/JSAC.2019.2933891
11. O'Shea TJ, Roy T, West N, Hilburn BC (2018) Physical layer communications system design over-the-air using adversarial networks. In: 26th European Signal Processing Conference (EUSIPCO). 529–532. https://doi.org/10.23919/EUSIPCO.2018.8553233
12. O'Shea TJ, Roy T, West N (2019) Approximating the void: learning stochastic channel models from observation with variational generative adversarial networks. In: International Conference on Computing, Networking and Communications (ICNC), pp. 681–686. https://doi.org/10.1109/ICCNC.2019.8685573.
13. Shilpa B, Radha R, Movva P (2022) Comparative analysis of wireless communication technologies for IoT applications. In: Raje RR, Hussain F, Kannan JR (eds) Artificial intelligence and technologies: select proceedings of ICRTAC-AIT 2020 lecture notes in electrical engineering. Springer, Singapore. https://doi.org/10.1007/978-981-16-6448-9_39
14. Liando JC, Gamage A, Tengourtius AW, Li M (2019) Known and unknown facts of lora: experiences from a large-scale measurement study. ACM Trans Sen Netw 15(2):1–35. https://doi.org/10.1145/3293534
15. What is LoRaWAN Specification, LoRa Alliance, San Jose, CA, USA, (2020). [Online]. Available: https://lora-alliance.org/about-lorawan/.
16. Shilpa B, Prabhat GH, Kumar JR (2022) A LoRa-based low-cost data transmission system for smart building with energy and delay constraints. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.19398713.v1
17. Sandoval RM, Garcia-Sanchez A-J, Garcia-Haro J (2019) Optimizing and updating LoRa communication parameters: a machine learning approach. IEEE Trans Netw Serv Manage 16(3):884–895. https://doi.org/10.1109/TNSM.2019.2927759
18. Tesfay AA, Simon EP, Kharbech S, Clavier L (2021) DL-based signal detection for uplink in LoRa-like networks. In: IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile

Radio Communications (PIMRC), Helsinki, Finland, 2021, pp. 617–621. https://hal.science/hal-03373813

19. Lee S, Lee J, Hwang J, Choi JK (2021) A novel DL-based IoT device transmission interval management scheme for enhanced scalability in LoRa networks. IEEE Wirel Commun Lett 10(11):2538–2542. https://doi.org/10.1109/LWC.2021.3106649
20. Purohit J, Wang X, Mao S, Sun X, Yang C (2020) Fingerprinting-based indoor and outdoor localization with LoRa and DL. In: IEEE Global Communications Conference. pp 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9322261
21. Tesfay AA, Kharbech S, Simon EP, Clavier L (2023) signal denoising and detection for uplink in LoRa networks based on bayesian-optimized deep neural networks. IEEE Commun Lett 27(1):214–218. https://doi.org/10.1109/LCOMM.2022.3217337
22. Friis HT (1946) A note on a simple transmission formula. Proc IRE 34(5):254–256. https://doi.org/10.1109/JRPROC.1946.234568
23. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536. https://doi.org/10.1038/323533a0
24. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359. https://doi.org/10.1109/TKDE.2009.191
25. Dörner S, Cammerer S, Hoydis J, ten Brink S (2017) DL based communication over the air. IEEE J Sel Top Sign Process 12(1):132–143. https://doi.org/10.1109/JSTSP.2017.2784180
26. Al Homssi B, Dakic K, Maselli S, Wolf H, Kandeepan S, Al-Hourani A (2021) IoT network design using open-source LoRa coverage emulator. IEEE Access 9:53636–53646. https://doi.org/10.1109/ACCESS.2021.3070976
27. Kingma DP, Adam JB (2014) A method for stochastic optimization. arXiv 1412:6980. https://doi.org/10.48550/arXiv.1412.6980

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.