Optimization of LoRaWAN Based IOT When Applied to Warehouse Battery Monitoring System Through Time Serial Forecasting

by

Benjamin Avery Tabatowski-Bush

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Electrical and Computer Engineering
in the University of Michigan-Dearborn
2024

Doctoral Dissertation Committee:

Professor Weidong Xiang, Chair
Professor Di Ma
Professor Alex Yasha Yi
Professor Hafiz Malik

Benjamin A. Tabatowski-Bush

btabatow@umich.edu

ORCID iD: 0000-0002-2696-8090

Table of Contents

List of Figures iv

List of Tables v

List of Equations vi

Abstract vii

Chapter 1 Dissertation Research Description 1

1.1 Objectives 1

1.2 Research performed 1

1.3 Significance 2

1.4 Description of methods, methodology, procedures, approach 2

Chapter 2 An Intelligent Optimization Scheme for LoraWAN Based Electric Vehicle Batteries Monitoring System Located in Warehouses 4

2.1 Introduction 4

2.1.1 Background 4

2.1.2 Motivation for work 8

2.2 System model 9

2.3 Use cases for network traffic 11

2.4 Optimization of reward 14

2.5 Simulation of LoRaWAN 17

2.6 Case study 19

2.7 Discussion and conclusions 22

Chapter 3 LPWAN Channel Status Prediction Based On LSTM 23

3.1 Introduction 23
3.2 Background on LoRa Radio 24
3.3 Warehouse Radios Implementation 34
3.4 RSSI Machine Future Prediction Approach 35
3.5 RSSI Characteristics 41
3.6 Results 46
3.7 Conclusion 46
References 47

List of Figures

Figure 1 (a) Reward vs. power over SF (b)Maximum reward vs. traffic 6

Figure 2 (a) LoRaWAN battery warehouse and (b) individual radio 10

Figure 3 Three use cases for batteries in a warehouse, by criticality 12

Figure 4 Details of NHITS architecture 14

Figure 5 Traffic level over time, 3 use cases 16

Figure 6 Overview of the simulation software tool 18

Figure 7 Three types of zone groupings in warehouse 19

Figure 8 Rewards by varying zone and shadowing 20

Figure 9 Mean rewards by approach (graph and table) 21

Figure 10 RSSI vs. time from experiment with LoRa radios and moving objects 36

Figure 11 K-means clustering on peak RSSI and peak width 38

Figure 12 RMSE values for multiple trained LSTMs 39

Figure 13 Nodes in warehouse at different distances from gateway 43

Figure 14 RSSI machine future prediction 45

List of Tables

Table 1 Parameters to generate representative traffic time series, by use case 13

Table 2 Relationship of traffic level to power, reward, and optimal Tx power..... 21

Table 3 LoRaWAN key characteristics by data rate..... 27

Table 4 Achievable communication distances by shadowing and data rate..... 43

Table 5 Results of RSSI machine future prediction vs. other methods 46

List of Equations

(1) 5
(2) 12
(3) 12
(4) 12
(5) 26
(6) 28
(7) 28
(8) 28
(9) 28
(10) 29
(11) 29
(12) 29
(13) 35
(14) 35
(15) 35
(16) 42
(17) 42

Abstract

This dissertation proposes and studies a novel optimization scheme for LoraWAN based electric vehicle batteries monitoring system located in warehouses by utilizing techniques to optimize packet delivery and power settings. At first, based on my simulation of such a LoraWAN system, I observed and concluded that the optimized setting of such a system mainly depends on the network traffic regulated by the number of active users in the next channel access window. I therefore define the reward of the system as the multiplication of the packet delivery rate and power efficiency. A system setting consists of a traffic level and an appropriate radiation power. The optimized system setting results in maximum reward. However, the network traffic, or the number of active users in the next access window is time-varying and unknown due to the adoption of pure ALOHA protocol as well as the different packet rates and operation modes of battery LoraWAN sensors. Several factors lead to optimization of reward, including duty cycle regulation and future traffic prediction. The most effective optimization requires management of the traffic level based on criticality. A method called duty cycle management allows the lowest network traffic level with highest throughput and reward. When batteries have critical or urgent data, the specific nodes drive elevated network traffic levels for timely notification. With a large number of batteries in critical status, network traffic is at a maximum level. The goal of duty cycle management is to achieve an optimal network traffic level for different battery operating modes. Another factor in optimization is the future prediction of traffic, utilized for setting transmit power. I also examine optimizing reward with radios distributed in different zones around the collector. Also, the role of shadowing is considered in finding optimal system settings. I develop a network

mechanism to survey the traffic level, power, and reward, storing the lowest power level that provides 80% of the maximum possible reward. Simulations validate the proposed scheme, which achieves the best results with an average 31% higher reward compared to competing approaches without the duty cycle management. Further chapters in this dissertation elaborate on topics related to networks of LoRaWAN radios and the optimization of their performance.

Chapter 1 Dissertation Research Description

For this dissertation research, I will now lay out what were my objectives, what research was performed, and what was its significance. I will then describe my methods, the methodology, procedures, and approach for the research. I cover what I wanted to do, why I wanted to do it, and what benefits are accrued now that the research is complete.

1.1 Objectives

The objective of my research was to overcome present limitations in internet of things (IoT) wireless networking, specifically in the area of LoRaWAN networking in order to enable its usage for in infrastructure for battery management, specifically for monitoring electric vehicle batteries and modules in warehouses. This monitoring of batteries in warehouses also supports a second life for these lithium batteries. The known limitations included interference by a number of sources including other battery transmitters, the variability of the wireless channel in warehouse environments based owing to shadowing, a wireless channel that, owing to interference, does not offer enough bandwidth to carry all of the possible traffic by stored battery modules, and interference sources that are periodic in nature or random in nature but are unknown at the time of system design.

1.2 Research performed

Research took place in the simulation domain. Also, data was measured in building settings, and as well, battery warehouses. One topic was to find optimal parameter settings for

LoRa radios, for example spreading factor and transmit power (Pt), in order to maximize a reward, by modeling network throughput and power efficiency. Another topic was the modeling receive signal strength indication (RSSI) of wireless signals received at present and to predict their values within a coherent window, up to 100ms, through adoption of long short term memory (LSTM) neural network.

1.3 Significance

A discrete event simulator for LoRaWAN can model the interactions between battery wireless nodes in the warehouse setting and allow the determination of optimal networking parameters. RSSI is readily captured by various wireless devices in real-time mode, the variations of which find new applications when advanced machine learning approaches are applied. Fundamental concerns of wireless networking in battery warehousing have been addressed, including reliability and density, which will enable expansion of the monitoring of batteries in infrastructure settings. This improves the distribution of batteries, and supports the reuse of electric vehicle batteries.

1.4 Description of methods, methodology, procedures, approach

I provide the description of experimental methods and procedures employed during the research. I studied battery warehouse wireless communications. I performed measurements on actual warehouse environments containing batteries, along with battery communication systems to establish system parameters. **What I wanted to do** is overcome foundational limitations in very large wireless networks (up to 30,000 transceivers operating in close proximity.) Typical mesh networks for internet of things have covered hundreds of nodes in close proximity, but most

existing protocols such as Bluetooth mesh do not cover large numbers of nodes. My challenge comes from the fact that each battery array contains an active transceiver to participate in warehouse networking. These transceivers are located in close proximity; for example, as will be demonstrated later in this paper, a standard US warehouse can hold 30,000 battery modules, each containing a LoRaWAN radio. For so many transceivers, we need to overcome the limitations to large scale IoT networking found in approaches such as Zigbee [1] and Bluetooth mesh [2]. **Why I wanted to do this** is because up until now, the existing radio architectures for wireless sensor networks [3] were not used for dense deployments of wireless transceivers as I found in my warehouse use case. **What benefits we now accrue as result of my research:** I have successfully demonstrated a practical approach for large scale deployments which supports 30,000 LoRaWAN radios in a typical warehouse, in which we can now obtain full density because the batteries are able to be monitored. This enabling technology for warehousing batteries support the battery monitoring use cases of pre-consumer, second life, and awaiting recycling.

Chapter 2 An Intelligent Optimization Scheme for LoraWAN Based Electric Vehicle Batteries Monitoring System Located in Warehouses

2.1 Introduction

2.1.1 Background

There is an important infrastructure that underpins electrification. Since most electrified transportation runs on lithium batteries [4], there are a large number of logistics warehouses worldwide that store lithium batteries in different stages of assembly (cells, modules, packs) and at different stages of life (pre-consumer, second life, awaiting recycling.) This logistics infrastructure has an interesting and growing challenge with the modern trend of increasing transportation electrification. This challenge has to do with risks associated with lithium batteries such as thermal runaway and thermal propagation [5]. Logistics warehouses in the automotive space have historically handled flame-retardant materials. Today, these warehouses must employ special considerations for the stacking, monitoring, and access of lithium batteries [6]. This paper contributes to the usefulness of LoRaWAN radios by addressing the LoRaWAN needed for critical monitoring applications, which is a low cost, low power, reliable telematics IoT solution for sensors that under some use cases, transmit critical messages and need to ensure high packet delivery and low power usage. The key attributes of these battery radios are high reliability (minimum interference/packet collisions and sufficient packet delivery rate), high radio density (large numbers of batteries in the warehouse,) low power consumption, and low cost. However, for LoraWAN the performance of high throughput with low power suffers from the collisions of

packets due to the adoption of ALOHA random access. In fact, system optimization of LoraWAN becomes a critical and challenging topic due to diverse and time-varying active nodes, radiation powers, operation modes, duty cycles, interferences, and wireless channel status. To this end, this work contributes by proposing an intelligent optimization scheme for LoraWAN critical monitoring applications, through timely regulating packets during next access windows by using one of latest time serial neural networks for traffic prediction. At first, a new metric of system reward, *r*, is adopted to evaluate the performance of LoraWAN, which is defined as the product of packet drop rate and power efficiency, given in (1) where p_d is packet delivery rate and η the power efficiency. P_{max} , P_{min} and P_t are the maximum, minimum and actual transmit power (Pt) of a LoraWAN sensor, respectively, in mW.

$$r = p_d \eta = p_d \frac{P_{max} - P_t}{P_{max} - P_{min}} \tag{1}$$

Certainly, several factors in the physical layer will impact the reward including radio power, spreading factor, and wireless channels. However, based on our simulations, of which the results are shown in Figure 1(a) we concluded that for LoraWAN applied in warehouses, by selecting a spreading factor of 7 and sufficient power, leads to large rewards. However, the adaptive management of system traffic in the media access (MAC) layer becomes more critical to optimize rewards, which indicates the timely adjustment of the packets to be sent during next access windows from all the nodes. Figure 1(b) introduces the relationship between traffic and maximum reward.

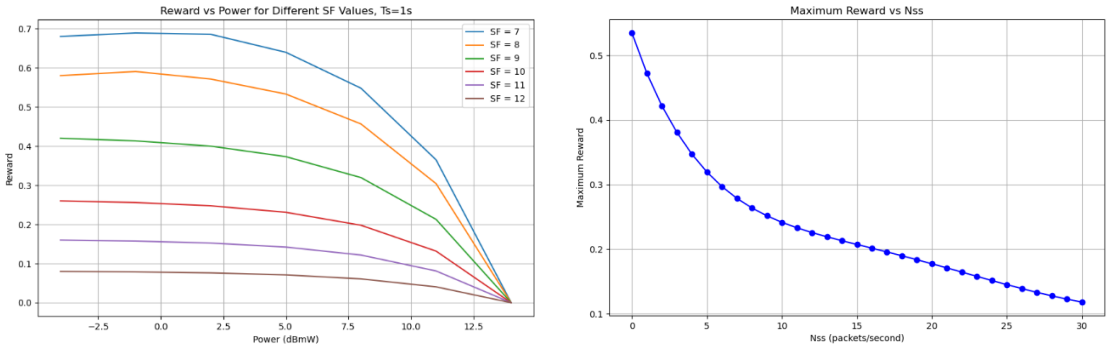


Figure 1 (a) Reward vs. power over SF (b)Maximum reward vs. traffic

We examine the overall relationship in our system of traffic level to maximum possible reward. Referring to Figure 1(b), we use network simulation described in detail later to iterate through traffic levels of interest, ranging from zero to 30 packets per second. Figure 1(b) shows a clear downward trend in the maximum possible reward as traffic level increases, because of the decrease in the packet delivery owing to additional radios transmitting and collisions. An examination of Figure 1(b) reveals reducing traffic below a certain threshold provides a significant improvement in the reward, which foreshadows the effectiveness of duty cycle management for optimizing results. We might suppose that the most effective reward improvement comes from managing the traffic level or network service speed (Nss,) measured in packets per second. However, more urgent battery situations warrant higher traffic in support of quickly notifying the warehouse owner. Therefore, we control traffic level on the criticality of batteries.

In order to find details about a LoRaWAN IoT solution that will meet the needs of battery warehousing, we consider the main attributes of LoRaWAN radios. One of the key characteristics of a LoRaWAN network is the traffic level, which is alternatively referred to as duty cycle, or in other networking contexts, activity level. Traffic level is measured in packets per second (PPS). The usual LoRaWAN metric for this activity, duty cycle, gives the ratio of time radios are

transmitting to the time no radios transmit in a transmission period. Other characteristics include the transmit power, measured in dBmW, and for LoRa radio, the Spreading Factor (SF) that relates to airtime per transmission. This chapter examines optimization of the LoRaWAN settings mentioned.

Optimizing system performance of LoraWAN through machine learning and other advanced technologies has been studied in previous literature. In [7] the authors applied k-means clustering, long short-term memory (LSTM), and decision tree approaches to predict packet inter-arrival times for LoraWAN networks. Moreover, the authors identified an incomplete dataset due to significant packet loss when performing training. This shortcoming requires predicting an optimized transmit power of a node in advance of transmission windows, to minimize the number of dropped packets. [8] uses principal component analysis (PCA), k-means clustering, and support vector machines (SVM) to predict packet delivery rate. The key finding is that measuring the latest network traffic and adjusting parameters plays the most important role in improving packet delivery, which underscores the importance of adaptive management of system traffic. [9] studied network congestion for LoRaWAN. It offered a method to identify the traffic congestion and provided the countermeasure of requesting nodes to aggregate their packets and reduce update rate. This approach is appropriate for background healthy battery transmissions, but not for time critical battery alerts. Another study on optimizing LoRaWAN is presented in [10]. The authors investigated LoRaWAN in indoor industrial areas through measuring received signal strength indicator (RSSI) and signal-to-noise ratio (SNR). They modeled RSSI using a neural network and further optimized by using particle swarm optimization. This approach enhanced bit error rate and highlights the importance of selecting optimal transmit power to maximize throughput. A study

on the control of power and spreading factor in LoRaWAN can be found in [11], although it does not utilize duty cycle management.

Moreover, there is plenty of research on predicting network traffic through using the latest time-serial neural networks. A study is presented in [12] to predict future values for network traffic, using an approach called neural forecasting with neural hierarchical interpolation for time series (NHITS.) This was based on an earlier work that also performed neural forecasting with neural basis expansion analysis for interpretable time series (NBEATS) [13]. Another study compared NHITS and NBEATS to modern techniques such as TimesNet [14] on the M4 dataset, and it is revealed the NHITS and NBEATS have superior performance. A further study compared a new technique SpectraNet [15] to NHITS and NBEATS; they beat SpectraNet for accuracy in the Exchange and ETTm2 benchmarks. The success of NHITS and NBEATS in these studies motivated their inclusion in this research for performing future traffic predictions.

The above observation and study motivates this study, where we adopt the latest time-serial prediction models, including NHITS and NBEATS, to predict network traffic and therefore an optimized system is achieved with a large reward.

The rest of the chapter is organized as follows. The balance of subsection 1 provides the motivation behind the work. Part 2 gives a full system model for LoRaWAN. Next, different use cases for network traffic are covered, followed by details on the simulation of LoRaWAN. After, we give the warehouse case study, and then a wrap-up with discussion and conclusions.

2.1.2 Motivation for work

The literature reviewed mentions usage of LoRaWAN radio for applications requiring reliable networking, and the need for optimization of parameters such as spreading factor (SF) and

transmit power, in order the achieve robust communications. Later in the chapter, we will demonstrate that optimization for these systems mainly depends on the traffic level in the upcoming transmit window. A first approach to optimization is to predict traffic in the upcoming window, and then set SF and power to optimal values. We define a reward by multiplying the packet throughput and a measure of power efficiency. This motivates our approach of predicting future traffic, so that we select optimal transmit power, which leads to optimized reward. However, the network traffic, or the number of active users in the next access window is time-varying and unknown due to the adoption of pure ALOHA protocol as well as the different packet rates and operation modes of battery LoraWAN sensors. To model and predict the network traffic, we utilize prediction approaches including average traffic, NHITS, and NBEATS for future prediction. We study the effectiveness of predictions and determine the most effective prediction approach.

A further approach to optimization is based on the observation that the most effective reward improvement comes from managing the traffic level based on battery criticality. The highest rewards come from low background traffic of nominal batteries, infrequently reporting. Optimal network performance is achieved by implementing duty cycle management on the network traffic, restricting higher traffic levels for use in critical battery reporting only.

2.2 System model

Our system is comprised of a LoRaWAN collector and many radio-equipped electric vehicle batteries and battery modules in a warehouse. A battery module is a mechanical assembly of multiple lithium cells that are grouped into a larger unit [16]. An example EV lithium battery module measures 400mm x 150mm x 230mm, referring to Figure 2b. A module can contain four

cells connected in series, which has four cell voltages and a temperature that needs to be monitored during warehouse storage [17]. In Figure 2(a) a warehouse has many racks holding batteries, all connected via LoRaWAN to a collector. The collector can further connect with the application server through the Internet. Users can therefore monitor and interact with the stored batteries. In Figure 2(b) one battery module is called a LoRaWAN end device (ED). For our system, we monitor each module using the Cell Monitor Unit (CMU) in order to measure cell voltages and module temperature. The CMU sends messages to the collector infrequently when it is in normal conditions, but frequently when in urgent conditions.

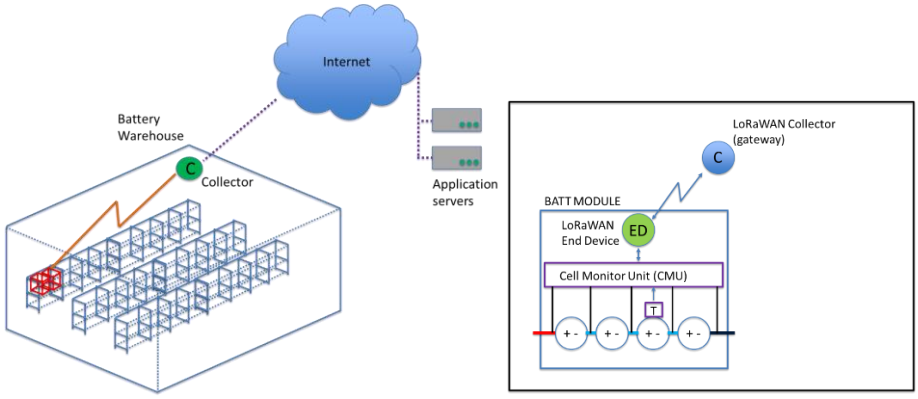


Figure 2 (a) LoRaWAN battery warehouse and (b) individual radio

Referring to Figure 2(a), the red container depicted on a storage shelf is a typical wooden shipping container 1m x 3m x 1.2m. A container can hold 20 modules while a warehouse rack contains 15 wooden crates, resulting in 300 modules. A standard warehouse size is about 40m x 40m and a typical warehouse can hold 100 metal racks [18]. Totaling, the warehouse contains as many as 30,000 modules, each acting as a LoRaWAN sensor node.

2.3 Use cases for network traffic

As depicted in Figure 3, a battery is either normal, colored in green, or critical, colored in red. A normal battery only infrequently reports messages to the collector in a random timeframe, modeled by a Poisson process with a small packet rate of λbg as in (2). Meanwhile, normal batteries have some additional traffic with non-critical notifications (NCNs) modeled as a uniform distribution with an expectation value of E_{ncn} and a width W_{ncn} as in (3). Furthermore, we define use case A as a warehouse containing batteries all in nominal condition. For example, batteries are all at room temperature and in the middle of the state of charge range with good state of health. A uniform distribution packet is used to report the changes in temperature and the state of charge. Similarly, use case B indicates there are certain number of critical batteries that need to report messages frequently, which is also modelled as a Poisson process with a rate of λcrit. Finally, use case C represents a warehouse with a large population of critical batteries needing attention. Table 1 lists the typical values of above parameters. Use cases A, B and C are used as network traffic in the study to simulate the active packets to be sent by all LoraWAN nodes during an access window.

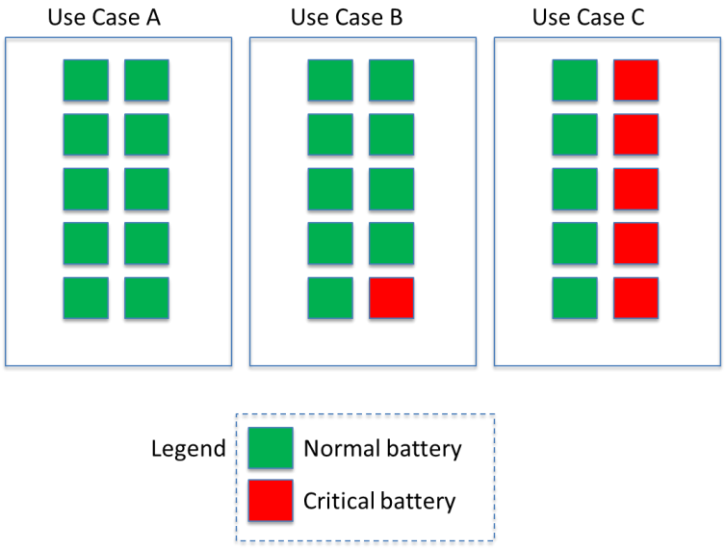


Figure 3 Three use cases for batteries in a warehouse, by criticality

A mixture of Poisson processes and a uniform distribution is adopted to model the traffic level of the above use cases. The probability of a traffic level, $P(Nss)$, of a Poisson distribution is given by (2):

$$P(Nss) = \frac{e^{-\lambda}\lambda^{Nss}}{Nss!} \tag{2}$$

For (2), N_{ss} is the number of active packets and λ is λ_{bg} for normal batteries and λ_{crit} for critical batteries. In addition, there are non-critical notifications of normal batteries, which follows a uniform distribution with an expectation value of E_{ncn} and a width of W_{ncn} as found in (3),

$$p(N_n) = \frac{1}{b-a} \tag{3}$$

where $p(N_n)$ is the probability density function at number of packets N_n and $a = \mathrm{E}_{ncn} - W_{ncn}/2$ and $b = \mathrm{E}_{ncn} + W_{ncn}/2$. Finally, the distribution models are combined through the formula of (4).

$$N(t) = N_{p\_bg}(t) + N_{u\_ncn}(t) + N_{p\_cr}(t) \tag{4}$$

Table 1 below provides the parameters of the three use cases.

Table 1 Parameters to generate representative traffic time series, by use case

Use Case	Critical Fraction	λbg	E_{ncn}	W_{ncn}	λcrit
A	0	0.3	2	2	0.1
B	0.1	0.3	2	2	3
C	0.5	0.3	1	2	10

Next, based on NeuralForecast python library [19], Neural Hierarchical Interpolation for Time Series Forecasting NHITS [12] is adopted to perform prediction on dataset generated by use cases A, B, and C. The input training dataset is past 100 samples of network traffic. Meanwhile, Neural Basis Expansion Analysis for Time Series (NBEATS) [13] is also used for reference. Figure 4 illustrates the concepts of NHITS [12]. The time series of network traffic data is shown in the upper left of the figure. The data from time t-L to t is at first sampled at S different rates. High sample rates are used to capture fast variations and low rates for trends in the data. Samples are fed into stacks, individually, each generating a forecast output labeled 'f' and a residual output labeled 'r'. The last stack S has a residual output that is not used. Each stack consists of B blocks in cascade. Each block contains multi-layer perceptrons (MLPs). Within a block, there is subtraction on the output compared to the block input, which determines the residual that is passed to the next lower block. The details of a block are illustrated in the right of figure. This MLP stack provides the core of the ability to forecast. It contains a MaxPool module that performs a pooling operation by sliding a window over the input data and taking the maximum value. There are two key MLP sections, one for the forecast and one for the backcast. The MLPs are trained to select the coefficients of basis functions or components of the time series. The last block in the MLP layer is a rectified linear unit (ReLU) that form a non-linear activation function for learning.

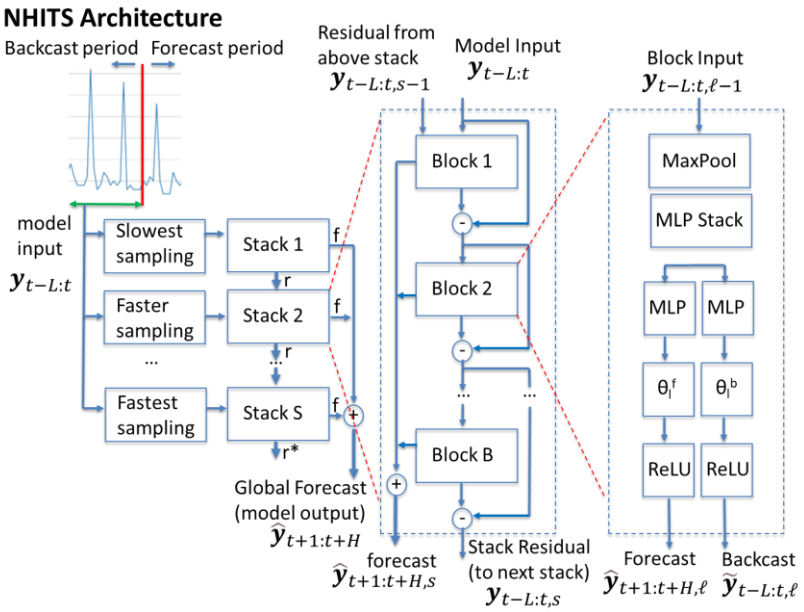


Figure 4 Details of NHITS architecture

2.4 Optimization of reward

Referring of Figure 5, we find the NHITS and NBEATS predictions on the right side of the graph. Since this data is comprised of random (not periodic) Poisson and uniform distributions, it is challenging for NBEATS and NHITS to make perfect future predictions. In order to improve reward, we look for other opportunities beyond future traffic prediction. We present a method of optimizing the throughput and power of LoRaWAN networks using a duty cycle regulation (DCR) approach, called Duty Cycle Management or DCmanage. The approach is implemented by managing the traffic of nodes across the network, especially nodes with urgent critical information that would otherwise increase the traffic of the network to levels with low reward. In Figure 1b, the portion of the graph below 5 packets per second has much higher reward than any other region, so we can see duty cycle management for reducing traffic level has the greatest improvement on network performance. The reason is twofold. First, by reducing traffic across the network to the

lowest possible level, we take advantage of the highest rewards according to the curve in Figure 1b. Reviewing Figure 5, we notice that network traffic is largely below 15 packets/second, and the management divisor of 3 means the DCmanage values are below 5 packets/second, which is the region of highest reward. Also with usage of DCmanage, the variation in traffic is a third of the baseline values, so that when average DCmanage traffic is used for selecting the single power level, this single power is the optimal value across the range of traffic. This virtuous property means that for DCmanage, accurate future traffic prediction can be implemented with a simple traffic average. The other approaches for future traffic prediction are challenged owing to the wider traffic variation that needs to be predicted. Later in the chapter we will compare the rewards of this lower traffic DCmanage approach to other methods of reward optimization.

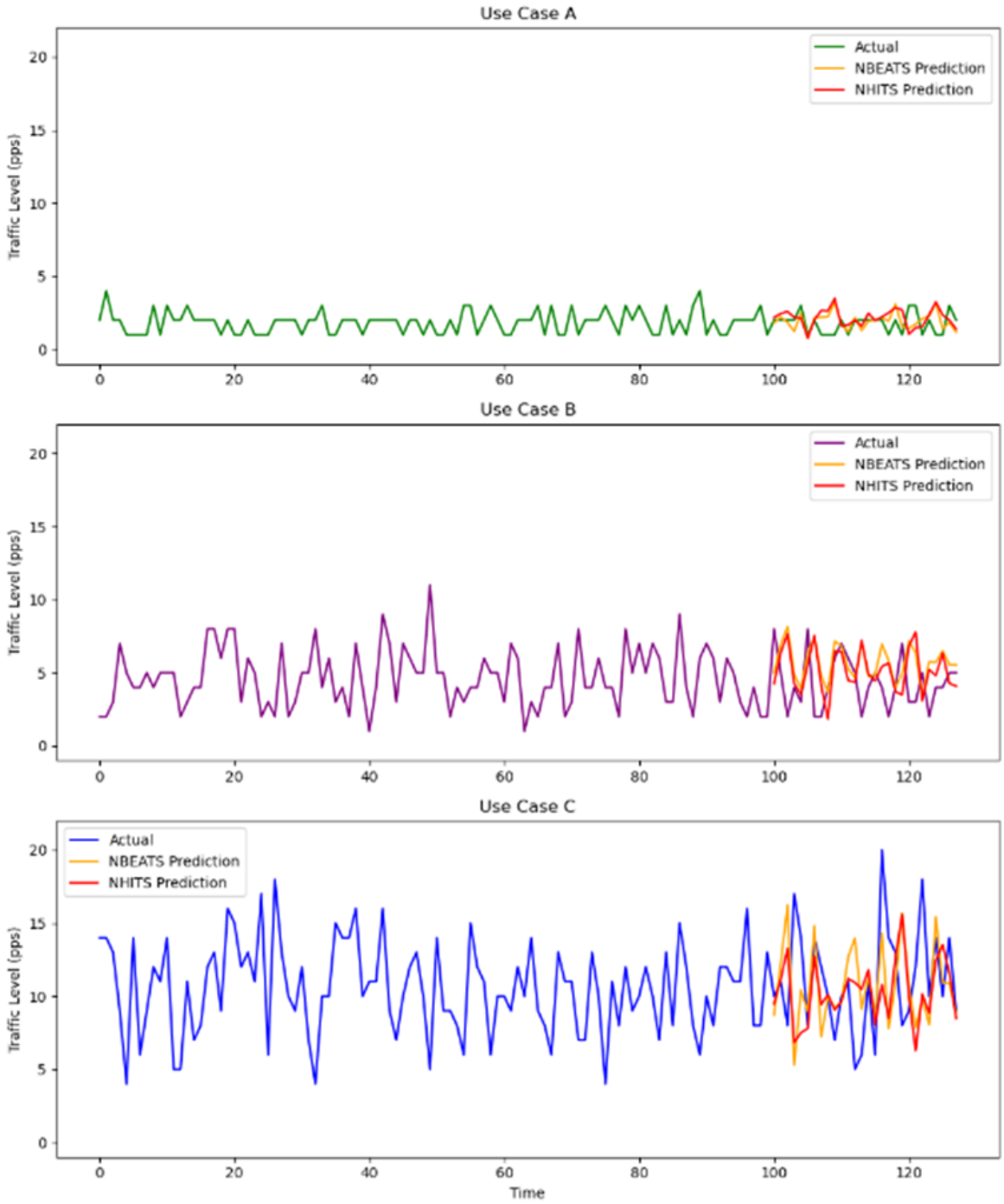


Figure 5 Traffic level over time, 3 use cases

2.5 Simulation of LoRaWAN

A simulator is implemented in Python to evaluate system performance of the previously defined reward for LoRaWAN in a warehouse. The end nodes are located on the racks in three dimensions, located in difference places for test purposes, with the collector in a fixed location. A path loss (PL) model is implemented, and the capture effect of chirp modulation is also considered to determine the collision of packets [20]. The typical warehouse configuration is shown in Figure 2. The traffic level prediction is performed at the computer that is connected to the collector. It sets the transmit power of end nodes, achieved through the ACK from collector to end node. For larger warehouses the nodes farthest from the gateway need larger valued SFs for the signal to reach the gateway, but this effect is not significant for a typical warehouse.

Referring to Figure 6, the flow chart illustrates the processes of the simulator that optimizes system settings to get best rewards. The simulation generates traffic data in packets/second and uses the previously described prediction network to predict future traffic. The prediction is thereafter fed into a pretrained multiple regression tool that models the relationship of reward vs transmit power. This regression tool provides the means to move from a predicted future traffic level to an optimal power level, achieving 80% of maximum reward 3dBmW lower, saving ½ of the transmit power.

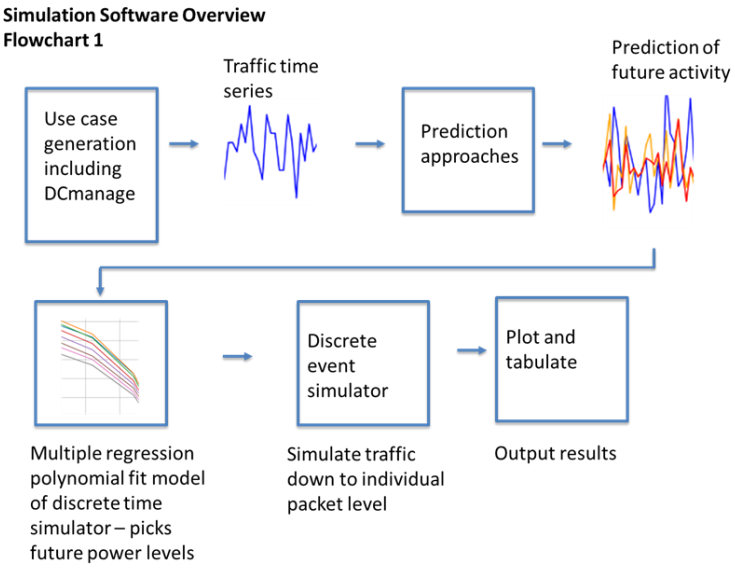


Figure 6 Overview of the simulation software tool

A baseline (BL) approach is adopted which selects sufficient power to achieve 80% of the maximum reward. This baseline is practice criteria and is used as a reference. Another approach is traffic averaging, here called AVG, for which we use 80% of maximum reward power criterion for one specific value of traffic, the scalar average value across the observation window prior to the next access window. The next approach, called RND, randomly selects a traffic value from the span of network traffic observed over the prior observation window, and prediction for the next access window is this random value. For RND, we also use 80% of maximum reward power criterion for this random traffic value. We also use the DCmanage approach to control network traffic, controlling traffic at all nodes to 1/MD of baseline, where MD is the management divisor of the baseline values, for example 3 across all use cases. The DCmanage approach uses the simple AVG method described above to take a single average traffic level across the prior observation window and use 80% of maximum reward power criterion for this scalar average value.

2.6 Case study

The following three cases shown in Figure 7 are studied using the simulator. In the case of zone 0, the batteries are located in a small grouping of racks and the collector is nearby. In the case of zone 1, the batteries occupy a large section of racks along the warehouse middle and the collector is located within the rack area. Zone 2 combines the two cases. The simulation results are displayed in Figure 8. The traffic level Nss ranges from 5-25 packets/s. From Figure 1a we saw that for all cases SF=7 generates a larger reward. On the other hand, a better reward is achieved when an appropriate value of power is selected for the given Nss. The bottom plot in Figure 8 shows the reward vs transmit power in zone 2 and with a shadowing to 36dB, which means a large amount of variability in path losses for different radios. Compared to the plot for zone 2, a higher reward is achieved at a lower power because of the throughput improvement of the radios with the lower path losses. Also, the radios with higher path losses collide less, also improving the reward.

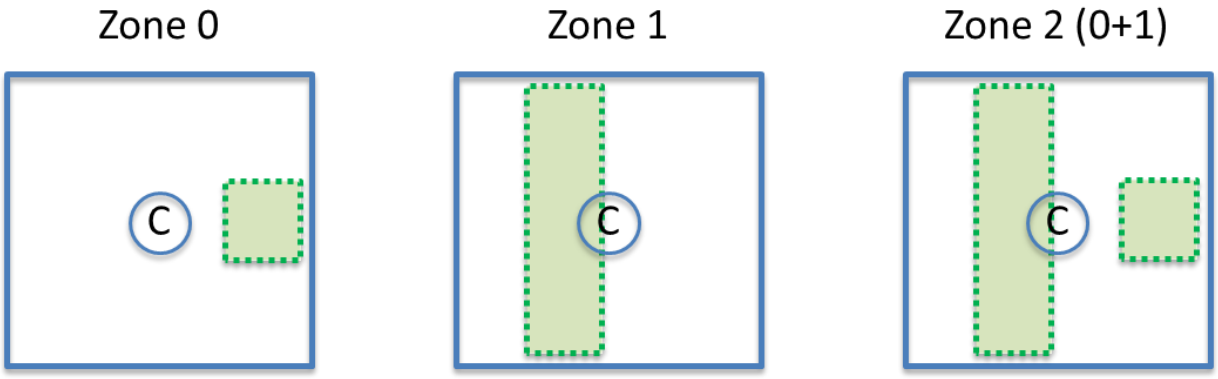


Figure 7 Three types of zone groupings in warehouse

This initial simulation of our LoRaWAN system gives us the observation and conclusion that the optimized setting of such a system mainly depends on the network traffic regulated by the number of active users in the next channel access window. We see in Figure 8 that reward is a strong function of traffic level Nss, and at a given traffic level, the power level chosen has a significant impact on the reward obtained.

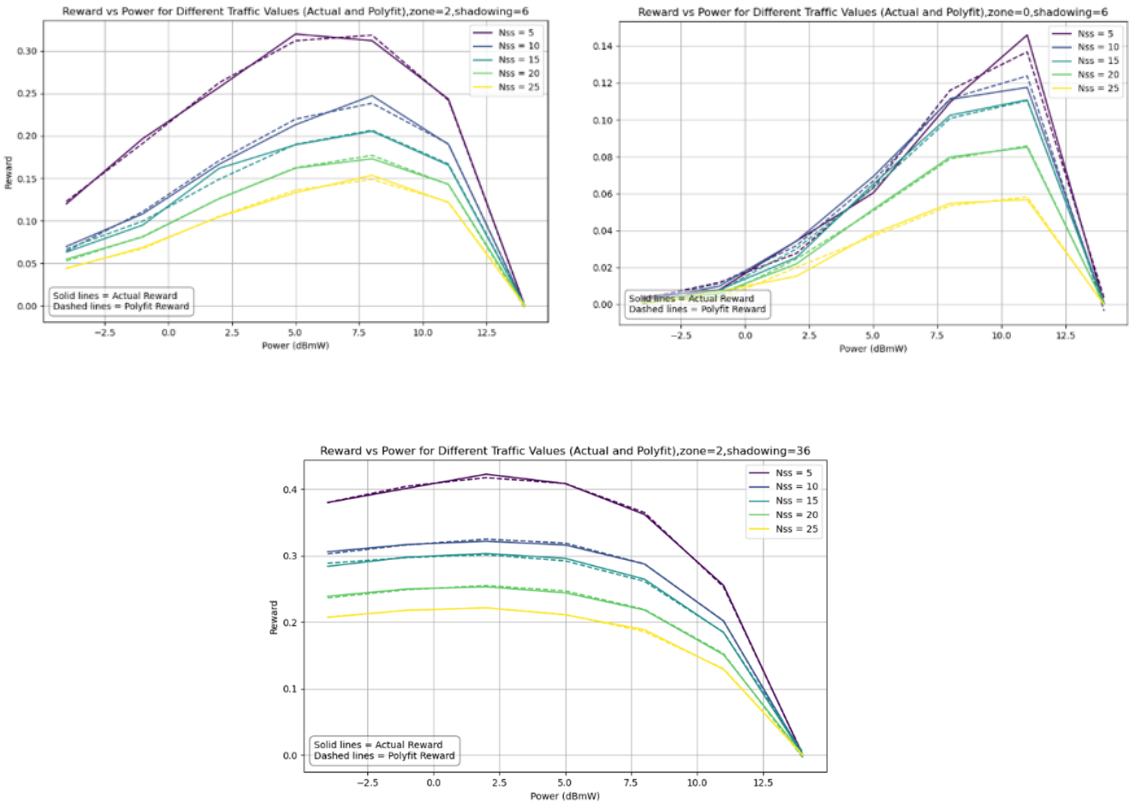


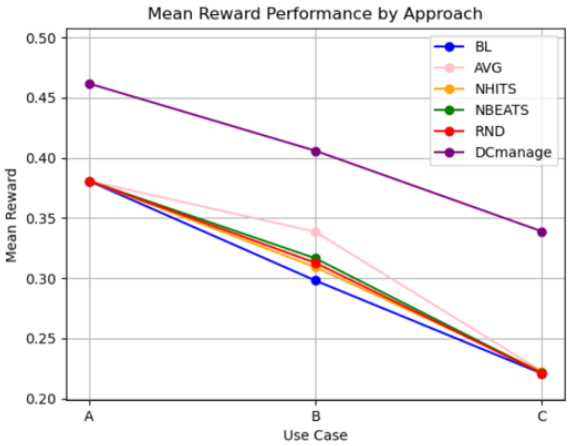
Figure 8 Rewards by varying zone and shadowing

A transmit power (Pt) criterion is to use sufficient power to achieve at least 80% of the maximum reward. Table 2 lists value of sufficient power based on above simulations. This determination must be made in advance of the transmission, which is delivered to each node through acknowledgements.

Table 2 Relationship of traffic level to power, reward, and optimal Tx power

Nss(packets/sec)	Transmit power Pt(dBmW) for Max reward	Maximum reward	80% of Max reward	Optimum Tx power (dBmW)
5	5	0.32	0.26	2
10	8	0.25	0.20	5
15	8	0.21	0.17	5
20	8	0.17	0.14	5
25	8	0.15	0.12	5

The network traffic, or number of active users in the next access window is time-varying and unknown due to the adoption of pure ALOHA protocol as well as the different packet rates and operation modes of battery LoraWAN sensors. The LoRa system has a time-varying reward and the proposed traffic level predictor can help system achieve good performance of 80% of the best reward with sufficient radiation power. For the warehouse use case listed above, the system reward also changes with system configuration and wireless channel status, which necessitates the use of traffic level prediction and other techniques for reward optimization. To summarize the simulation results, Figure 9 illustrates the reward performance of the DCmanage approach, which, on average, yields a 31% higher reward compared to other approaches.



	Use Case A	Use Case B	Use Case C
Dcmanage	0.461788	0.405891	0.338849
BL	0.380794	0.298	0.220691
avg	0.380794	0.338514	0.222532
NHITS	0.380794	0.309119	0.222532
NBEATS	0.380794	0.316483	0.221071
rnd	0.380794	0.312727	0.220782

Figure 9 Mean rewards by approach (graph and table)

2.7 Discussion and conclusions

This chapter studies the optimization scheme for LoraWAN applied to warehouse battery monitoring. At first, a system metric, reward, defined as the multiplication of packet drop rate and power efficiency, is adopted to evaluate the system performance. The maximum reward of 1 indicates the system delivers all the packets successfully using the minimal radiation power. Meanwhile, a practical criterion is suggested to select minimal power where the reward is within 80% of its maximum value. Next, a simulator is built to extensively explore the system settings achieving the optimized reward for various cases and under diverse channel status. We concluded that for LoraWAN applied to warehouses, the physical settings are to adopt sufficient power based on the size of warehouse and channel status and to select the fastest spreading factor of 7 since the packet drop mainly comes from collisions between overlayed packets. Secondarily, we identified the reward variation with the locations of sensor and collector as well as shadowing of the wireless channels. For a given environment and configuration with optimized physical settings, the duty cycle (packet rate) is the decisive factor for reward. To this end, a network traffic model is used to estimate the number of awake sensors, by combining a background Poisson distribution and a uniform distribution for non-critical notifications of normal batteries, and a separate Poisson distribution for critical batteries. A latest time-serial prediction model, NeuralForecast, is adopted to predict the network traffic in next transmission window, upon which the duty cycle is adaptively adjusted for all awake sensors to maintain high system rewards. The simulation results validate the proposed optimized scheme for LoraWAN applied to warehouses.

Further work is to adopt such scheme to a real LoraWAN based battery monitoring system in a warehouse.

Chapter 3 LPWAN Channel Status Prediction Based On LSTM

Low power wide area networking poses new challenges in the protocol for channel access. LoRa radios are limited in their ability to sense an occupied radio channel. For some use cases involving the tracking of lithium batteries in a typical North American warehouse, we need to manage multiple access for 30,000 radio nodes. Since these long-range radios give visibility of a given node to most of the other radios in the network, our method of channel access must accommodate a large amount of potential interference in the contention for the radio medium. We consider the case in which objects (people and machinery) can move and affect the path loss and cause interference to transmissions in predictable ways. The focus in our work is on channel status prediction. There is a limit to how accurately the RSSI could be predicted for a radio channel.

In this chapter, a deep-learning channel status prediction model is studied to model and predict the RSSI values at the next time step for LoRa systems. Experimental data was captured in real usage and used as a training dataset and test dataset, resulting in an accuracy of 2.6dBmW.

3.1 Introduction

We explore the use case of channel prediction for radio trackers for lithium battery arrays stored in warehouses. Since each battery array is monitored by a tracker radio, there can be many transmitter nodes in a typical warehouse. This chapter will consider the use case of 30,000 nodes in a 1500 square meter warehouse. The trackers are using LoRa radios [21]. In further sections, we will cover more details of this type of communication system. At a high level, this chapter is focused on the prediction of RSSI of the radio channel, which is needed as a method of predicting channel occupancy to avoid collisions to enable a higher density of LoRa radios in warehouses.

Looking further into what has been accomplished in the field of channel status prediction, we find that other researchers have applied machine learning techniques for indoor Received Signal Strength Indication (RSSI) prediction, for example [22]. In [22], a simple artificial neural network (ANN) is utilized to predict the RSSI at a given distance, and the best root of the mean square error (RMSE) obtained is 6.2dB. We investigate other works that improve RSSI measurement stability as a function of multipath fading effects, such as [23]. In [23], techniques were applied to mitigate RSSI variation through filtering, and the best technique had the result of 5dBm variation. Other research uses ray tracing for the prediction of RSSI [24], which achieves an accuracy of 5dB in its RSSI predictions. For RSSI prediction, the summary is that if our approach can predict RSSI more accurately than 5dB, it will be considered good performance.

This chapter is organized as follows. Section 1.1 provides a background on LoRa radio and available tools and resources. Section 1.2 describes the warehouse radio tracker use case and environment. Section 1.3 covers the new approach for improving channel status prediction, called RSSI machine future prediction. Section 1.4 studies the characteristics of RSSI in the warehouse use case. Section 1.5 provides the results, and section 1.6, the conclusion.

3.2 Background on LoRa Radio

For a background on slotted Aloha and LPWAN and dynamic frame-slotted ALOHA, see [25]. It states that LoRaWAN uses the ALOHA mechanism and that there is a high probability of collisions. A survey on LP-WA technology covers narrowband internet of things, and gives good coverage on Long Range Radio (LoRa) from Semtech [26]. A reference of an implementation of a LoRa receiver with good mathematical description can be found in [27]. To perform work with LPWAN, we will need good simulation capabilities to cover the specific features of LoRa radios. We look at a review of LoRaWAN simulators from 2019 [28]. The first LoRa simulator was

LoRaSim [29]. The LoRaSim simulator is built using SimPy [30]. The formal repository for the LoRaSim simulator is on GitHub [31]. The key mechanism for managing access to the radio medium and optimizing data extraction from end nodes in LoRa is the Adaptive Data Rate (ADR) mechanism from Semtech [32]. We can find survey on adaptive data rate (ADR) optimization [33] to find out about prior efforts for improvement. The LoRa developer's website gives useful information on packet size and data rate by region [34] that helps to develop specifics of an implementation that supports better channel prediction, will enable a high density of radios for US warehouses.

We are interested to know if better channel status prediction can improve the collision performance of LoRa radios. We start out by looking at a Performance Evaluation of LoRaWAN Communication Scalability [35] and review an analysis of capacity of the network [36]. To examine the collision properties of LoRaWAN, we start with the fundamentals of channel access, and a study that examines increases traffic collisions owing to the adaptive data rate (ADR) algorithm in LoRa [37]. This paper [37] mentions that LoRa's ADR mechanism suffers in highly dense networks. The paper proposes four algorithms and evaluates them using LoRaSim in a simulation where end devices transmit to a gateway every hour, and then predicting Data Extraction Rate (DER) as the network size increases. This paper [37] also states that while its four algorithms provide improvement over baseline ADR, it points out that as network size increases, the studied algorithms are still not adequate for a reliable and energy efficient network, and further research is called for. The paper starts by mentioning that ADR is used to achieve the highest data transmission rate and lowest transmit power possible. The paper then cites a survey paper [38] that covers ADR approaches.

The following application note is from Semtech and gives a background on the LoRa modulation method [39]. We now examine the LoRaWAN specification to find out details about LoRaWAN networks [40]. We find a definition for uplink messages from an end device to the network server through one or more gateways [40]section 3.1. The uplink packet for LoRaWAN has a preamble, a physical header (PHDR) with a header CRC (PHDR_CRC), the PHY payload, and a payload CRC. Downlink packets never use a payload CRC. The PHY payload per [41] contains a one byte MAC header MHDR, a MACPayload of length 7 to M bytes constrained by [42], and a four byte Message Integrity Check (MIC). Maximum payload PL size as a function of maximum MACPayload bytes M is found in (5) as:

$$PL = 5 + M \tag{5}$$

[42] section 2.5.3 defines a Data Rate DR*n* by region as a function of Spreading Factor (SF) and bandwidth. For US902-928 band using LoRa, the indicative physical bit rate ranges from 980bps for SF10 /125kHz as Data Rate (DR) 0, written as DR0, to 21.9kbps for SF7 / 500kHz as DR13. The other Data Rate values from DR1 to DR12 fall in between the given indicative physical bit rates. The maximum MACPayload size M as mentioned above ranges from a minimum size of M=19 bytes at DR0, to the highest M value of 230 bytes as found at DR13. So, the largest PHYPayload per [41] section 4.1 and [42] section 2.5.6 is 235 bytes for DR13. Some other physical layer requirements on a LoRa radio from [42]section 4.1.2 indicate the Preamble size for a LoRa packet to be 8 symbols, and the Synchronization word is 4.25 symbols. The combination of PHDR and PHDR_CRC is 8 symbols total per [42] section 4.1.1. For uplink packets only, there are 2 additional bytes at the end for the payload CRC.

To find out about hardware details of the LoRa implementation, we start out with the data sheet for LoRa radios from Semtech [43]. The given data sheet [43] covers four devices SX12376

through SX1279 which cover the frequency range of 137Mhz to 1020Mhz, with spreading factors ranging from six to twelve, and communication bandwidths from 7.8kHz to 500kHz. The effective bitrates range from 0.018 to 37.5 kbps and receive sensitivity down to -148dBm. In this chapter, we use demonstrated results from [29] instead of the data sheet sensitivity figures for receive sensitivity. The parts described in this datasheet [43] are in a 28 pin QFN package. The specification discusses details about the SF, which range from 6 to 12. Carrier Access Detection (CAD) is a method by the radio IC [43] to monitor the radio channel for a preamble, which would be an indication of transmission activity per [43] section 4.1.6. LoRa radios have several available data rates by market; for example, the US market has data rates from DR0 with SF10/125kHz to DR13 at SF7/500kHz. Table 3 provides a list organized by data rate for the US that includes Spreading Factor (SF), the bandwidth (BW) in kilohertz, symbol time in milliseconds, number of CAD symbols, the total CAD time required to detect the preamble in milliseconds, and the receiver sensitivity for the given data rate in dBm.

Table 3 LoRaWAN key characteristics by data rate

Data Rate	SF	BW (kHz)	Tsym (ms)	CAD #Ts	CAD time (ms)	Rcvr Sens (dBm)
0	10	125	8.2	1.8	14.5	-133
1	9	125	4.1	1.8	7.2	-131
2	8	125	2.0	1.8	3.7	-127
3	7	125	1.0	1.9	2.0	-127
4	8	500	0.5	1.8	0.9	-124
8	12	500	8.2	1.9	15.2	-132
9	11	500	4.1	1.8	7.4	-129
10	10	500	2.0	1.8	3.6	-129
11	9	500	1.0	1.8	1.8	-128
12	8	500	0.5	1.8	0.9	-124
13	7	500	0.3	1.9	0.5	-121

By reading the CAD time column in Table 3, we find that it could take as long as 15.2ms for the radio to detect a preamble using the Carrier Access Detect approach [43] page 44. Another item found in Table 3 is the receiver sensitivity which is as reported in [29]. For a communication bandwidth BW in bits per second and spreading factor SF, we define the symbol time Ts in (6) as:

$$Ts = \frac{2^{SF}}{BW}\ secs \tag{6}$$

As an example, the time to transmit a symbol at bandwidth 125kHz and SF10 is 8.19ms. Each Hertz (Hz) of bandwidth is equivalent to one chip per second [43] section 4.1.1.5. From the application note about LoRa modulation [39] we find the chip rate Rc in (7) given the symbol rate Rs in symbols/sec:

$$Rc = Rs * 2^{SF}\ chips/sec \tag{7}$$

The documentation [39] defines a RateCode as function of the code rate (CR) 1 through 4, where CR 1 maps to a cyclical coding rate of 4/5, CR 2 for 4/6, CR 3 for 4/7, and CR 4 for 4/8. We then define in (8) a RateCode:

$$RateCode = \frac{4}{4 + CR} \tag{8}$$

Notice that RateCode has possible values 4/5, 4/6, 4/7 and 4/8 for code rate values for CR values of 1 through 4. Also, from the modulation basics document [39] we find in (9) the nominal bit rate Rb

$$Rb = SF * \frac{RateCode}{\left[\frac{2^{SF}}{BW}\right]}\ bits/sec \tag{9}$$

As an example, at SF=7, BW=500kHz, and CR=1, nominal bit rate Rb is 21.9kbps. LoRaWAN requires an explicit header on the LoRa packet [40] section 3.1, which means that LoRaWAN cannot run at SF=6, because SF6 is not able to support explicit header [43] section

4.1.1.6. At a SF=12 and coding rate of 4/5, the nominal bitrate Rb ranges from 18bps at a bandwidth of 7.8kHz to a bitrate of 1172bps at a bandwidth of 500kHz. There is a variable called *LowDataRateOptimize* that affects the time on air. When asserted, *LowDataRateOptimize* improves the robustness of the transmission over variations in frequency of the transmitter and must be used in LoRa for symbol durations longer than 16ms. For (7) below, CR is the code rate 1 through 4, PL is the number of payload bytes from 1-255, SF is spreading factor (6-12), IH=0 if header is present and IH=1 if header is absent; DE=1 if Low Data Rate is optimized and CRC is 1 if the 16-bit payload CRC is present and 0 if it is not. For (10), (11), and (12) below, the total number of symbols *Nts* in a LoRa packet is the sum of the number of preamble symbols *Npre_s* (8 for LoRaWAN) and the number of packet symbols *Nps*:

$$Nts = Npre\_s + Nps \tag{10}$$

$$UB = \left\lceil \frac{(8PL - 4SF + 28 + 16CRC - 20IH)}{4(SF - 2DE)} \right\rceil \tag{11}$$

$$Nps = 8 + \max\left(ceil\,(UB) * (CR + 4), 0\right) \tag{12}$$

To assess the largest value for number of packet symbols *Nps* from (8), for value PL we assume a maximum payload of MACPayload size M of 230bytes from DR13 as given in [42] section 2.5.6 for US. Spreading factor 6 has the highest data rate possible for LoRa and is used with implicit header mode only [43] section 4.1.12, which means the header is left out of the transmitted packet. For other SFs, the header may optionally be included as needed, and the header is mandatory for LoRaWAN. A LoRa packet has up to 3 parts which are a preamble, header, and a data payload [43] section 4.1.1.6. In the receiver, the preamble length can be configured identically to the transmitter preamble length if known; for LoRaWAN preamble is 8 symbols.

The header in explicit header mode carries the number of bytes in payload, the forward error correction (FEC) code rate, and the presence of a CRC in the payload. The header has a CRC

to support invalid header detection. The header is always transmitted at FEC rate 4/8 [43] section 4.1.16. For the case of implicit header, the length of payload, payload CRC presence and FEC rate must be properly preconfigured on both sides of the data link.

We now examine the details of the LoRaWAN link layer implementation as found in [41]. This is the formal link layer specification from the LoRa Alliance. This specification [41] provides three LoRaWAN classes, A, B, and C (not to be confused with battery use cases A, B, and C from the previous chapter of this dissertation.) Class A covers bidirectional end devices, in which each uplink transmission is followed by two short receive windows. The transmission slot is scheduled by each end device based on its own needs with a small random time basis and is considered ALOHA-type contention [41]. Uplinks are initiated by the end device, and any downlink from the server will have to wait until after the end device chooses to uplink [41]. The next LoRaWAN class is Class B, end devices with scheduled receive slots. These devices open more receive windows at times that are scheduled with the gateway. For Class B, the gateway knows when the end device will be listening for receive packets [41]. The final LoRaWAN class is Class C, end devices with the maximum number of receive slots. Class C devices continuously receive except when transmitting [41]. The definition of uplink packets is those that are sent from end devices to a network server, relayed through one or more gateways [41] section 3.1. Downlink packets are transmitted by a network server to exactly one end device through one or more gateways [41] section 3.2. When an end device detects a preamble during one of its receive windows, it should stay active until the downlink frame is demodulated and the end node determines if the receive packet is addressed to it [41] section 3.3.1. There is a different document that specifies uplink and receive data rates that is region specific [42]. The link layer specification requires that the duration of the receive window has to be long enough for the receiver to detect a download preamble,

starting at RECEIVE_DELAY1 or RECEIVE_DELAY2 after the uplink window, with a default of 1s for each delay [42] section 2.3. The same section mentions that the delay from uplink to first receive window can be changed to 15s for over the air activation (OTAA.) This specification requires that a network server start a downlink frame (referred to as a Class A downlink) at the beginning of either of the two receive windows if it has a downlink packet for the end device, even if this interrupts a Class B or class C downlink [41] section 3.3.5. We can expect in a situation where the network server has a downlink packet targeted for an end device that it will start the transmission either RECEIVE_DELAY1 or RECEIVE_DELAY2 seconds after the finish of that end devices transmission [41] section 3.3. There is a retransmissions backoff strategy [41] section 7. This requires uplink frames from end devices that require an acknowledgement from the network and that are retransmitted if they do not receive acknowledgement. These can be triggered by external events that affect many end devices. There is a required interval between the end of the first RX2 slot and the next uplink transmission to enforce the regional duty cycle regulations (for example 1%) and a stepdown duty cycle that limits from 1% in the first hour to 0.01% after the first 11 hours. The LoRaWAN specification mentions the maximum length of a PHY payload which is one octet for the MAC header (MHDR), up to M octets for MACPayload as specified in the regional document for example, for US, M ranges from 19 octets to 250 octets depending on the data rate DR0 through DR15 [41] section 2.5.6.

Looking into the Channel Activity Detection (CAD) feature found in LoRa radios, we examine the radio data sheet [43] page 44. The radio is looking to detect a LoRa preamble on the radio channel with good power efficiency. For LoRaWAN usage, the CAD duration is a function of SF and spans from 1.75 to 1.92 symbol times over the range SF7 to SF12, with a maximum duration of 15.2ms (see Table 3.) An important aspect of CAD as implemented in LoRa radios is

that it only is only sensitive to the LoRa packet preamble and will not indicate other portions of the packet. Therefore, for this chapter [32] we are obliged to use other mechanisms to assess channel occupancy, for example by noticing an increase at the receiver RSSI value based on the activation of the transmitter.

The Adaptive Date Rate (ADR) mechanism in LoRa is ordinarily requested by the end node application, and managed by the network server [32] page 7. Network-wide, the server manages power and data rate by sending MAC commands to end nodes that modifies the end node transmit power and data rate based on the received signal strength at gateways [32] page 3. The network server has an available command *LinkADRReq* that requests a rate adaptation of the end device, specifically by providing a transmit power and data rate from network server to end nodes [40] section 5.3. The end node requests ADR services from the network server by setting the ADR bit in the frame control byte FCtrl in the frame header FHDR in an uplink packet from the end node [41] section 4.3.1. For the end node, in addition to requesting ADR service from the network server and obeying the *LinkADRReq* command, there is another mechanism, Acknowledgement Limits (AL) [32] page 7 to implement ADR. The AL mechanism uses the ADR and ADRACKReq bits in the FCtrl field in the frame header FHDR in uplink packets [41] section 4.3.1. It also uses a fixed value *ADR acknowledge delay* set in the end node. This AL part of ADR is initiated from the end node. To start the AL process, the end node sets itself to lowest data rate and longest range [32] figure 4. The network server may decide to send a MAC command to set the end node to a much faster data rate. However, if conditions change, this faster data rate might not allow the end node to reach the gateway anymore, so AL must be used in waiting the acknowledgement delays, and upon no messages from gateway, the end node automatically drops data rate step by step. The

network server uses global information in ADR to get the optimal power/rate settings, and at the end node, if ACKs are not received, the node adjusts to longer range.

Generally speaking, this ADR mechanism is useful in constructing efficient low power wide area networks especially with multiple gateways [32] page 6. However, ADR as implemented has some shortcomings. If acknowledgements are not received from gateway to the end node, the response of the end node is to increase airtime which is a limiting factor for density of LoRa radios. Some other method to prevent collisions between end nodes will lead to greater density. Also, the white paper about ADR [32] page 14 mentions that mobile nodes cannot use static ADR because it is controlled by the network server, and by the time the ADR command arrives at the mobile node, the conditions will have changed and the ADR command will not work as intended. For mobile use, a different method is recommended called blind ADR [44]. Blind ADR suggests the for a node that is mobile or potential mobile (a pet is the example given), we schedule several planned transmissions at different data rates. For [44] figure 1, we see the example of the pet tracker that plans for one hour of operation, one transmission at SF12, two transmissions at SF10, and three transmissions at SF7. The key idea is that if the pet is moving outdoors, the shorter, more frequent SF7 transmission is a better choice owing to the smaller impact on battery capacity [44] table 2; whereas if the pet is deep indoors in a stable position, the SF12 will be a better choice for reaching the tracker [44] page 3. The above descriptions of the existing adaptive data rate mechanism demonstrate a simple approach for resource constrained nodes. Blind ADR does not allow for adjustment due to dynamic conditions. We are interested in improving the adaptive data rate approach for a dynamic warehouse environment that includes people and machinery which move around and create a situation of variable path loss. To the extent that these movements are periodic and predicable, a machine learning or pattern recognition

approach will enable the future prediction of path loss changes. These future predictions enable us to adjust the data rate in a way to overcome the variability in path loss. Alternatively, there can be interference sources in a warehouse environment, which might be in the form of mobile radios carried by personnel or mounted on moving vehicles. Both situations, namely variable path loss and moving interference sources, create a pattern of packet losses over time which can be analyzed by the long short term memory implemented in the network server.

3.3 Warehouse Radios Implementation

Now we discuss the mechanisms implemented in the server and in the nodes that support the advanced functionality such as LSTM predictions to improve channel status prediction. Let us explore the simple relationship between the *airtime* of a LoRa packet in seconds, the average time between node transmissions *AvgPeriod* in seconds, the number of nodes in the warehouse *NumNodes*, and the Date Extraction Rate (DER) which is the fraction of packets that successfully travel from node to gateway, from (13). To provide a specific example, at spreading factor SF7, code rate CR1, and bandwidth 500kHz, with average transmission period 273 minutes and number of nodes 30,000, the data extraction rate is 95%, which is the minimum acceptable for a good communication link. 95% DER means 19 out of 20 packets transmitted are successful, which is efficient use of the medium. For this to work, all the 30,000 nodes must have their transmissions spread out over the 273-minute period evenly, which means that on the average, there is a quiet time of no transmission for 546ms followed by an airtime of 14ms. (14) starts with (13) and solves for $AvgPeriod$. For the case where the DER is known such as 95%, the number of nodes is fixed, and the airtime is known because it is tied to the data rate. Then, (15) is presented, which is the more generalized version of (13).

$$(1 - \frac{2 * airtime}{AvgPeriod})^{NumNodes} = DER \tag{13}$$

$$AvgPeriod = \frac{2 * airtime}{1 - 10^{log10(DER)/NumNodes}} \tag{14}$$

$$\prod_{k=0}^{MaxDataRate} (1 - \frac{2 * airtime_k}{AvgPeriod_k})^{NumNodes_k} = DER \tag{15}$$

Considering (15), when there are multiple radios in the warehouse operating simultaneously on different airtimes, we iterate k through all the data rates used, from 0 to 13. For each of these 14 data rates, there will be an airtime associated with the data rate, and as well an average period, and a certain number of nodes in the warehouse using the given data rate.

3.4 RSSI Machine Future Prediction Approach

For performing a LSTM channel status prediction, we may start out by gathering received signal strength (RSSI) data from the gateway LoRa radio. As an example, we show an example of the variable RSSI that is caused by motion of a objects in the environment. Referring to Figure 10, we find a data set taken in the engineering lab building at University of Michigan Dearborn, with people moving in the space to provide variability in the path loss. There are 2520 data points with 2 seconds between each reading. For the 84-minute trace shown, the mean RSSI is -85dBm with a standard deviation of 3.27dBm.

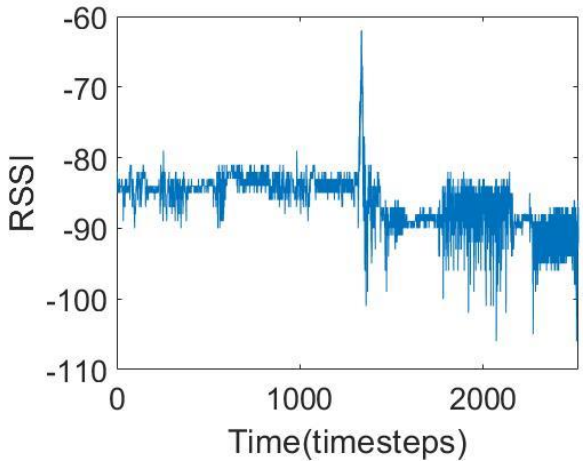


Figure 10 RSSI vs. time from experiment with LoRa radios and moving objects

The innovation for this chapter to improve the channel status prediction for LPWAN radios is called RSSI machine future prediction (RMFP). A long short term memory [45] is used to predict future values of RSSI for the radio channel of interest. The gateway records the RSSI value over a representative period, for example 84 minutes for a specific example as shown in Figure 10. The example data set from Figure 10, consisting of RSSI samples, is broken up into two sets as is typical for machine learning work, into a training set of the first 2000 points, and a test set of the remaining 520 points.

The gateway applies the RMFP technique. Our technique is to train an LSTM to model future values of the RSSI to help the node avoid collisions. We require an automatic mechanism to perform the training for us, which is considered unsupervised learning. This mechanism must take the data in Figure 10 and identify short regions from data trace that we can consider representative of the whole sequence. For this RSSI sequence, peaks can be identified. We employ a suitable technique for defining a threshold level, for example the mean value of the RSSI plus one standard deviation. We only use peak values above this threshold for further steps. For the

selected peaks, we assess the width of each peak at half prominence. We wish to find the time offset in the data sequence from Figure 10 for these peaks and as well, the peak width for each. In our example, we will choose widths that are above the mean width plus 1.25 standard deviations. By this threshold choice, we obtain a selection of relatively larger peak widths. This offset and width describe a sample of data taken from the given RSSI time series. The multiple peaks from the time series provide multiple data samples, with variable length based on the peak width. The combination of each peak value and the peak width describe a short time series taken as a subset of the Figure 10 data set. We apply an unsupervised machine learning technique, K-means clustering on two features, the peak values, and widths. Figure 11 shows the results of this K-means clustering. For this example, we find five peaks that meet our criterion, and we notice that k=1 is the largest category with 3 peaks close to each other in the feature space of peak height and peak width. This unsupervised clustering provides interesting information, as later we will find the clustered values have similar performance in the LSTM training task. Notice in Figure 11 that the legend, labeled one through three, indicates the shape for each of the clustered categories. For this clustering in Figure 11, category one has the most members and categories two and three are outlier categories with one member each. Even for this small experiment we can see the value that the K-means algorithm provides in grouping together data samples. We will see that these clustered values end up having similar performance.

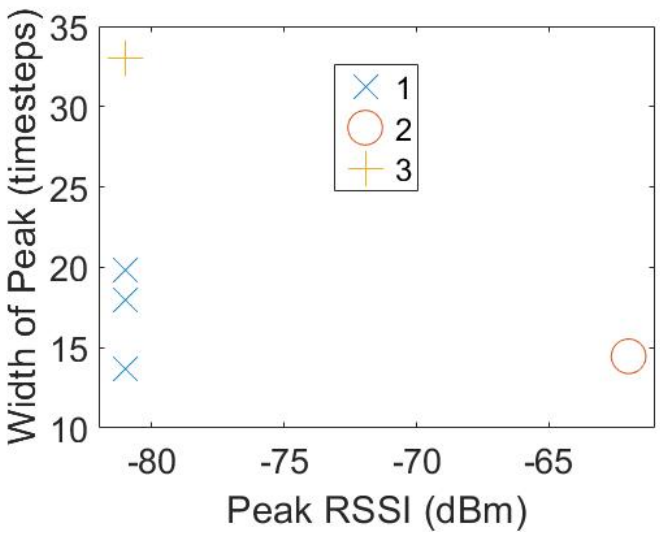


Figure 11 K-means clustering on peak RSSI and peak width

Our next task for RSSI machine future prediction will be to train multiple LSTM's, one for each of the data samples picked in the peak selection process just described. It is worth mentioning that every step up to this point is unsupervised, using automatic peak selection and K-means clustering for grouping into categories. We find that this automatic process has good success in finding areas of interest in the RSSI time series that we are examining, since the peak values tend to occur in areas of dynamic changes in the time series. Each of our trained LSTMs will consist of an input layer of one feature for our time sequence of RSSI values, an LSTM layer with 9 hidden units, a fully connected layer with one output, and a regression layer. We use a regression layer since we are using the LSTM to predict the future of a time series. We use the Adam optimizer, with 1000 maximum epochs for training. Some other parameters used are a gradient threshold of 1, initial learning rate of 0.005, piecewise learning rate schedule, learning rate drop period of 125, and learning rate drop factor of 0.2. One at a time, we take the short data samples that were automatically selected, and we create multiple trained LSTM models.

With LSTMs that have been trained on subsamples of the training set, we are now ready to select the best performing LSTM. Each LSTM has been trained on a short data sample provided by the peak selection process. To evaluate the best of these multiple LSTM's, we test on the full training set of 2000 data points from Figure 1. The new unseen data provides a good measuring bar of the quality of the multiple LSTMs. We find that the best performing LSTM coming from this part of the RSSI machine future prediction process will also tend to have good performance on data that it has never seen, the test set held back from the Figure 10 data set. Figure 12 gives a visualization of this competition between LSTMs.

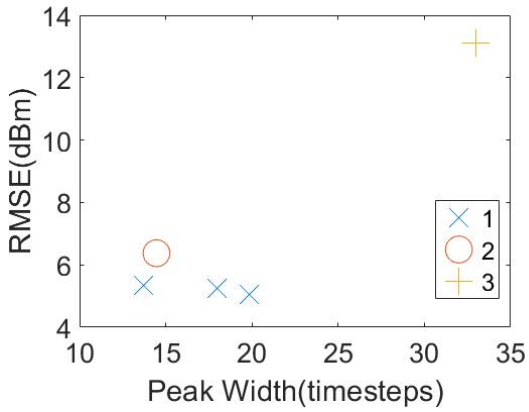


Figure 12 RMSE values for multiple trained LSTMs

In Figure 12, we focus on the item from feature space that has the most relevance to our training, which is the peak width. The reason why it matters more than the peak value is because the peak width determines the length of the data sample that each LSTM gets trained with. The peaks with higher width have a longer training sample since we do training on a data sample whose length is the same as the peak width, as part of the LSTM competition. Figure 12 gives the RMS error values of the multiple trained LSTMs. The shape of each point represents the k value from

K-means clustering. For this case, category 1 has the lowest RMSE values. If there had been very large number of peaks in the earlier steps of RMFP, we could have saved processing effort by only processing width values from the dominant category (here, category one.) We see the mainstream category in this K-means clustering outperforms the outliers for RMS error.

We select as the single winning LSTM as the one that demonstrates the lowest Root Mean Squared Error (RMSE) value on the task of next step prediction on this 2000-point training set. At this point, we have down selected a single trained LSTM that has been trained for next step prediction of the RSSI time series from Figure 10. To finish the preparation of optimally training our LSTM, we take the single LSTM that scored the lowest RMSE and perform one more training step. We consider the steps so far as a pre-selection to identify the area of the training set with the best results. We start at the winning peak location and train with a data sample that is a multiple of the winning peak width, for example twelve times the peak width value. The logic is that the pre-selection process identifies a promising area of the data set, and this 'second training' portion takes more surrounding data in the vicinity of the peak. By this approach, we are still working with a small subset of the overall training set, but through the peak selection process, we find the portion of the training set that shows the best results in next step prediction. Once a single LSTM is trained on this extended length data sample, we are ready to pass the trained LSTM from the gateway and network server down to the end nodes, that will use the trained LSTM for future prediction of the RSSI.

To tie this back to our warehouse example, this series of steps so far in RMFP can be readily carried out by our network server, by reading RSSI values at a gateway location and performing the steps given. The network server periodically picks time periods for monitoring the RSSI at the gateway location, to store the RSSI time series values in memory at the network server,

and then perform the RMFP steps. The LSTM design given has just 9 hidden units, and so does not take much memory for the description of the LSTM model. We arrange a plan to periodically download this trained LSTM model to the end nodes in the warehouse. To the extent that the warehouse environment is consistent in terms of number of users of the LPWAN radio channel and in the patterns of moving objects such as workers and equipment, there is not a need to transfer this model to end nodes very frequently. One update of the LSTM model every few months is sufficient for a stable environment.

The next step for RMFP assumes that the gateway has successfully transmitted the parameters of the final LSTM after the 'second training' process. With a trained LSTM model in the memory of each end node, we are ready to perform future prediction on the RSSI value of the radio channel.

3.5 RSSI Characteristics

It is worthwhile to examine the nature of RSSI in the warehouse application, since the RFMP process will provide future values of this quantity. The gateway needs each end node to transmit with a transmit power and a data rate that will be above the gateways' receiver sensitivity in every case. For portions of the warehouse that are close to the gateway and that have good link budget, this will not be a problem. We examine the path loss expression for LoRa radios as found in [29] and (16) where $Lpl(d)$ is the path loss at distance d in meters, $Lpl(d_0)$ is the path loss at reference distance d_0, γ is the path loss exponent and $\chi_\sigma \sim N(0, \sigma^2)$. Here, we can assume $Lpl(d_0)$ is 127dB at a reference distance of 40 meters, and γ is 2.08. In (17) Prx is power at the receiver in dBm, and Prx is power at the transmitter. GL is general gains and losses summed up.

$$Lpl(d) = Lpl(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + \chi_\sigma \quad (16)$$

$$Prx = Ptx + GL - Lpl(d) \quad (17)$$

From [29] section 3.2 in built up environments the reported range for LoRa radios is 100 meters. At full gateway power of 17dBm, the least sensitive end node can receive the message at this 100m distance. We apply this to the warehouse use case. [29] mentions that power levels higher than 17dBm can only be used on a 1% duty cycle, which will be an issue for the gateway; so, the link distance will be limited by transmission power on the gateway side to 17dBm, and in some scenarios, maximum power is +14dBm, which we will assume. From the assumptions in the simulator program, least sensitive data rate is -123dBm at SF=7 and -136dBm at SF=12. Also based on assumptions in the simulator, we assume a nominal GL value of 0, and χ_σ varies from -6dBm to +6dBm. With this level of performance, we have a working radio system that communicates fully at 160m distance between node and gateway, with a variable data rate depending on the value for χ_σ. With χ_σ at a value of +6dBm, our system only can communicate at data rate 0 with a sensitivity of -136dBm. For any other data rate, the receiver is not sensitive enough and packets are lost. Conversely, with χ_σ at a value of -6dBm, we may successfully communicate at any date rate between data rate 0 and data rate 13 from Table 3. It is possible for the path loss to fluctuate between -6dBm and 6dBm over the time we use the network. Of course, in a real warehouse, many items are in a constant state of movement such as personnel, vehicles, and material. From [29], the reported σ is 3.57dBm, and in our own environment in the lab building at U of M Dearborn, the σ was similar at 3.27dBm.

Figure 13 provides a view in the warehouse from the gateway to a receiver node B, whose receive sensitivity will be operating at either -136dBm (maximum sensitivity) or at -123dBm (minimum sensitivity.) χ_σ is set to either +6dBm (more path loss) or -6dBm (less path loss.) Node

A is always placed at distance Dist_A from the gateway to be able to run at the fastest data rate US DR13, which is to say that Node A is always placed so it can run at the minimum receiver sensitivity of -123dBm. For rows 1 and 2 of Table 4, all the nodes in Area A can run at the fastest data rate US DR13 while nodes outside Area A (but still within Area B) run slower than DR13, possibly as slow as DR0 for nodes at distance Dist_B. For rows 3 and 4 of Table 2., we see that Node A and Node B are at the same distance from the gateway, since our goal in these two rows are to keep all the nodes in the communication area running at the fastest data rate. In Table 4 below, the two different extreme values for χ_σ,+6dB and -6dBm, are covered for both nodes A and B in Figure 13 yielding 4 rows. For the given situation, each row contains the receiver sensitivity for nodes A and B, the χ_σ value.

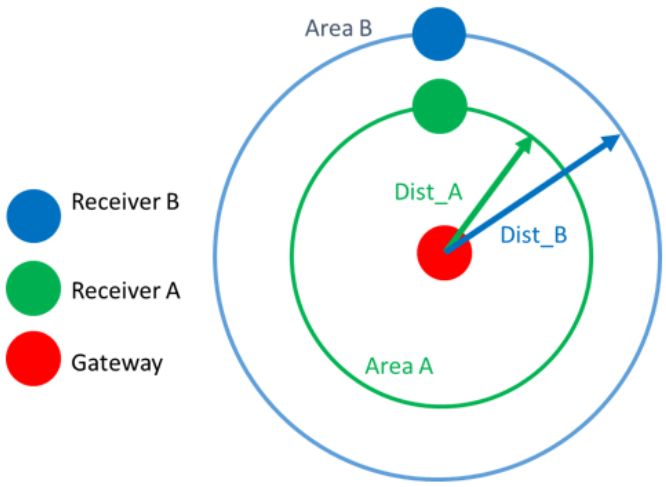


Figure 13 Nodes in warehouse at different distances from gateway

Table 4 Achievable communication distances by shadowing and data rate

Row	Sens Rcvr A (dBm)	Sens Rcvr B (dBm)	χ_σ (dBm)	Dist_A (m)	Dist_B (m)	Slowest Data Rate @B
1	-123	-136	6	160	350	DR0
2	-123	-136	-6	335	750	DR0
3	-123	-123	6	160	160	DR13
4	-123	-123	-6	335	335	DR13

We can use Figure 13 and Table 4 to plan our network use. The maximum density of LoRaWAN can be realized when all nodes are able to run at the maximum data rate DR13. From Table 4 we would prefer to be able to communicate with radios operating at the fastest and least sensitive data rate (Date Rate 13 with a sensitivity of -123dBm.) Rows 1 and 2 of Table 4 suggest one approach to manage the network that achieves maximum range of the LoRa radio in the warehouse, which is given as 750m in the case of lower path losses as in row 2. The range of the link drops to 350m as shown in row 1, when χ_σ is +6dBm. For Table 4, both rows 1 and 2 allow the slowest data rate DR0 to maximize the distance from gateway to node. This does not maximize the density of LoRa radios, since the by using (9) and an 8 hour transmit period we can only use 1800 nodes in the network and still achieve 95% DER. On the other hand, we can go use the alternative strategy shown in lines 3 and 4, that run at the fastest data rate DR13. The range is less, but we always use the fastest airtime. With lower range, it may be necessary to use a greater number of gateways. Using (8) and the same period we find that we can accommodate 30,000 nodes in the warehouse. This use case is the enabler for the higher density that we are looking for with LoRaWAN radios. To achieve the highest density of radios in our warehouse, we must minimize the airtime, which is to run the network at the highest data rate. In so doing, we operate the radio at its lowest sensitivity. With the radio at the fastest data rate, it becomes more important to avoid collisions, since there as a greater likelihood that the interference lasts the entire duration of the LoRa transmission. For example, when we run at DR13, the airtime is 19.52ms at CR4. By

comparison, the airtime is 493.6ms at DR0, about 25 times as long. The faster airtime we need to support higher density deployments will be less immune to interference, so it becomes important to add a mechanism to predict the interference and delay the transmission briefly when it is predicted. The channel status prediction method RMFP described in this paper provides the mechanism to avoid collision at the faster airtimes. To provide a visualization of the LSTM prediction as described, we take advantage of [46] for a sample of LoRaWAN RSSI data, taken at about 3 minute intervals for 128 data points. The trained output of an LSTM, as illustrated in Figure 14, start with the 100 samples of data shown on the left of Figure 14 and form LSTM predictions as shown in red.

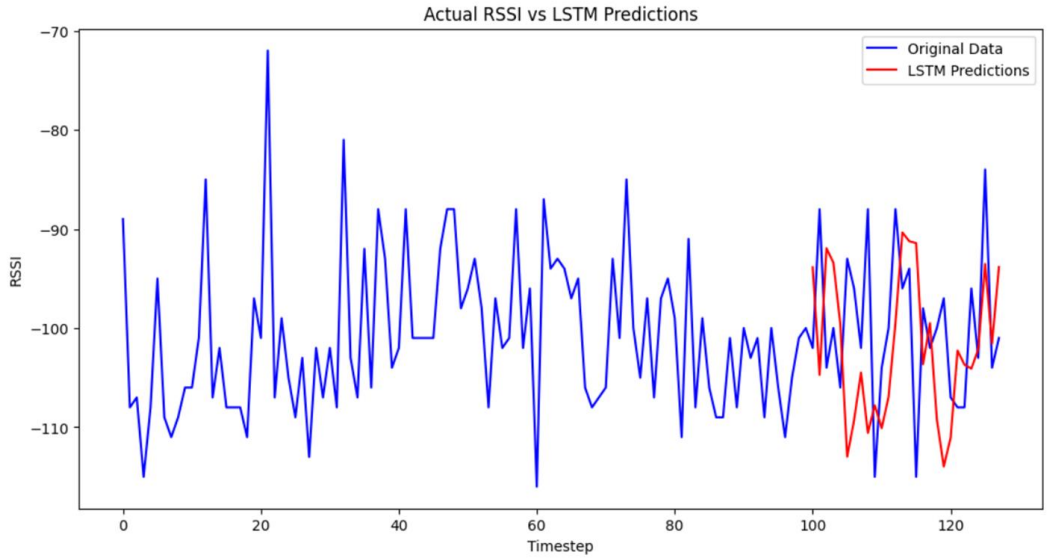


Figure 14 RSSI machine future prediction

Our approach for interference prediction is for the node to observe the value of the LSTM prediction of RSSI as demonstrated in Figure 14, and when it is over the mean value of the RSSI in the time window shown in Figure 10, the node delays one additional time step before transmitting. This procedure gives the node the ability to predict interference and avoid it.

3.6 Results

Referring to Table 5, we find the comparison of the new channel prediction approach against prior techniques for RSSI prediction. The use of Artificial Neural Networks (ANN) for RSSI prediction has been shown [22] to provide RSSI prediction accuracy of 6.2dB. An alternative approach where the variation in RSSI is reduced through filtering techniques [23] shows slightly better performance at 5dB. Another popular approach for RSSI prediction, Ray Tracing, in one study [24] also shows results of 5dB for prediction. For our experiments, the use of RSSI Machine Future Prediction as shown in this paper, for the training set shown in Figure 10 shows a next step RSSI prediction error of 2.6dB, better than the other methods shown.

Table 5 Results of RSSI machine future prediction vs. other methods

Approach	RSSI Prediction Accuracy
ANN	6.2dB
Reduction of RSSI variation through filtering	5dB
Ray Tracing	5dB
RSSI Machine Future Prediction	2.6dB

3.7 Conclusion

The channel status prediction technique as described in this paper, RSSI machine future prediction, shows good results at about half the error of the other approaches studied. Our approach can be used for the prediction of interference in the channel for LoRa radios. Our warehouse use case calls for a much larger number of end nodes, and the use of channel status prediction supports the higher data rate that is needed for the higher density of radios.

References

[1] N. C. F. Tse, W. H. Lau, and J. Y. C. Chan, "ZigBee based smart metering network for monitoring building integrated electric vehicle charging circuits," in *IEEE PES General Meeting*, PES 2010, 2010.

[2] S. M. Darroudi and C. Gomez, "Bluetooth low energy mesh networks: A survey," Sensors," in *Sensors*, Switzerland, 2017.

[3] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, " Wireless sensor networks: A survey on recent developments and potential synergies," *Supercomput.,* 2014.

[4] K. V Vidyanandan, " Batteries for electric vehicles," *Energy Scan: A House e-Journal of Corporate Planning, Power Management Institute, NTPC Ltd., New Delhi,* vol. 1, no. 38, June 2019.

[5] X. Feng, M. Ouyang, X. Liu, L. Lu, Y. Xia, and X. He, "Thermal runaway mechanism of lithium ion battery for electric vehicles: A review," *Energy Storage Mater.,* vol. 10, pp. 246-267, 2018.

[6] J. Xie, J. Li, J. Wang, J. Jiang, and C.-M. Shu, "Fire risk assessment in lithium-ion battery warehouse based on the Bayesian network," *Process Safety and Environmental Protection,* 2023.

[7] F. Cuomo, D. Garlisi, A. Martino, and A. Martino, "Predicting LoRaWAN behavior: How machine learning can help," *Computers,* vol. 9, no. 3, p. 60, 2020.

[8] M. Guerrero, C. Cano, X. Vilajosana, and P. Thubert, "Towards dependable IoT via interface selection: predicting packet delivery at the end node in LoRaWAN networks," *Sensors,* vol. 21, no. 8, p. 2707, 2021.

[9] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "Improving the sustainability of confirmed traffic in LoRaWANs through an adaptive congestion scheme," *IEEE Sens J,* vol. 23, no. 2, pp. 1660-1670, 2022.

[10] G. Kaur, S. H. Gupta, and H. Kaur, "Optimizing the LoRa network performance for industrial scenario using a machine learning approach," *Computers and Electrical Engineering,* vol. 100, p. 107964, 2022.

[11] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," *2017 IEEE International Conference on Communications (ICC), IEEE,* pp. 1-6, 2017.

[12] C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski, "Nhits: Neural hierarchical interpolation for time series forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence,* pp. 6989-6997, 2023.

[13] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437,* 2019.

[14] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186,* 2022.

[15] C. Challu, P. Jiang, Y. N. Wu, and L. Callot, "SpectraNet: Multivariate Forecasting and Imputation under Distribution Shifts and Missing Data," *arXiv preprint arXiv:2210.12515,* 2022.

[16] T. Stuart, F. Fang, X. Wang, C. Ashtiani, and A. Pesaran, "A modular battery management system for HEVs," *SAE Transactions,* pp. 777-785, 2022.

[17] S. Arora, A. Kapoor, and W. Shen, "Application of robust design methodology to battery packs for electric vehicles: Identification of critical technical requirements for modular architecture," *Batteries,* vol. 4, no. 3, 2018.

[18] "The Typical Warehouse - What it is, When it was built," 29 June 2020. [Online]. Available: https://www.cisco-eagle.com/industries-served/order-fulfillment/the-typical-warehouse.

[19] "GitHub - Nixtla/neuralforecast: Scalable and user friendly neural forecasting algorithms," 3 January 2024. [Online]. Available: https://github.com/Nixtla/neuralforecast.

[20] C. Pham and M. Ehsan, "Dense deployment of LoRa networks: Expectations and limits of channel activity detection and capture effect for radio channel access," *Sensors,* vol. 21, no. 3, p. 825, 2021.

[21] LoRa Alliance Technical Commitee, *LoRaWAN 1.1 Specification,* 2017.

[22] N. Raj, "Indoor RSSI Prediction using Machine Learning for Wireless Networks," in *Conference on Communcation Systems & Networks*, 2021, doi: 10.1109/COMSNETS51098.2021.9352852.

[23] C. C. Pu and W. Y. Chung, "Mitigation of multipath fading effects to improve indoor RSSI performance .," *IEEE Sensors Journal,* 2008, doi: 10.1109/JSEN.2008.2006453.

[24] F. Hossain et al., "Indoor 3-D RT radio wave propagation prediction method: PL and RSSI modeling validation by measurement at 4.5 GHz," *electronics*, 2019, doi: 10.3390/electronics8070750.

[25] L. Xiongfei, Z. Yunyi, and B. Liao, "LoRaWAN Anti-collision Algorithm Based on Dynamic Frame-slotted ALOHA," *Journal of Physics: Conference Series*, vol. 1486, no. 3, 2020, doi: 10.1088/1742-6596/1486/3/032045.

[26] R. S. Sinha, Y. Wei, and S. H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14-21, Mar. 2017, doi: 10.1016/J.ICTE.2017.03.004.

[27] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "Lora Digital Receiver Analysis and Implementation," in *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 2019-May*, 2019, doi: 10.1109/ICASSP.2019.868.

[28] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "A Review of LoRaWAN Simulators: Design Requirements and Limitations," in *IMITEC*, 2019, doi: 10.1109/IMITEC45504.2019.9015882.

[29] M. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?," 2016, doi: 10.1145/2988287.2989163.

[30] Team SimPY, "Website for SimPy," 08 January 2022. [Online]. Available: https://simpy.readthedocs.io/en/latest/index.html.

[31] T. and B. M. Voigt, "GitHub repository for LoRaSim," 10 July 2017. [Online]. Available: GitHub.

[32] Semtech Corporation Technical Paper, "Understanding the LoRa Adaptive Data Rate," Semtech, Dec. 01, 2019.

[33] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges," *Sensors (Switzerland)*, vol. 20, no. 18, 2020, doi: 10.3390/s20185044.

[34] LoRa Developer Portal, "LoRaWAN packet size considerations," Semtech, Jan. 08, 2022.

[35] A. Lavric and V. Popa, "Performance Evaluation of LoRaWAN Communication Scalability in Large-Scale Wireless Sensor Networks," *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018, doi: 10.1155/2018/6730719.

[36] K. Mikhaylov, J. Petäjäjärvi, and T. Hänninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *22nd European Wireless Conference*, 2016.

[37] L. Charles, B. Isong, F. Lugayizi, and A. M. Abu-Mahfouz, "Empirical Analysis of LoRaWAN-based Adaptive Data Rate Algorithms," in *IECON Proceedings (Industrial Electronics Conference)*, 2021-October, doi: 10.1109/IECON48115.2021.9589038.

[38] C. Lehong, B. Isong, F. Lugayizi, and A. M. Abu-Mahfouz, "A Survey of LoRaWAN Adaptive Data Rate Algorithms for Possible Optimization," in *2nd International Multidisciplinary Information Technology* , 2020, doi: 10.1109/IMITEC50163.2020.9334144.

[39] Semtech Corporation AN1200.22, "LoRa Modulation Basics," May 01, 2015.

[40] Y. A. Sornin, "LoRaWAN 1.1 Specification," Semtech, Oct. 11, 2017.

[41] LoRa Alliance, "LoRa Link Layer Specification TS001-1.0.4 LoRaWAN® L2 1.0.4 Specification," Oct. 01, 2020.

[42] LoRa Alliance, "LoRaWAN Regional Parameters," May 05, 2021.

[43] Semtech Corporation, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver Data Sheet," May 01, 2020.

[44] Semtech Corporation Technical Paper, "LoRaWAN Mobile Applications: Blind ADR," Dec. 01, 2019.

[45] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, 2000, doi: 10.1162/089976600300015015.

[46] "LoRaED Sample," Kaggle, 4 Feb 2019. [Online]. Available: https://www.kaggle.com/datasets/basemmelhalawany/loraed-sample. [Accessed 28 Feb 2024].

[47] "FAQ - Stein Service & Supply," 29 June 2020. [Online]. Available: https://steinservicesupply.com/information-center/faq/.

[48] "LoRaED Sample," Kaggle, 2019 4 February. [Online]. Available: https://www.kaggle.com/datasets/basemmelhalawany/loraed-sample. [Accessed 28 February 2024].