Estimation of the Probability of Successful Transmission of Uplink Messages in LoRaWAN

Ante Lojić Kapetanović

University of Split, Split, Croatia

May 10, 2019

This report is a short overview of applying forecasting methods on data from real LoRaWAN deployment in Svebølle, Denmark. The main idea is to examine whether there is a possibility of predicting successful transmission of LoRa messages for the given future time interval. Proposed model for prediction is Long Short-Term Memory (LSTM), an artificial recurrent neural network (RNN) architecture used in the field of deep learning.

1 Introduction

TBA

2 Network Topology

LoRaWAN - Long Range Wide Area Network - low power, wide area media access protocol designed to wirelessly connect 'things' to the internet. LoRaWAN is useful in IoT deployment, specially in smart environments where there are thousands of end-devices (i.e. sensors) that require minimal bandwidth and minimal energy.

The network architecutre is deployed in a star-of-stars topology in which gateways relay messages between end-devices and a central network server. [3] Toplogy overview is shown in Fig. 1.

End-device is transmitting data to base-station using LoRa technolgy, which means that data is modulated using CSS technique and transmitted using 868MHz RF, also it is worth mentioning that data is encrypted using AES128 in Counter mode (CTR). If the packet's FPort is set to 0 then the NwkSKey is used, otherwise the AppSKey is used. An important feature of all messages in LoRa is that the counters for sent (FCntUp) and received (FCntDown) messages are maintained by the Node and Network Server, and that these counters never repeat. [6] Base-stations relay messages from end-nodes to network server using a back-haul wireless network with higher frequency, often 2.4GHz, or using wired connection (Ethernet, optics). Network server decodes the packets received from base-station and performs security checks and adaptive data rate so it can generate the packets that should be sent back to end-devices. [3]

LoRaWAN network model deployed in Svebølle consisted of a single base station which communicates with few hundreds of different end-devices; network topology is quite simple and shown on Fig. 2

Data was collected during the period of 4 months and 14 days; during that period approximately 689 thousands measurements were generated and captured. There is only one base-station, and, judgdin by the exploration of the dataset 251 end-device in the network. Measurements were carried out on the base-station side and there are more than few relevant labels:

- Time - basic datetime format to microsecond precision (might want to treat this data as time series)
- DevAddr - device address written as HEX string (easily can be converted to binary if needed)
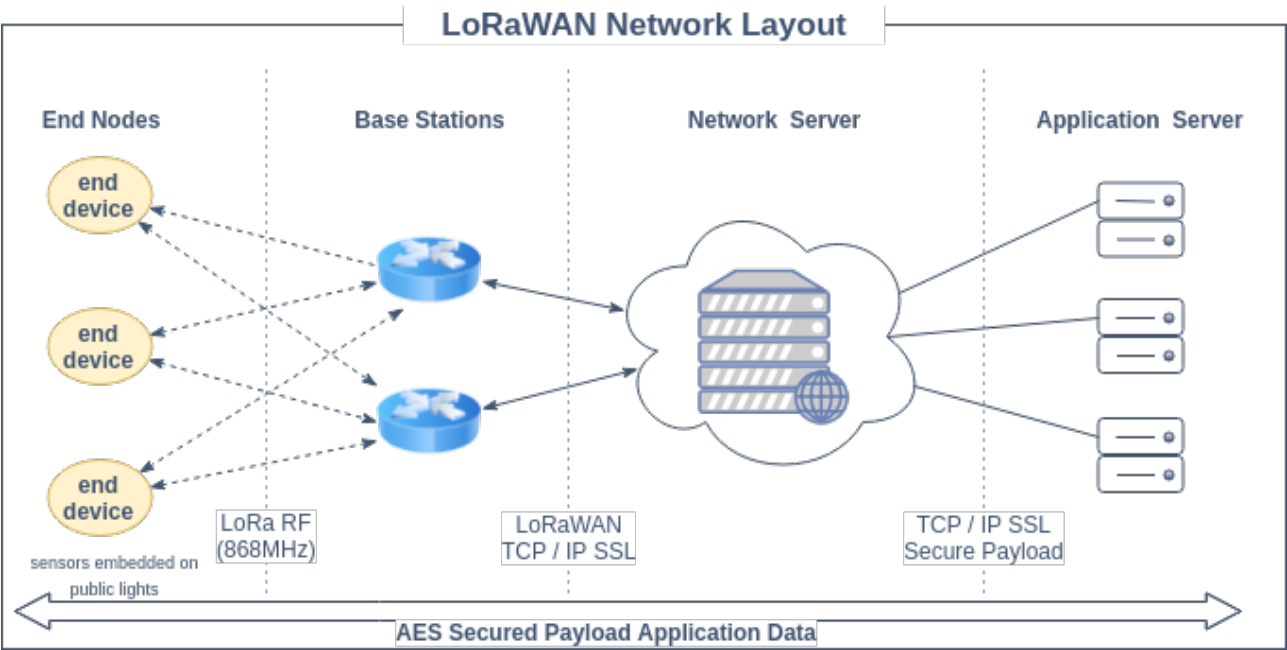- Freq - radio frequency (in MHz) of signals that carry information
- Chan - channel
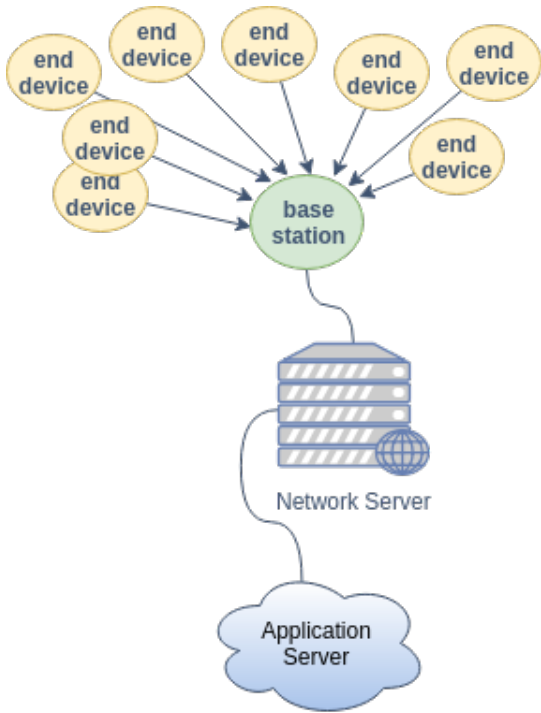
Figure 1: *LoRaWAN Network Layout*

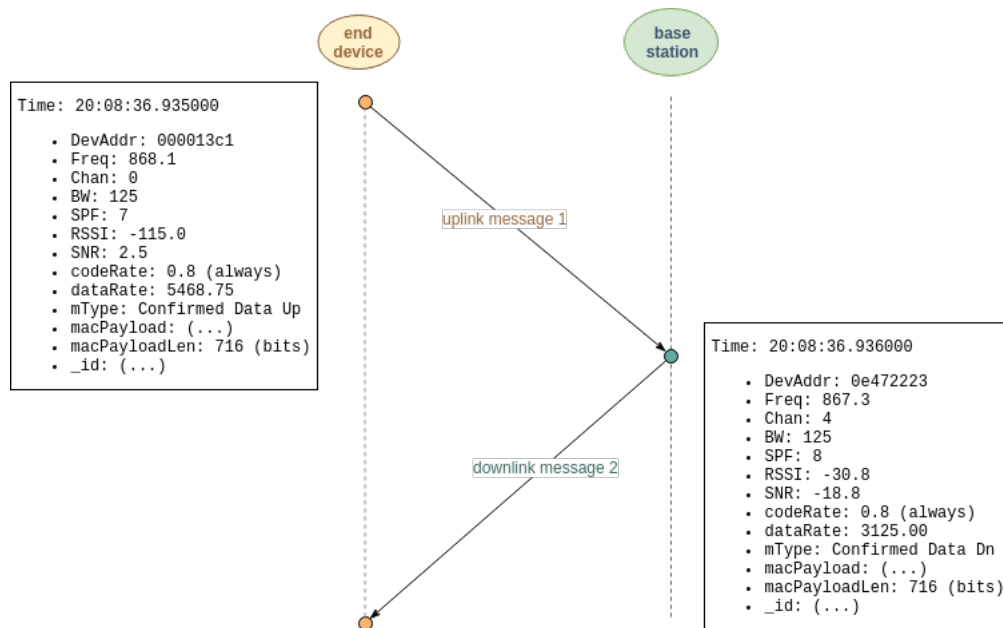**Figure 2:** *Overview of Svebølle LoRaWAN network*

Figure 3: *Communication between end-device and base-station in LoRaWAN network in Svebølle*

- BW - bandwidth (in kHz)
- SPF - spreading factor
- RSSI - received signal strength indication
- SNR - signal-to-noise ratio
- CR - code rate
- DR - data rate
- crcStatus - is there an error in transmission
- mType - type of MAC message
- macPayload - part of transmitted data that is the actual intended message
- id - identification of package

All of these indicators are described in details in [2].
Considering the data simple communication model of this LoRaWAN network could be deduced, Fig. 3

3 Introducing the Idea of ANN

Measurements in data set were not structured as time-series data, only times when transmission of LoRa message actually happend in uplink direction are captured and stored. Time series data is a series of data points indexed in time order and taken at successive equally spaced points in time. A time series is generally described through parameters such as trend, seasonality, irregularity and cyclicallity. [1]

In order to create time series data, the only important label is wheter the end-device has successfully transmitted the message to the gateway or not. For each time point (second-precision) there is activity flag assigned. If the device was active for observed time point, activity flag is 1 otherwise activity flag is 0. Since the measurements were carried out through almost 5 month period, newly created dataset was huge and as such perfect candidate for applying deep learning methods to perform forecasting of future time points.

Artificial Neural Networks, or ANN for short, are modeling the intelligence of human brain - recognize regularities and patterns of the given data, learning from past experience and providing inference on the output. [1] Why are the ANN such good predictors for time series type of data? There are couple of reasons:

Data-driven and self-adaptive There is no need to make *a priori* assumption about statistaical distribution of the data before feeding ANN with the data. [1]
Non-linearity ANNs are perfect for modeling data with non-obvious patterns. [8]
Universal functional approximators Any contionous function can be approximated to any accuracy. [5]

Even though vanilla ANNs are very powerful tool to create predictions based on historical events they lack persistence. Recurrent neural networks (RNNs) are trying to solve this problems implementing loops and allowing
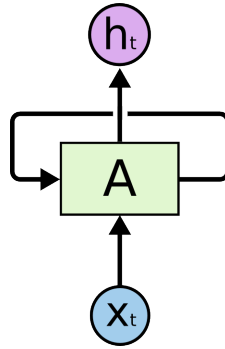
Figure 4: *RNN simple overview - feedback provides that the ouput (or a hidden layer) of the network, together with the new input at the current timestamp, is taken as input*
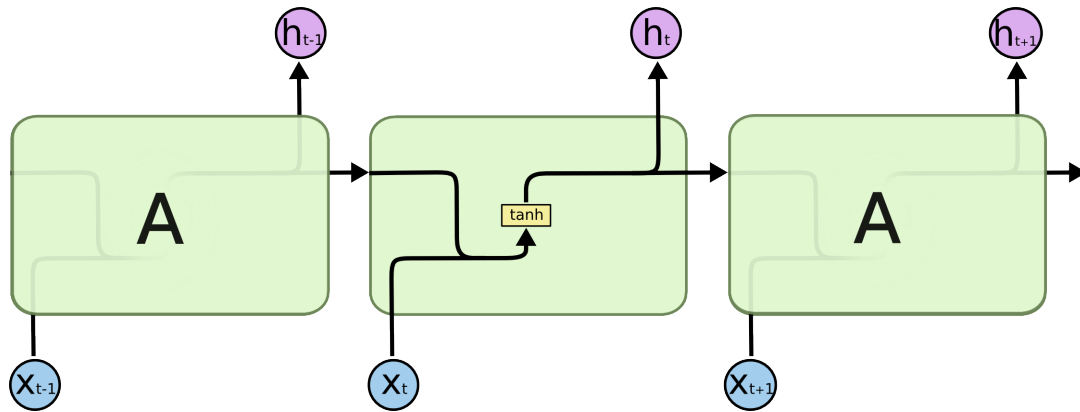

Figure 5: *RNN are formed in a chain-like nature what makes them applicable for predicting future events of sequence data*

information to persist., Fig. 5. RNN looks at the input x_t and outputs a value h_t, a loop allows information to be passed from one step of the network to the next.

Unpacked RNNs have the form of a chain of repeating nodes defined by very simple structure, Fig. **??**. [7]

RNNs are using gradient descent (first-order iterative optimization algorithm, usually used for finding the minimum of a function) as a way to minimize the error caused by changing each weight in proportion the derivative of the error with respect to that weight. Problem with this approach is error gradient vanishing which is causing the RNN to become unable to learn to connect the information if the relevant data is devided by large gap. The problem of Long-Term Dependencies is solved by applying Long Short-Term Memory network, LSTM for short, which are introduced by the Sepp Hochreiter and his team in 1997. in the [4]. They proposed a model that is capable of remembering relevant information for long period of time and avoid the long-term dependeny issues.

4 Proposed LSTM model

Just like the RNN, LSTM architecture is also formed as chain structure. Difference is that every cell has four in-layers each performing specific operations on input data and interacting with others in a special way, general overview of LSTM network is shown on Fig. 6.

An LSTM cell is consisted of 5 components which allow it to model both long-term and short-term data.

- cell state
- hidden state
- input gate
- forget gate
- output gate

Cell state, presented with the upper horizontal line in the Fig. 6, runs through the whole and is affected by some minor linear changes.

Gates are the mechanism that allows information to be passed in selective fashion from the LSTM to the cell state. They consist of a sigmoid layer, which outputs value between 0 and 1 describing value of each component, and a multiplication operation. There are three types of gates in LSTM network:
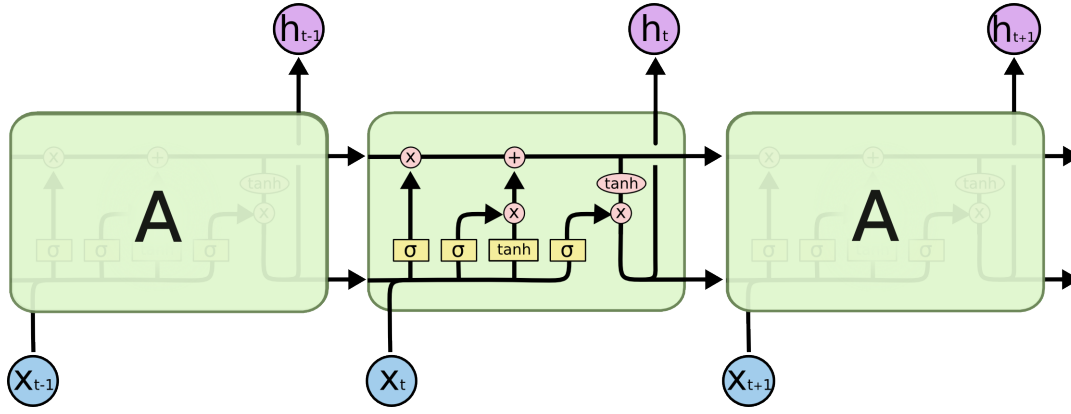
Figure 6: *LSTM network preview*

- Forget gate - determines which information to remove from the cell state

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- The second gate - decides which new information shoul be written to the cell state. First, the sigmoid function decides which values to update and then the *tanh* layer creates a vector of candidates to be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$C_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- The output gate - decides what is going to be output of the concrete cell. The output is based on the filtered version of the cell state.

$$o_t = \theta(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t)$$

References

[1] Ratnadip Adhikari and R. K. Agrawal. "An Introductory Study on Time Series Modeling and Forecasting". In: *LAP Lambert Academic Publishing* (Feb. 2013).

[2] Aloÿs Augustin et al. "A Study of LoRa: Long Range and Low Power Networks for the Internet of Things". In: *Sensors* (Oct. 2016).

[3] J. de Carvalho Silva et al. "LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities". In: (2017), pp. 1–6.

[4] Sepp Hochreiter and Jurgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* (Sept. 1997).

[5] Kurt Hornik, Maxwell Stinchcombe, and Halbert sWhite. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2 (1989).

[6] Robert Miller. "LoRa Security: Building a secure LoRa solution". In: *MWR Labs* (Mar. 2016).

[7] Christopher Olah. "Understanding LSTM Networks". In: *colah's blog* (Aug. 215).

[8] Guoqiang Zhang, B.Eddy Patuwo, and Michael Y. Hu. "Forecasting with artificial neural networks: The state of the art". In: *Elsevier BV* (July 1997).