Bayesian-based channel quality estimation method for LoRaWAN with unpredictable interference

Daichi Kominami
Graduate School of Information Science and Technology, Osaka University
Osaka, Japan
d-kominami@ist.osaka-u.ac.jp

Yohei Hasegawa
System Platform Research Labs, NEC Corporation
Kanagawa, Japan
y.hasegawa@nec.com

Kosuke Nogami
System Platform Research Labs, NEC Corporation
Kanagawa, Japan
nogami@nec.com

Hideyuki Shimonishi
System Platform Research Labs, NEC Corporation
Kanagawa, Japan
h-shimonishi@nec.com

Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
Osaka, Japan
murata@ist.osaka-u.ac.jp

***Abstract*—The "Internet of things" has become a common term, and low-power wide-area (LPWA) technology is attracting much attention as one of its elemental technologies. LPWA achieves wide-area communication without consuming much energy, allowing various data sensing and gathering applications. LoRa is an LPWA communication technology that uses unlicensed bands. Because it is possible to build a self-managed network with LoRa, many LoRa-based services will be scattered in the same area without an overall administrator. As a result, the communication performance of LoRa may degrade due to unintended radio interference. Unfortunately, many LPWA techniques, including LoRa, have low data rates, making it difficult to gather sufficient control information to avoid such degradation of communication performance. In this paper, we propose a method for estimating network congestion states through successive estimation using Bayesian updates of prior distributions. Computer simulations show the network state can be estimated by our proposed method with accumulating a little control information.**

***Index Terms*—Bayesian attractor model, human cognition model, state estimation, LPWA**

I. Introduction

Low-power wide-area (LPWA) networks, which realize low-power and wide-area communication, are rapidly attracting attention [1] because LPWA techniques facilitate the development of the Internet of things (IoT) by simplifying data collection from multiple users and devices. SigFox, a representative LPWA standard, has been rolled out in more than 65 countries as of 2020 [2]. Another representative standard, LoRa (long range), is available in more than 160 countries, and the LoRa Alliance was launched in 2015 and now has more than 500 member companies [3]. One company in each country is allowed to deploy a SigFox network service as its network operator, and users must use the public network operated by this operator. With LoRa, on the other hand, users can freely build their own networks by using products standardized by the LoRa Alliance.

When using LPWA, considering only power consumption of the wireless module, it is possible to operate communication equipment for several years, even with commercially available batteries. Regarding communication distances, unobstructed signals can be delivered to the gateway from about 10 km. By combining LPWA with a communication module that supports various sensing devices, it is possible to easily collect information from devices. All technologies realizing low power consumption and wide-area communication are combinations of existing technologies, but LPWA's applicability is very high, and it is therefore expected to advance IoT [4].

Most wide-area networks (WAN) using LPWA devices construct a star network consisting of a gateway and nodes, in other words, a many-to-one communication system between nodes and a gateway [1]. In particular, uplink communications from node to gateway are considered to be widely used in IoT applications, occupying most of the traffic. In this case, because nodes do not need to relay data, they can power the wireless module only when they want to transmit data, leaving it unpowered at other times. Such intermittent communication significantly reduces power consumption of the wireless module. LPWA is also designed to have a high link budget, which is one of the reasons it can achieve long-distance communication. Its elemental technologies include use of relatively low-frequency bands, a modulation method that is robust against interference and noise, and an antenna with high reception sensitivity. In particular, the data rate is designed to be relatively low (several hundred bps to several kbps) compared with conventional mobile networks.

However, this low data rate lengthens LPWA data transmission times. LoRa and SigFox use the ALOHA protocol for the MAC layer, and there are concerns that data frame collisions will increase as the number of nodes increase [5]. By confirming received signal strengths before transmitting signals, it may be possible to avoid collisions by detecting a carrier radio wave like that used in IEEE 802.11. However, if the IEEE 802.11 clear channel assessment (CCA) threshold of about -80 dB is used, radio signals with signal strengths below the threshold can reach the gateway due to the high antenna reception sensitivity of LPWA nodes, so collisions are likely, as in the case of using the ALOHA protocol. However, lower thresholds increase the probability that nodes judge the wireless channel to be busy, thereby increasing loss of transmission opportunities.

Users can use LoRa to build private networks with nodes and gateways [6]. Therefore, there will eventually be an environment where many private LoRa wide-area networks (LoRaWAN) will be constructed close to each other. This will increase the influence of interference due to the resulting increased number of nodes [7]. In addition to this interference, communication quality between nodes and their gateway will fluctuate over various time scales due to various causes, such as the existence of other systems using the same frequency band or the occurrence of obstacles. Multiple data rates and wireless channels are available in LoRa, and the gateway can cope with fluctuations in communication quality by assigning to nodes an appropriate data rate and wireless channel. However, to determine appropriate data rates or wireless channels according to changes in communication quality, the gateway must capture changes in communication quality and make decisions according to those changes at the right time.

In this paper, we propose a method by which LoRaWAN gateways can autonomously grasp changes in communication quality.

Here, wireless communication quality fluctuates temporally, but nodes do not always observe the communication quality of available wireless channels. Furthermore, the wireless resources available for information collection are limited, and it would take a very long time to gather sufficient information for performing any type of optimal control. Various studies have proposed methods for overcoming channel fluctuations in LPWA networks [8], but most approaches involve the physical layer, which is inflexible.

We therefore focused on the information recognition mechanism of the human brain, which performs appropriate inferences even when sufficient information is not available. It is known that in the process of information cognition in the human brain, there is a top-down type of information processing that makes decisions by comparing information input from various sensory organs with memories stored in the brain [9]. It has recently been reported that this series of information processes can be explained by a decision-making model based on Bayesian inference. In the Bayesian attractor model (BAM) proposed in Ref. [10], a hidden (decision) variable representing the decision state of the brain is defined on the state space, and this variable follows the dynamics with multiple attractors. In addition, a nonlinear function for converting the state space of the decision variable into the feature space is defined. Feature variables are defined in the feature space and each feature variable corresponding to each attractor corresponds to a memory in the brain, as described above. At this time, the Bayesian attractor model models neural information processing as follows: it (1) observes sensory information, (2) updates decision variables by Bayesian inference based on the observed information and the dynamics of the decision variables, and (3) make a decision. It is worth noting that the posterior probability distributions of the decision variables give us an appropriate timing of the decision making.

In this paper, we assume that multiple LoRaWANs exist in the same area. At this time, decision variables in the Bayesian attractor model are associated with the degree of communication congestion in the wireless channel. Figure 1 shows an overview of our proposal. Each attractor represents a different congestion degree, mapped to feature values at each congestion degree by conversion using the above-described nonlinear function. Our proposal is based on a Bayesian attractor model operating on a gateway (or a network server ahead of it), so feature values need to be observable by the gateway. The gateway periodically calculates the feature values based on communication with nodes, and inputs the calculated feature values to the Bayesian attractor model. The decision variable is then updated according to the input. By performing this for each wireless channel, it is possible to estimate the congestion degree of each wireless channel.

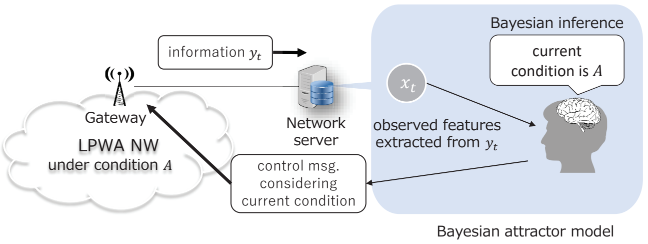


Fig. 1. Autonomous control loop with the Bayesian attractor model.

The remainder of this paper is organized as follows. In Section II, we show the detail of the Bayesian attractor model. In Section III, we present our extension of the BAM at first. And then we describe the definition of feature values and how we determine the memories embedded in attractors of the BAM. In Section IV, we show the performance of our proposal through the computer simulation. Section V gives the conclusion of this chapter.

II. Bayesian attractor model

The BAM uses a Bayesian estimation framework to model information perception and decision-making by the human brain. Our group has shown that the BAM can be used to reconfigure a virtual network in an optical network to create a virtual network suited to traffic situations [11], and that the BAM can be used for the rate control of the video streaming system [12].

The BAM represents the cognition in the brain by a state-space model where $\boldsymbol{z}$ is a state vector and $\boldsymbol{x}$ is an output vector. In the BAM, variables representing the internal decision state of the brain are defined as hidden variables that are updated according to known dynamics. The decision variable $\boldsymbol{z}$ approaches one of the attractors existing in the state space, due to the aforementioned dynamics. The BAM estimates $\boldsymbol{z}$ based on perceived information. However, since $\boldsymbol{z}$ is a hidden variable, Bayes' theorem is used to estimate $\boldsymbol{z}$.

A. Generative model

In the BAM, the decision variable $\boldsymbol{z}$ for the brain is represented as a random variable, and $\boldsymbol{z}$ is updated by nonlinear dynamics with K attractors. Given an initial state, $\boldsymbol{z}$ approaches one of the attractors as

$$\boldsymbol{z}_t = \boldsymbol{z}_{t-\Delta} + \Delta g(\boldsymbol{z}_{t-\Delta}) + \sqrt{\Delta}\boldsymbol{w}_t, \tag{1}$$

where $\boldsymbol{z}$ is a $K \times 1$ vector, Δ is the update interval, and $\boldsymbol{w}_t$ is a random number following a normal distribution $\mathcal{N}(0,\ q^2/\Delta)$, with q representing the magnitude of the process error included in the generative model. g is winner-take-all network dynamics, defined as

$$g(\boldsymbol{z}) = k(L\sigma(\boldsymbol{z}) + b^{lin}(\phi - \boldsymbol{z})), \tag{2}$$

where k is a constant determining the update scale, ϕ is a $K \times 1$ matrix, and all elements of ϕ have the same value ϕ_g. b^{lin} indicates the strength of a goal-state attractor. Also, $L = b^{lat}(I - \mathbf{1})$, where I is a unit matrix, $\mathbf{1}$ is a $K \times K$ matrix in which all elements are 1, and b^{lat} indicates the strength of lateral inhibition in the winner-take-all network dynamics. σ is a sigmoid function, $\sigma(z_i) = 1/(1+e^{-d(z_i-o)})$, and each element z_i of $\boldsymbol{z}$ is normalized to the range $[0\ 1]$. d determines the attenuation characteristic and o determines the position of an inflection point of the sigmoid function. By repeating dynamics g, only one element of $\boldsymbol{z}$ converges to ϕ_g. By setting $o = \phi_g/2$ and $b^{lat}/b^{lin} = 2\phi_g$, the other elements of $\boldsymbol{z}$ converge to $-\phi_g$. In other words, the K attractors in the dynamics of $\boldsymbol{z}$ are $K \times 1$, where only the ith element is ϕ_g and the other elements are $-\phi_g$ $(i = 0, \ldots, K-1)$.

In the BAM, each attractor of the generative model is associated with a feature vector representing past memory and experience. A feature vector $\boldsymbol{x}_t$ corresponding to a certain decision variable is generated as

$$\boldsymbol{x}_t = \boldsymbol{M}\boldsymbol{\sigma}(\boldsymbol{z}_t) + \boldsymbol{v}_t, \tag{3}$$

where $\boldsymbol{M} = [\boldsymbol{\mu}_0\ \boldsymbol{\mu}_1\ \ldots\ \boldsymbol{\mu}_{K-1}]$ is a feature matrix listing feature vectors, $\boldsymbol{\mu}$ is an $m \times 1$ vector, and M is an $m \times K$ matrix. $\boldsymbol{v}_t$ is a random number following the normal distribution $\mathcal{N}(0,\ r^2)$, where r represents sensory uncertainty.

B. Bayesian filter

The BAM estimates the decision variable $\boldsymbol{z}$ based on the predefined generative model and perceived information. Because $\boldsymbol{z}$ is a hidden variable and is updated temporally according to the generative model, a sequential Bayesian filter is used for estimation. Reference [10] uses a Bayesian filter called the unscented Kalman filter (UKF) [13]. While the general Kalman filter has poor estimation performance when dealing with nonlinear dynamics, the UKF mitigates these shortcomings, approximating the probability distribution by using a generative model and a small number of samples called "sigma points," which are calculated based on the estimated standard deviation.

Using the UKF gives the posterior probability distribution $P(\boldsymbol{z}_t|\boldsymbol{x}_t)$ of $\boldsymbol{z}$ at time t, allowing calculation of the probability density for each attractor $P(\boldsymbol{z}_t = \phi_n|\boldsymbol{x}_t)$. The authors of Ref. [10] refer to this probability density value as *confidence*, using it instead of marginal likelihood as a decision-making index in the BAM because the computational load required for marginalization increases exponentially when the number of attractors increases. When the confidence exceeds a predefined threshold λ, ϕ_n is determined as the estimation result. Here, ϕ_n is a $K \times 1$ matrix, and in this paper, the n-th and other elements are ϕ_g and $-\phi_g$, respectively (with $n = 0, \ldots, K-1$). Because UKF is used in the BAM, confidence is defined using the density function of the multivariate normal distribution. The BAM confidence thus exponentially decreases as the dimensionality of $\boldsymbol{z}$ increases; therefore, an appropriate value for λ must be carefully considered in advance.

III. Bayesian-based channel estimation Method

In this paper, we propose a method for estimating the degree of congestion in wireless channels. First, we describe our extension of the BAM to make it suitable for LoRaWAN applications. Then, we describe how we designed features and attractors of the BAM in our proposed method.

A. Extension of the BAM

From an engineering point of view, the BAM can be regarded as an estimation tool for determining coincidence between features observed from noise sources and features memorized in advance. However, there are some issues to address when applying the BAM in this way. In general, when the dimension of $\boldsymbol{z}$ exceeds the dimension of $\boldsymbol{x}$, estimation of $\boldsymbol{z}$ by $\boldsymbol{x}$ is an underdetermined problem; that is, solutions are not uniquely determined. In the BAM, $\boldsymbol{z}$ is updated by dynamics with attractors, so we can expect $\boldsymbol{z}$ to eventually converge to one of the attractors. However, the Kalman filter minimizes variance of the model error, so that once $\boldsymbol{z}$ is estimated at a position other than the attractor and the estimated variance at that time has a small value, it will stop at an equilibrium point other than an attractor. The confidence value is low at such points, and it is uncertain whether confidence exceeds the threshold.

Bitzer et al. verified characteristics of the BAM as an information processing model of the brain in [10]. In doing so, they did not assume situations where the dimension of $\boldsymbol{z}$ is equal to or larger than the dimensionality of $\boldsymbol{x}$, or where it becomes a sub-determination problem. Instead of the UKF we use a particle filter [14], which is a kind of Bayesian filter, to make estimations possible even in sub-determined cases. A particle filter is a state estimation method based on Bayesian estimation. Probability distributions required for state estimation are represented as sets of many particles rather than as mathematical expressions, and posterior probability distributions are approximated using weighted particles in the state space. The particle filter performs a sequential Monte Carlo simulation and weights particles based on the likelihood. By defining a likelihood function with lower values at non-attractor positions, we can expect $\boldsymbol{z}$ to have a higher probability of approaching attractors than staying at a non-attractor equilibrium point, as described above, and so the estimated result should be in the vicinity of one of the attractors.

The following shows the particle filter algorithm applied to the BAM in this paper. We assume that all particles p_i $(i = 0 \dots N_P - 1)$ are initialized in advance.

1) Update p_i according to Eq. (1): $p_i \leftarrow p_i + \Delta g(p_i) + w$, where we use a random number following $\mathcal{N}(0,\ s_{PF}^2)$ for each element of w in our evaluation.
2) Calculate the weight of a particle i as $W_i = P(y|p^i)$, where y is observed data. If the distribution of the observed value y is known, it is used for the likelihood function $P(y|p^i)$. If unknown, use an approximation with an appropriate distribution.
3) Calculate the weighted average of p_i as $\hat{\boldsymbol{z}} = \sum_{i=0}^{N_P-1} \mathcal{W}^{-1} W_i p_i$, where $\mathcal{W} = \sum_{i=0}^{N_P-1} W_i$.
4) Resample p_i using a sampling importance resampling (SIR) method [15]. In the SIR method, the current particle set is replaced by a new particle set that consists of N_P particles selected from the current particle set with probabilities proportional to their weight, W_i.

In general, increasing the number of particles N_P improves the estimation accuracy, but the calculation time also increases in proportion to N_P.

B. Design of features and attractors

Assuming a communication system using LoRa, we aim to achieve efficient communication even under unexpected changes in the quality of wireless communication. Various factors affect communication quality, such as increased numbers of private networks using LoRa, increased numbers of devices using the same wireless frequency band, or blocking structures that disrupt communication. Information with sufficient granularity and quantity is required to identify any such factors. However, as mentioned in Section I, LPWA communication systems have limited wireless resources for collecting control information.

If a feature value is unique when an event S occurs, we can estimate whether the occurring event is S by storing this feature value in the BAM. Although the event is not necessarily S for a feature, here we assume one-to-one relations between features and events. In addition, it is generally difficult to acquire feature values for a given event before it occurs. It is also possible that unforeseen events may occur during network operations, making it necessary to store new features in the BAM during operations. The problem here is that network systems can observe features, but not events themselves. We thus consider that network administrators should observe and set the event. We assume relations between features and events to be known and one-to-one; updates of the numbers of attractors and features is beyond the scope of this study.

Because users can build private LoRaWANs, multiple networks can coexist in the same area and congestion of wireless channels may change. In such cases, we use the BAM to predict changes in congestion for each wireless channel. The BAM stores features $\boldsymbol{\mu_i}$ representing the degree of congestion when there are N_i nodes, where i is the number of states to remember and is equivalent to the number of attractors. In this paper, we use data reception and data decode success rates, which are realistically available to the gateway, as feature variables. These values are periodically calculated for the previous interval and input to the BAM. The data reception rate is obtained by dividing the number of data points actually received by the gateway and successfully decoded during the interval by the expected number of received data points. We assume that the gateway has a known data transmission schedule for each node. The data decode success rate is the percentage of data points received during the interval that was successfully decoded. In addition to these two rates, we use the ACK reception rate, the rate at which a node does not receive an ACK after data transmission. Note that the gateway cannot directly observe this rate; therefore, we assume that the gateway can approximate this rate by storing in headers of node-transmitted data information showing whether an ACK was received for the last data transmission. We use these three rates because they show different characteristics, the first two representing degrees of congestion around the gateway and the third representing degrees of congestion around nodes. Figure 2 shows an overview of our proposal.

IV. Results

A. Simulation settings

In this section we describe a simulation for evaluating our proposal in a LoRaWAN scenario. Two hundred LoRa nodes and one gateway are installed in a 5×5 km^2 area. The x and y coordinates for a node are determined according to uniform random numbers in the range 0–5 km, with the gateway set at $(0, 0)$. Each node periodically generates data at 5-min intervals. The data generation timing is asynchronous among nodes, with a uniform random timing of -2.5 to 2.5 s added to the interval to avoid continuous data collisions. The number of wireless channels available for nodes is set to 4 (c_1, c_2, c_3, c_4), and the number of nodes using each wireless channel is set to 50 at the start of the simulation.

Data and ACK frame sizes are 50 and 10 bytes, respectively, and the data rate is 1.5 kbps. The nodes and the gateway perform 5 ms carrier sensing before transmitting data and an ACK frame. The CCA threshold is set to -83 dB, and frame transmissions are cancelled when a signal using the same channel is detected. In data transmissions, nodes retransmit data only once in total when a wireless channel is detected to be busy by carrier sensing or when the gateway does not return an ACK.

In the signal propagation model, we consider only direct waves as given by the Frisian transfer formula. The frequency used here is 920 MHz, the attenuation coefficient is 2.5, the antenna reception sensitivity of the nodes and the gateway is -131 dB, the transmission power is 13 dB, and the gain of the transmitting and receiving antenna is 5 dB. For data and ACK, decoding errors are stochastically generated according to the signal-to-noise ratio (SNR) at the time of reception. We used a simple decoding error model in this study, because the bit error rate will vary greatly in actual applications, depending on the environment. Specifically, the bit error rate is 100% when SNR is 0 dB or less, 50% when SNR is 0–5 dB, 10%

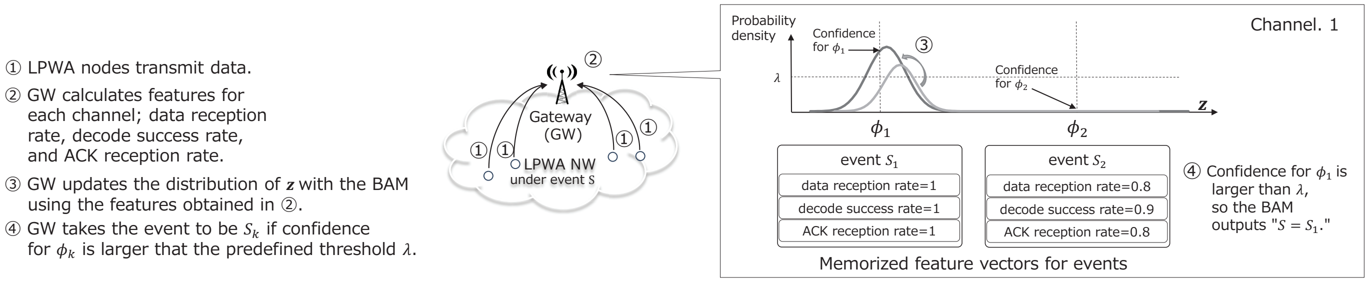

① LPWA nodes transmit data.
② GW calculates features for each channel; data reception rate, decode success rate, and ACK reception rate.
③ GW updates the distribution of $\boldsymbol{z}$ with the BAM using the features obtained in ②.
④ GW takes the event to be S_k if confidence for ϕ_k is larger that the predefined threshold λ.
②
Gateway (GW)
①
①
①
①
LPWA NW under event S
Channel. 1
Probability density
Confidence for ϕ_1
③
Confidence for ϕ_2
λ
z
ϕ_1
ϕ_2
event S_1
data reception rate=1
decode success rate=1
ACK reception rate=1
event S_2
data reception rate=0.8
decode success rate=0.9
ACK reception rate=0.8
④ Confidence for ϕ_1 is larger than λ, so the BAM outputs "$S = S_1$."
Memorized feature vectors for events

Fig. 2. Overview of our proposal.

TABLE I
BAM PARAMETERS.

Parameter	Value	description
b^{lat}	1.7	strength of the lateral inhibition
b^{lin}	$b^{lat}/20$	strength of a goal state attractor
g	10	distance factor between attractors
r	0.7	slope of sigmoid function
o	g/2	center of sigmoid function
Δ	0.004	time difference
k	500	scale of dynamics

when SNR is 5–10 dB, 1% when SNR is 10–20 dB, and 0% when SNR is 20 dB or more.

The total simulation time is 400 min. When 200 min has passed in the simulation, 50 additional LoRa nodes are added at random positions only on the specific wireless channel (c_1). These added nodes do not belong to the same network as the above-mentioned nodes and gateway. They generate data at the same interval (5 min), but the gateway does not return ACK messages to them.

Table I shows parameters for the BAM. We set k for the BAM to 500, a larger value than that used in [10]. This is because the particle filter is based on randomly generated particles, increasing the variability of $\boldsymbol{z}$ itself. The speed and accuracy of the estimation depend on some parameters, and here we focus on s_{PF} that determines the degree to which the particles spread. We set $s_{PF}^2 = 2.5$. It is also necessary to determine the number of particles and the likelihood function. In this evaluation, the number of particles is set to 1,000. 1,000 particles are enough to estimate the distribution of $\boldsymbol{z}$ that has three attractors, but when the number of the attractor, K get larger, more number of particles is needed.

For the likelihood function L of particle p_i, a multivariate normal distribution is used as an approximate distribution of observed values. Then, $L(y|p_i) = \mathcal{N}(y - M\sigma(p_i), 2*\Sigma)$ and Σ is a variance-covariance matrix of observation. The variance-covariance value in $L(y|p_i)$ is set larger than those for feature observations to make the estimation robust. We use a variance-covariance matrix of features when the number of nodes using a given wireless channel (c_1) is 100, which is obtained by simulation in advance. The actual value of Σ used in the simulation is

$$\Sigma = \begin{pmatrix} 0.0043 & 0.0033 & 0.0058 \\ 0.0033 & 0.0037 & 0.0063 \\ 0.0058 & 0.0063 & 0.013 \end{pmatrix}, \tag{4}$$

where Σ_{11} is the variance of the data reception rate, Σ_{22} is the variance of the data decode success rate, and Σ_{33} is the ACK reception rate. As the degree of congestion, we derive features in advance by a simulation in which the number of nodes using a given wireless channel (c_1) is 50, 100, and 150. The features are stored in the attractors $\phi_1,\ \phi_2,\ \phi_3$. The feature matrix M used in the simulation is

$$M = \begin{pmatrix} 0.98 & 0.97 & 0.91 \\ 0.95 & 0.94 & 0.83 \\ 0.91 & 0.90 & 0.76 \end{pmatrix}, \tag{5}$$

where the first column is the data reception rate, the second column is the data decode success rate, and the third column is the ACK reception rate. The first, second, and third rows show values when number of nodes is 50, 100, and 150, respectively.

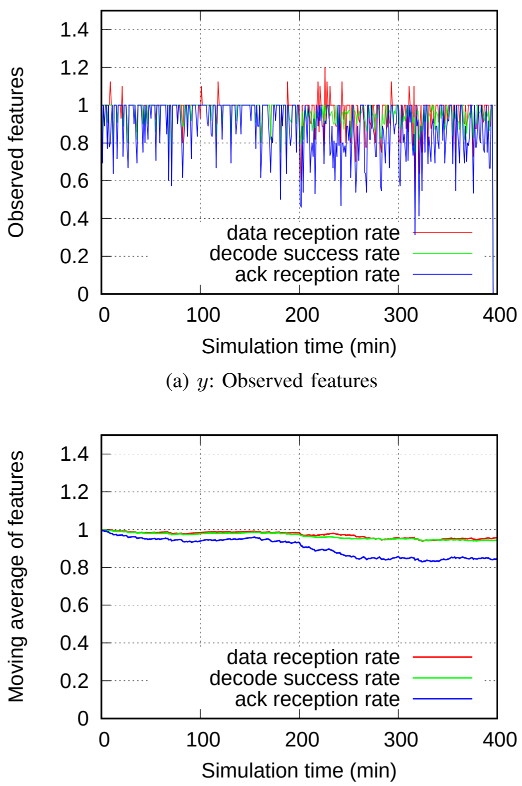
The gateway calculates features (the data reception rate and data decode success rate) each minute, inputting them to the BAM immediately after calculation.

B. Channel quality estimation results

Figure 3 shows a sequence of input features y and an exponential moving average of $\bar{y}$ obtained in one LoRaWAN simulation trial. $\bar{y}_t$ is calculated as $(1-\alpha)\bar{y}_{t-1} + \alpha y_{t-1}$, setting α to 0.02. First of all, it can be seen that the observed features fluctuate with time. The use of moving averages smooths out this fluctuation, but does not allow us to estimate the degree of congestion.

We show the change in confidence of the BAM when the features y_t obtained from the LoRaWAN gateway in Fig. 4 are observed every minute. As the BAM repeats observations, confidence for ϕ_1, which corresponds to the current congestion degree, increases and stabilizes. Although this confidence may temporarily fluctuate, it remains sufficiently large compared with those for ϕ_2 and ϕ_3. After 200 min, when the congestion changes, the confidence corresponding to ϕ_1 gradually decreases and the confidence corresponding to ϕ_2 increases, with that for ϕ_2 exceeding that for ϕ_1 at 248 min. The proposed method can appropriately estimate the posterior distribution by repeating observations with Bayesian estimation, and the degree of congestion can be estimated using the confidence.

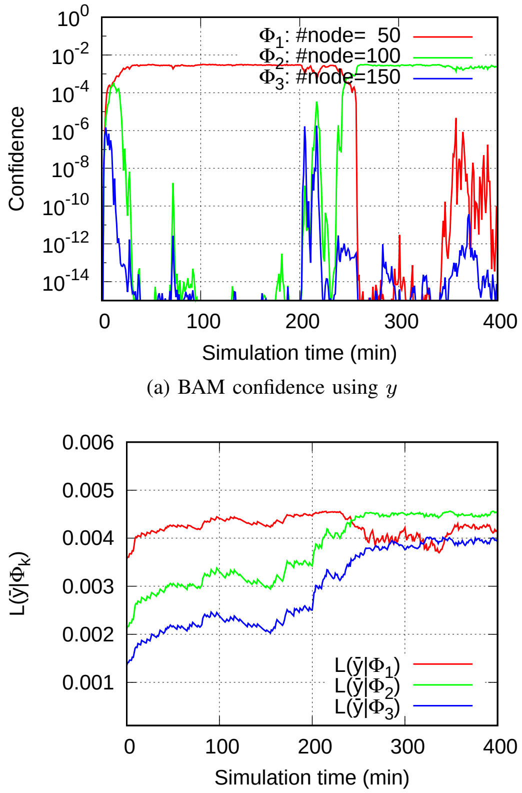
We also show the change in the probability density of $\bar{y}$, calculated as $L(\bar{y}|z = \phi_k)$ with $k = 1,\ 2,\ 3$. Since L is defined with the distribution of the observed features measured in advance, it well grasps the network state. The



(a) y: Observed features

(b) $\bar{y}$: Moving average of y

Fig. 3. Input values used in Fig. 4.



(a) BAM confidence using y

(b) Probability density of $\bar{y}$

Fig. 4. Results of sequential state estimation.

TABLE II
ESTIMATION TIME AND ACCURACY.

Method	Time	Accuracy
BAM ($s_{PF}^2 = 1.8$)	83.7 min	86.6 %
BAM ($s_{PF}^2 = 2.5$)	61.3 min	79.0 %
Moving average ($\alpha = 0.010$)	103.5 min	90.5 %
Moving average ($\alpha = 0.015$)	70.7 min	80.6 %
Moving average ($\alpha = 0.020$)	53.0 min	75.3 %

probability density $L(\bar{y}|z = \phi_1)$ is highest up to 200 min, and can therefore be used to estimate the degree of congestion. However, there is a problem that after 200 min, the values of $L(\bar{y}|z = \phi_1)$, $L(\bar{y}|z = \phi_2)$, and $L(\bar{y}|z = \phi_3)$ get closer, which makes it difficult to set an appropriate threshold in advance for the naive probability density.

Table II summarizes the estimation time and accuracy of the two methods described above as average values over 50 simulation trials. The estimation time is the time required to first estimate the degree of congestion corresponding to ϕ_2 after 200 min. After the estimation result outputs this value, we calculate the rate of time for which the confidence of ϕ_2 is highest, defining this as the estimation accuracy.

It can be seen that the BAM could adjust the estimation time and accuracy by changing the value of s_{PF}. This is similar to changing α of the moving average. Both of them use the same function L, so the performance is almost the same level. The BAM can make a decision based on confidence, and by changing s_{PF}, the estimation time and the accuracy of the estimation can be flexibly adjusted. This is an important advantage of our method for applying it to various control methods.

V. Conclusion

In this paper, we proposed a method for estimating the degree of wireless channel congestion in LoRaWAN using the Bayesian attractor model (BAM), which is based on a human cognitive mechanism model. By using the confidence output from the BAM, we can flexibly determine the timing of decision making according to the amount of observed information and the magnitude of fluctuation in observed information. To make the BAM easier to apply to LoRaWAN scenarios, we used a particle filter instead of the unscented Kalman filter that was adopted in the original BAM work. Using the BAM, our proposal can predict changes in the degree of wireless channel congestion with comparatively little information but without being greatly affected by temporary fluctuations in observed values. This should allow adaptive and stable control of a LoRaWAN system. In future research, we will combine our method with channel assignment control and compare it with other methods to demonstrate the advantages of the proposed method, especially in terms of communication performance.

Acknowledgment

This research was supported in part by “Grant-in-Aid for Challenging Exploratory Research 18H04096” of the Japan

Society for the Promotion of Science (JSPS).

References

[1] 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, 2017.

[2] “Sigfox,” (accessed Aug. 27, 2020). [Online]. Available: https://www.sigfox.com/en/coverage

[3] “LoRa alliance,” (accessed Aug. 27, 2020). [Online]. Available: https://lora-alliance.org/

[4] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: An overview,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[5] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovács, P. Mogensen, and M. Sorensen, “Coverage and capacity analysis of Sigfox, LoRa, GPRS, and NB-IoT,” in *Proceedings of IEEE vehicular technology conference (VTC Spring)*. IEEE, Jun. 2017, pp. 1–5.

[6] LoRa Alliance, “LoRa Specification V1.1,” Oct. 2017.

[7] K. Staniec and M. Kowal, “LoRa performance under variable interference and heavy-multipath conditions,” *Wireless communications and mobile computing*, vol. 2018, Apr. 2018.

[8] V. Berg, J.-B. Dore, and V. Mannoni, “Channel estimation strategy for LPWA transmission at low SNR: application to Turbo-FSK,” in *Proceedings of IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, May 2019, pp. 1–5.

[9] K. Kveraga, A. S. Ghuman, and M. Bar, “Top-down predictions in the cognitive brain,” *Brain and cognition*, vol. 65, no. 2, pp. 145–168, Nov. 2007.

[10] S. Bitzer, J. Bruineberg, and S. J. Kiebel, “A Bayesian attractor model for perceptual decision making,” *PLoS Computational Biology*, vol. 11, no. 8, p. e1004442, Aug. 2015.

[11] T. Ohba, S. Arakawa, and M. Murata, “Bayesian-based virtual network reconfiguration for dynamic optical networks,” *Journal of Optical Communications and Networking*, vol. 10, no. 4, pp. 440–450, Apr. 2018.

[12] M. Iwamoto, T. Otoshi, D. Kominami, and M. Murata, “Rate adaptation with Bayesian attractor model for mpeg-dash,” in *Proceedings of IEEE Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2019, pp. 0859–0865.

[13] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*. IEEE, Oct. 2000, pp. 153–158.

[14] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[15] A. F. Smith and A. E. Gelfand, “Bayesian statistics without tears: a sampling–resampling perspective,” *The American Statistician*, vol. 46, no. 2, pp. 84–88, 1992.