Internet of Things 25 (2024) 101016

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

LoRa-based outdoor localization and tracking using unsupervised symbolization

Khondoker Ziaul Islam [a,b], David Murray [a], Dean Diepeveen [c,d], Michael G.K. Jones [d], Ferdous Sohel [a,b,*]

[a] *School of Information Technology, Murdoch University, Murdoch, WA 6150, Australia*
[b] *Centre for Crop and Food Innovation, Food Futures Institute, Murdoch University, Murdoch, WA 6150, Australia*
[c] *Department of Primary Industries and Regional Development, Western Australia, South Perth, WA 6151, Australia*
[d] *School of Agricultural Science, Murdoch University, Murdoch, WA 6150, Australia*

ARTICLE INFO

Keywords:
LoRa localization
RSSI
Path loss
D-Markov machine
Chinese restaurant process
Maximum entropy partitioning

ABSTRACT

This paper proposes a long-range (LoRa)-based outdoor localization and tracking method. Our method presents an unsupervised localization approach that utilizes symbolized LoRa received signal features, such as RSSI, SNR, and path loss, where each symbol represents a system state. To identify the partitioning boundaries between the symbols in time series, we employ maximum entropy partitioning. The D-Markov machine is used to construct nondeterministic finite-state automata for extracting temporal patterns. We incorporate the Chinese restaurant process for online estimation, especially in scenarios with an unbounded number of probable areas around each LoRa gateway. An adaptive trilateration approach is then used to localize the target node from the estimated ranged radii of areas. The point-wise localization data was used for time-series continuous tracking. We collected a dataset using three LoRaWAN gateways, sensor nodes powered by single-use batteries, and a Chirpstack server on a sports oval. We thoroughly evaluated the proposed method from the perspectives of localization accuracy and tracking capability. Our method outperformed state-of-the-art machine learning-driven range-based and fingerprint-based localization techniques.

1. Introduction

Localization and tracking are of paramount importance in mobile wireless sensor networks as it enables spatial understanding of the sensor data, facilitate navigation, and support numerous real-world applications relying on location estimation. Localization involves determining the spatial position of an object, whereas tracking aims to determine the object's positions over time. The importance of localization and tracking within a defined region has been emphasized in several studies [1,2]. Enabling IoT-based localization and tracking while maintaining minimal infrastructure, cost-efficiency, and low power consumption is challenging [3,4]. Especially, the rural locations of large farms make it impossible to remotely monitor livestock or equipment due to poor or unavailable network infrastructure and poor signal coverage.

There are several wireless sensor network technologies now in use, including cellular, satellite-based, and low-power wide area networks (LPWANs). However, because of their high power consumption, global navigation satellite system (GNSS) is inappropriate for low-cost and low-power devices. On the other hand, high deployment and bandwidth subscription costs make cellular communications expensive and often unavailable in remote rangeland areas. A well-known wireless network architecture

[*] Correspondence to: School of Information Technology, Murdoch University, 90 South Street, Murdoch, WA 6150, Australia.
E-mail address: F.Sohel@murdoch.edu.au (F. Sohel).

https://doi.org/10.1016/j.iot.2023.101016
Received 24 August 2023; Received in revised form 11 November 2023; Accepted 24 November 2023
Available online 30 November 2023

that offers extensive connection for IoT devices is LPWAN [5]. LoRa stands out among LPWAN technologies as an affordable option that uses unlicensed ISM frequencies. It uses chirp spread spectrum (CSS) modulation to provide long-range capabilities and resilient communication in noisy settings. LoRa has become more prevalent in indoor and outdoor applications, such as sensing [6,7] and localization [8]. Constructing LoRa-based localization or tracking is a challenging and increasingly well-studied area [9]. Multilateration based on a precise radius is a simple tracking method. However, the timestamp precision of current commercial off-the-shelf (COTS) LoRa devices is insufficient [10], demanding additional work and software-defined radios (SDRs) to improve accuracy [11]. Alternately, target tracking can be accomplished by combining the time difference of arrival (TDoA) and angle of arrival (AoA) [9,12]. However, due to hardware restrictions, the AoA estimate is not supported by standard LoRa technology. Multiple-input multiple-output (MIMO) devices are suggested in recent studies as a way to enable AoA estimates at the anchor side [13]. However, these designs have similar high cost issues as the ToA/ TDoA strategy since they rely on expensive SDRs like universal software radio peripherals (USRPs) [14]. Received signal features (received signal strength indicator (RSSI), path loss (PL))-based localization holds significant importance over other techniques due to its simplicity of implementation [15,16], as it does not require special hardware for time synchronization or angle measurements [17], making it a practical and cost-effective solution for localization in wireless sensor networks.

Localization studies in wireless sensor networks using received signal features employ two primary approaches [17,18]: fingerprint-based direct location estimation and range-based distance estimation followed by trilateration. Fingerprint approaches in localization utilize data-driven machine-learning models that are specifically tailored to the physical coordinates of the nodes. In these methods, the target node's location coordinates and the RSSI readings from nearby gateways are measured. Ten machine learning models were compared in related research [18,19], utilizing a variety of benchmark criteria. A non-data-driven machine learning approach has been presented in [20] for WiFi indoor localization utilizing hierarchical symbolic dynamic filtering. This method employs a modified fingerprinting approach for indoor positioning in low-range environments. It has been demonstrated in [21] that when gateway density increases, fingerprint-based algorithms become more accurate. It is crucial to note that increasing the number of gateways also results in higher localization costs because of hardware expenditures, challenges with data collection, and increased computational needs for processing the increased data [22]. On the other hand, range-based techniques estimate distances by analyzing received signal information using machine learning or path loss models [18,23]. Time-weighted distance mapping based on path loss models, machine learning, and deep learning methods was assessed in a recent study [24]. We focused on range-based distance estimation followed by trilateration, considering the trade-off between localization accuracy, deployment cost, and computational performance [18,25].

Other work [25] focused on leveraging multiple received signal features to enhance localization accuracy. In general, localization models are trained using offline data in the fingerprinting and range-based techniques, and then, in the online phase, they estimate the unknown position by comparing the observed continuous time series values with stored values. However, both approaches suffer from limitations, such as the requirement of an offline phase that is time and labor-intensive and vulnerable to environmental dynamics. Such offline training faces several issues, such as the costly and time-consuming collection of training data, inherent data-driven bias, and the requirement for high-performance computing resources. Any changes in the operating environments, e.g., an addition of new structures or significant vegetation, necessitate the re-collection of training data and the retraining of models, making these approaches less adaptable to dynamic settings. It is essential to note that fingerprints collected at one point may not remain valid in future instances. As a result, the periodic updating of the fingerprint database becomes indispensable [26], leading to significant time and storage consumption.

This study presents a novel approach to outdoor localization. Our proposed approach specifically focuses on addressing the localization and tracking of target nodes within a defined region. This approach utilizes unsupervised symbolization techniques, specifically for LoRa received signal features, without requiring prior knowledge of the number of unique stationary characteristics or classes present. It simultaneously learns and estimates the distance range without the need for a separate offline training phase and the storage of reference coordinates. It leverages the maximum entropy partitioning (MEP), the D-Markov machine, and the Chinese restaurant process (CRP) concept to achieve this. Symbolization of LoRa received signal features involves converting time-series data into symbol strings through coarse-graining, followed by encoding nondeterministic state machines to capture statistical characteristics and patterns. Significantly, our method offers the advantage of being applicable to continuous time series data without the need for an offline training phase, effectively reducing the impact of environmental dynamics and operational costs. It seamlessly combines learning and estimation into a single process. We conducted experiments using RSSI and path loss data from two distinct setups employing a low-cost LoRaWAN architecture to validate our algorithm. The datasets were gathered using three gateways (RAK2245 with Raspberry Pi 3 Model B+) and one target node (Pycom LoPy4), with the ChirpStack server. The LoRa-based outdoor localization architecture presented in this study offers the advantages of enhancing accuracy, reducing complexity and hardware requirements, making it potentially suitable for monitoring livestock in rangeland areas where other networks may be unavailable or prohibitively expensive.

The major contributions of this paper are:

- A novel method for localization utilizing unsupervised symbolization on LoRa received signal features, incorporating MEP, the D-Markov machine, and the Chinese Restaurant Process.
- Collection and preparation of a LoRaWAN dataset targeting GNSS-independent low-cost LoRa localization and tracking using RAK2245 Pi LoRaWAN gateways, Pycom LoPy end-devices, and ChirpStack server.
- Two different case studies in an outdoor environment focusing on localization and tracking for two received signal features (RSSI and PL).

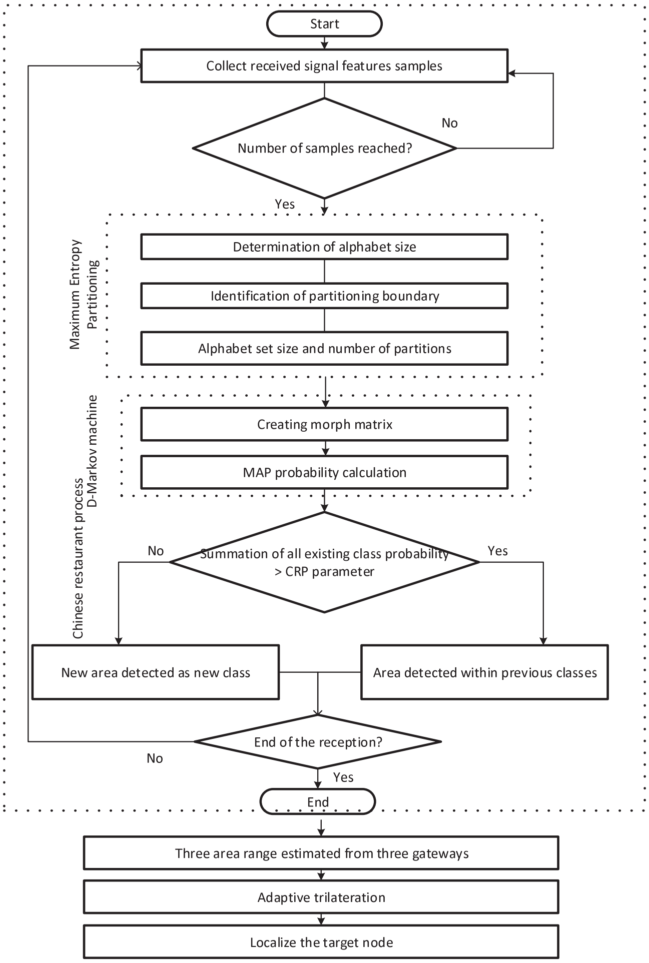
Start
Collect received signal features samples
Number of samples reached?
No
Yes
Maximum Entropy Partitioning
Determination of alphabet size
Identification of partitioning boundary
Alphabet set size and number of partitions
D-Markov machine
Creating morph matrix
MAP probability calculation
Chinese restaurant process
Summation of all existing class probability > CRP parameter
No
Yes
New area detected as new class
Area detected within previous classes
End of the reception?
No
Yes
End
Three area range estimated from three gateways
Adaptive trilateration
Localize the target node

Fig. 1. Flow chart for our proposed model.

2. Methodology

This paper addresses a range-land localization and tracking problem using unsupervised symbolization with LoRa-based received signal time series data, employing information-theoretic Markov modeling as the underlying framework. Our methodology comprises Maximum Entropy Partitioning (MEP), D-Markov machine, Chinese Restaurant Process, and adaptive trilateration. In Section 2.1, we introduce our theoretical framework, discussing the concept of unsupervised symbolization and its significance in existing literature. We utilize MEP algorithm for efficient data partitioning, as discussed in Section 2.2. We use the D-Markov machine to construct the required nondeterministic state automata, detailed in Section 2.3. The process involves creating a probability function based on a multi-nominal distribution to establish state transition mapping and morph matrix for the D-Markov machine, described in Section 2.4. Furthermore, we enhance the approach for real-time tracking by incorporating a CRP distribution into the multi-nominal distribution, as explained in Section 2.5. The proposed method utilizes an adaptive trilateration approach, which is explained in Section 2.6. Fig. 1 illustrates a flowchart of our proposed model.

2.1. Theoretical framework

Symbolic dynamics is a mathematical approach that converts the continuous trajectory of a dynamic system into a discrete sequence of symbols for analysis. Symbolic dynamics-based analyses have been used in many application areas including, communication encoding [27], information extraction [28], symbolic time series analysis [20,29], Markov decision processes [30], anomaly detection [31], and pattern recognition [32]. Unsupervised symbolization involves converting time series data into a string of (spatially discrete) symbols from which the embedded dynamic (statistical pattern) information can be extracted in an unsupervised manner (i.e., no requirement for labeling of time series). Extraction of statistical features from time-series data generated from a dynamic system (e.g., in our case, received signal features collected using gateways) can be posed as a two-time-scale problem. In a dynamic system, the term fast scale refers to the time scales at which the statistical characteristics of the system's dynamics are presumptively constant. In contrast, the slow scale involves time scales where these characteristics may progressively change. By generating a sample of a fast scale from several consecutive RSSI/path loss readings in a short time, we can extract a statistical pattern. Observing several fast time scale samples enables us to create a slow time scale and discern whether the statistical pattern remains constant or undergoes changes. In the case of changes, the new sample represents a new state; otherwise, it belongs to the previous state. Extracting information from sensor signal outputs, such as time series data, involves several key steps:

- Coarse-graining: By segmenting the data and giving each segment a different symbol, this process entails turning scalar or vector-valued data into symbol strings. These symbols are from a limited alphabet.
- Encoding probabilistic state machines: Probabilistic state machines are built using the symbol strings obtained in the preceding stage. These devices record the statistical characteristics and patterns seen in the series of symbols.

Information loss may occur when continuous time series are converted into discrete symbol series. But researchers have addressed this problem and shown that symbolization may improve the signal-to-noise ratio in noisy communications [33]. Additionally, working with symbolized data has benefits over working with continuous-valued data in terms of efficiency and efficacy in digital communication and numerical computing [31,34]. Creating appropriate symbol blocks (words) for information representation and defining symbols are the two key responsibilities of time series symbolization [29,35]. One typical method is to divide the time-series data range into exhaustive and mutually exclusive parts, giving each segment a different symbol. The partitioning scheme, such as maximum entropy partitioning, is based on equal frequency of symbol occurrence. The size of the alphabet, or the amount of symbols employed, might change depending on the application and the properties of the data. The process of building symbol blocks (words) that reflect significant temporal patterns begins once symbols have been specified. In order to create dynamic models from the symbol series that can be applied to event prediction, this step is essential. Word combinations can hold all of the dynamical information, similar to how time-delay embedding works in phase space. A context tree in a variable-order Markov model [36], in which the symbols in a series display Markov property and rely on earlier symbol blocks (words), can be used to depict the outcomes of word formation.

2.2. Algorithm for Maximum Entropy Partitioning of time series data

Maximum Entropy Partitioning (MEP) is an algorithm used to identify partitioning boundary locations in a time series signal. The objective is to obtain a symbol sequence that retains the temporal transition behavior embedded in the signal. LoRa received signal features, including RSSI, SNR, and path loss, do not exhibit a straightforward correlation with distances. To address this challenge, we determine partitioning segment locations. Linear or uniform partitioning, where they have equal-sized segments, is not suitable as variations in the received signal caused by interference or noise may not be evenly distributed. The key idea is to uniformly distribute the occurrence probability of each symbol in the generated symbol sequence such that the information-rich regions of the time series are partitioned finer and those with sparse information are partitioned coarser. From the dataset analysis, we can understand that the concentration of RSSI or path loss values is not same at different ranges. So uniform partitioning is not suitable. MEP can be one of the best choices for the LoRa-based RSSI/path loss dataset partitioning to differentiate it at different distances.

The algorithm for Maximum Entropy Partitioning is as follows:

Algorithm 1 Maximum Entropy Partitioning

```
1: Initialization: Gather the time series data for path loss/RSSI denoted as $\mathcal{T}$ and determine the alphabet size, represented by $\mathcal{A}$.
2: Define the level of RSSI/path loss values.
3: Set the level of RSSI/path loss values.
4: Determine the number of partitions required.
5: Calculate the interval value by dividing the difference between the minimum levels by the number of partitions.
6: Arrange the values in $\mathcal{T}$ in ascending order.
7: Assign a variable to represent N, which corresponds to the length of $\mathcal{T}$.
8: Create a list to store partition values and add the minimum level value to it.
9: Begin a loop from 2, up to $\mathcal{A}$ (inclusive) using a variable.
10: Calculate $\lceil ((i-1)\cdot K)/\text{partition number}\rceil$ to include in the list of partition values.
11: Once complete, add the highest level value to conclude by adding all partition values.
```

2.3. Creating a D-Markov machine

This section provides details of the construction of D-Markov machines. D-Markov machine is a stochastic process that extends the concept of Markov chains to incorporate additional dimensions or variables. The application of Markov models has been explored in various literature domains, such as a fault diagnosis framework for rotating machinery [37], fault detection in a gas turbine engine [38], as well as feature extraction from time series data [20,39,40]. It is typically defined by a tuple $(\mathcal{S}, \mathcal{A}, \gamma, \mathcal{M})$.

- A finite or countable set of alternative states in which the system can exist is represented by the symbol $\mathcal{S}$. $\mathcal{S}$ is a non-empty finite set with cardinality $\mathcal{S} < \infty$. The states, which might be discrete or continuous variables, represent various system setups or circumstances.

- $\mathcal{A}$ represents the alphabet, a finite set of possible values for the additional dimensions or variables. $\mathcal{A}$ is a non-empty finite set with cardinality $\mathcal{A} < \infty$.
- γ is the state transition mapping, which specifies the probabilities of transitioning from one state to another, $\gamma : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$.
- $\mathcal{M}$ is the morph matrix, which represents the probabilities of transitioning from one state to another with the size of $\mathcal{S} \times \mathcal{A}$. $\mathcal{M} : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$ that satisfies the condition $\sum_{n=1}^{|\mathcal{A}|} \mathcal{M}_{xn} = 1$, $\forall s_x \in \mathcal{S}$ and $\mathcal{M} \triangleq \mathcal{M}(s_x, a_n)$ is the probability of emission of the symbol $a_x \in \mathcal{A}$ when the state $s_n \in \mathcal{S}$ is observed.

To correctly estimate the matrix $\mathcal{M}, \mathcal{M}(s_x, a_n)$, a sizable number of samples are required [39]. However, the amount of samples from the dataset is limited. As a result, it is necessary to consider the probabilistic implementation of A during the estimate phase, considering the probability distribution that symbolizes the continuous time series data sample.

2.4. Maximum a posteriori (MAP) estimation from multinomial distribution

The multinomial distribution is a frequently used probability distribution for modeling categorical data with numerous categories. By combining previous knowledge or assumptions about the parameters, the maximum a posteriori (MAP) estimate is a Bayesian method for estimating the parameters of a multinomial distribution. The posterior probability is maximized by multiplying the likelihood by the prior probability, which is the goal of MAP estimation.

In variational Bayesian methods, the posterior probability is the probability of the parameters

$$\mu(C_i|\tilde{A}) = \frac{\mu(\tilde{A}|A^i)\mu(C_i)}{\sum_{j=1}^{k} \mu(\tilde{A}|A^i)\mu(C_i)}, \quad \text{for } i = 1, \dots, k \tag{1}$$

where $\mu(\tilde{A}|C_i)$ is the likelihood function representing the probability of observing the data $\tilde{A}$ given the parameter C_i, $\mu(C_i)$ is the prior distribution representing the prior knowledge or beliefs about C^i, and $\mu(\tilde{A})$ is the marginal likelihood representing the probability of observing the data A. In other words, $\mu(C_i)$ is the known prior distribution of the class C_i. Then the classification decision can be made as follows [38]:

$\mu(\tilde{A}|A^i)$ has computed by the following equation,

$$\mu(\tilde{A}|A^i) = \prod_{x=1}^{|S|} \frac{(\tilde{N}_x)!(N_x^i + |\mathcal{A}| - 1)!}{(\tilde{N}_x + N_x^i + |\mathcal{A}| - 1)!} \prod_{n=1}^{|\mathcal{A}|} \frac{(\tilde{N}_{xn} + N_{xn}^i)!}{(\tilde{N}_{xn})!(N_{xn}^i)!} \tag{2}$$

$\tilde{N}_{xn}$ denotes the count of occurrences where the nth symbol from the alphabet set appears immediately after the xth state within the symbol string obtained from the test time series dataset. Additionally, $\tilde{N}_x$ be the sum of $\tilde{N}_{xn}$ for all n from 1 to the size of the alphabet set $\mathcal{A}$.

Similarly, N_{xn}^i represents the count of occurrences where the nth symbol from the alphabet set appears immediately after the xth state within the symbol string under each of the class or areas. Correspondingly, N_x^i is calculated as the sum of N_{xn}^i for all n ranging from 1 to the cardinality of the alphabet set $\mathcal{A}$.

2.5. Chinese Restaurant Process (CRP)

The CRP is a widely used Bayesian nonparametric prior, commonly employed in statistical modeling [41,42]. The CRP, which is based on De Finetti's theorem, is a distribution over partitions or clusters. We choose the CRP as it fulfills this requirement by enabling online classification with an unbounded number of data classes. A hypothetical Chinese restaurant with an unlimited number of tables serves as an illustration of CRP. A probability distribution that allocates a table to a new client (the k + 1th customer) in discrete time is the stochastic process in this situation. The CRP is defined by the following equations:

$$\mu(z_i = k|z_1, \dots, z_{i-1}) = \begin{cases} \frac{n_k}{i-1+\alpha} & \text{if } k \text{ is an existing table} \\ \frac{\alpha}{i-1+\alpha} & \text{if } k \text{ is a new table} \end{cases} \tag{3}$$

where z_i represents the table assignment for the ith customer, n_k is the number of customers sitting at table k excluding customer i known as strength or concentration function, i is the index of the current customer, and α is the concentration parameter controlling the level of clustering in the CRP.

The initial data sample is allocated to table/area number one. For subsequent samples, there are two alternatives for categorization: either the sample is added to an existing table or area already formed, or a new table has to be made with the sample as its initial member. The likelihood of the first scenario is expressed as $\frac{n_k}{i-1+\alpha}$, where n_k denotes the concentration function associated with the amount of data grouped for the ith class. The second case has a probability of $\frac{\alpha}{i-1+\alpha}$, where α is the concentration/CRP parameter. In this formulation, $\mu(\tilde{A}|A^i)$ in Eq. (2) can be used as the strength or concentration function n_k, which provides a measure of the sample's association with each existing table.

For real-time tracking, decision classes (probable localization areas) are not fixed. Here comes the need for probability distribution where the new class is always welcomed. Chinese restaurant process is one of that type of distribution. The CRP is applied in non-parametric modeling.

Algorithm 2 Trilateration Algorithm with Radius Adjustment

```
function TRILATERATION($circles$ and $partitioning\_density$ as input)
    while the circles are not $intersecting$ do Make them intersects
        for every two circles do
            calculate the first circle center, radius, and partitioning_density
            calculate the second circle center, radius, and partitioning_density
            calculate the distance between two centers $\sqrt{dx^2 + dy^2}$
        if $distance \leq radius1 + radius2$ then
            continue
        Not intersecting
        if $partitioning\_density1 < partitioning\_density2$ then
            new_radius is calculated from $radius1 + partitioning\_density1 \times distance$
            $circles[i].radius \leftarrow new\_radius$
        else if $partitioning\_density2 < partitioning\_density1$ then
            new_radius is calculated from $radius2 + partitioning\_density2 \times distance$
            $circles[j].radius \leftarrow new\_radius$
    return $circles$
```

2.6. Adaptive trilateration

Due to the incorrectly anticipated radius values, the trilateration technique with radius adjustment aims to estimate the points' locations based on the intersection of the circles. By repeatedly changing the circle radii for a collection of circles with known centers and radii, the approach tries to pinpoint exactly where these points are. The procedure begins by supposing that the circles do not overlap. The total of their radii is then compared with the distance between each pair of circle centers to check if any of the circles overlap. This is done after determining the distances between each pair of circle centers. If circles are found to intersect, the algorithm advances to the following iteration. The method chooses the circle with the lower partitioning density and enlarges it if circles do not intersect. The required distance between circles is represented by the partitioning density, allowing for more accurate placement of points. By expanding the circle's radius with lesser density, the algorithm makes sure that the circles maintain the locations where intersections are most likely to occur. The procedure is finished and the requisite positioning precision is reached when the circles finally join. The method ends here and outputs the points' final positions based on the modified circle radii. The required partitioning density can be maintained while the positioning precision is fine-tuned by iteratively changing the radii of non-intersecting circles.

3. Experimental setup

Our experimental setup comprises several steps, starting with data collection and preparation, elaborated in Sections 3.1 and 3.2, to ensure the availability of relevant datasets. A short description of the LoRa received signal time series data and the performance evaluation parameters are given in Sections 3.3 and 3.4, respectively.

3.1. Data collection

We collected a LoRaWAN dataset with 5,553 rows of messages, utilizing the AU915 frequency band. Each transmission comprises 37 columns collected from three gateways (RAK2245 with Raspberry Pi 3 Model B+) and two target nodes (Pycom LoPy4), connected to the ChirpStack server. No specialized gateways with specific localization characteristics are used in our data-collecting arrangement. Our system comprises of three open-source 8-channel LoRa gateway (RAK2245 Pi HAT LoRaWAN module 8 Channels with Raspberry Pi 3 Model B+) and one LoRa target node (Pycom LoPy4) with ChirpStack LoRaWAN network server. When the target nodes sent signals to the gateways, the ChirpStack server captured and stored the information. An iPhone 13 Pro was used to indicate the geodesic coordinates of the target nodes and gateways. The distance is computed according to geodesic distance.[1] The distance maps are acceptable for the trilateration approach, and the gateways were arranged in a triangle such that each gateway experiences distinct propagation circumstances to the destination node. All transmission variables are logged by the ChirpStack server, which is linked to the gateways and target nodes. Fig. 2 shows the setup for data collection arrangement. For high precision and consistency, the gateways are stationed at their locations, serving as fixed points of reference for the subsequent uses of the trilateration step. The parameters were selected based on the testbed region and experimental analysis. The location chosen for the data collection was the Sports Oval at Murdoch University ($32°04'21.3''S$ $115°49'40.0''E$) (Fig. 3). The site is semi-flat with full LoS (Line-of-Sight) and an approximate area of 30 000 m^2. Tables 1 and 2 show the positions of all three gateways and their Euclidean distances.

[1] https://geopy.readthedocs.io/en/stable/#module-geopy.distance

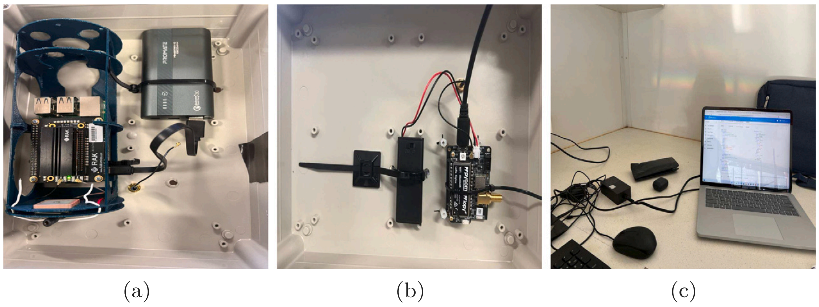

Fig. 2. Hardware and server setup: (a) Gateway (RAK2245 with Raspberry Pi 3 model B+) (b) Target node (Pycom LoPy4) (c) ChirpStack server.

Table 1
Positions of the gateways.

Gateway	Latitude	Longitude
Gateway 1	32°04′19.6″S	115°49′35.9″E
Gateway 2	32°04′26.2″S	115°49′41.2″E
Gateway 3	32°04′20.7″S	115°49′45.7″E

Table 2
Euclidean distance between the gateways.

From	To	Distance
Gateway 1	Gateway 2	237 m
Gateway 2	Gateway 3	233 m
Gateway 3	Gateway 1	220 m

3.2. Dataset preparation

We explain the experimental design used to gather the dataset for two different case studies in this section. The dataset was initially stored in JSON format and had 5,553 rows of messages in 37 columns. The JSON files were converted to CSV format to speed up processing. Gateway ID, time, RSSI, LoRaSNR, channel, latitude, longitude, altitude, frequency, bandwidth, spreading factor (SF), coding rate, device address (devAddr) for target nodes, frame count (fcnt), adaptive data rate (adr), and payload were significant columns in the dataset analysis. The obtained data can be divided into specific gateways and target nodes using the gateway ID and devAddr, respectively. The coordinates of the gateways were represented by the latitude and longitude numbers, which were afterwards confirmed with a dual-frequency global positioning system (GPS) on an iPhone. A count of each received data was supplied via the *fcnt* column, allowing for the estimation of data loss. Data analysis showed that 98.7% of the sent packets were successfully received by the gateways. The experiment's frequency range was 915–928 MHz, which was selected based on the testbed's location and the intended application area in Australia's rangeland region. In accordance with [43], the transmission parameters like transmission power, carrier frequency, spreading factor, bandwidth, and coding rate can have a considerable impact on LoRa performance. In our example, the bandwidth was set to 125 kHz, and the coding rate for both target nodes was set to 4/5. We concentrated on the SF7 values in our dataset analysis since the pattern of the gathered RSSI values could be clearly distinguished across the various SF levels [25].

3.2.1. Case study 1: on full site

The first case study focuses on a full-site scenario and involves distance area estimation followed by trilateration. In our experimental setup, we used three gateways because a single gateway offers an estimated circular area, two gateways allow for estimating a linear area, and three gateways are required for trilateration to determine a specific estimated point or small region. It should be noted that more than three gateways within an area already covered by three gateways will be redundant, increasing cost and complexity. In this setup, 15 areas were selected under each gateway, and the borders of each region were increased by 20 m. The location of each gateway is shown in the center of the circles in Fig. 3, which shows the geographic details of these areas. The three gateway locations were carefully chosen to create intersecting regions that covered the whole sports oval, ensuring thorough coverage. The earth map from the *folium* library collection was used to create the illustration. 28 tuples of inputs were gathered for a total of 180 locations and were used in the dataset. For the purpose of evaluating the effects of time-weighting on signal strength indicators, three subsets of the main dataset were generated, each of which contained 10 tuples of RSSI/path loss values. These subgroups made it possible to assess how time-weighting affected the study's signal strength indicators.

Positively skewed dataset: The dataset considers that earlier received signals at a waypoint are more significant in order to determine distances from transmitting gateways. This weighting considers arrival time into account and implicitly represents a feature space with three degrees of freedom.

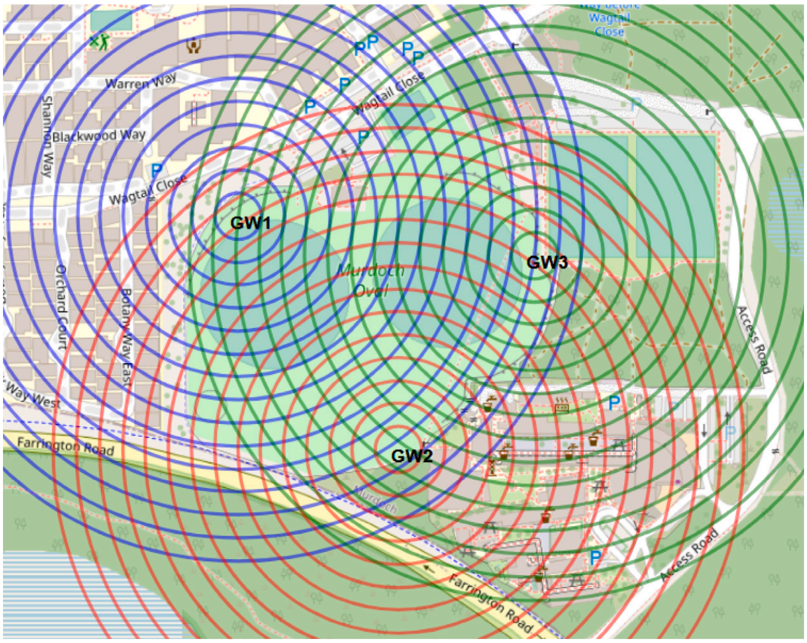

Fig. 3. Map of the experimental site with three gateway locations and 20-m incremental radius boundaries (Blue, red, and green zones are under GW1, GW2, and GW3 respectively).

Negatively skewed dataset: This mimics the idea that the most recent signals that are received at a waypoint have a greater ability to predict distances from transmitting gateways.

Middle-range dataset: This dataset uses the median values of the signals that were received at a waypoint since it is assumed that the most recent or oldest signals may be inaccurate as a result of being transferred from one waypoint to another.

3.2.2. *Case study 2: on a designated area*

The second case study focuses on three designated areas. The study assumes that the bearer, e.g. livestock, of the node can move freely within a defined area. Still, they also exhibit certain instincts that lead them to specific regions, such as ground-water tables, suitable grazing fields, or living sheds. This study aims to determine the probable areas where livestock may be located. By identifying these areas, farmers can better manage and address various aspects related to livestock management. Consequently, we carefully selected three distinct areas within the target region based on their varying received signal qualities. Our analysis specifically focuses on the RSSI values associated with these areas. By scrutinizing these signal characteristics from the primary dataset, we aim to gain insights into the unique signal properties exhibited by each designated area. The areas are shown in Fig. 4.

Area 1 was picked because it is situated close to the boundary of one gateway and generally equidistant from two other gateways, according to the values of the RSSI. This configuration enables a balanced coverage area from two gateways and describes how it functions if a third gateway is out of reach or has signal degradation.

Area 2 is deliberately positioned in contrast, almost equidistant from the three gateways. This location is essential for consistent coverage and the best signal dispersion over the whole region. Area 2 can act as a hub where devices can connect to the closest gateway, minimizing signal attenuation and guaranteeing effective data transfer by choosing a site roughly an equal distance from the gateways.

Area 3, on the other hand, is deliberately placed extremely near one particular entryway. This positioning focuses on offering focused coverage and improved signal strength in a particular localized area, highlighting a new kind of quality in area selection. Devices in Area 3 can make use of the strong signal and low latency provided by the neighboring gateway by placing Area 3 adjacent to one.

One of the symbolization concepts [44] is symbolic time series filtering (STSF) [29,45], a theory that focuses on the discretization of dynamical systems in both space and time. A pattern recognition tool that uses the idea of STSF and represents a time series of sensor data as a symbol sequence to build a Markov model [46–51].

3.3. *Time series data*

3.3.1. *RSSI*

The RSSI, which measures the strength of a signal received from a sender, may be used to calculate the distance between the transmitter and the receiver. The normal highest RSSI value is −30 dBm, which represents the strongest signal. The typical lowest

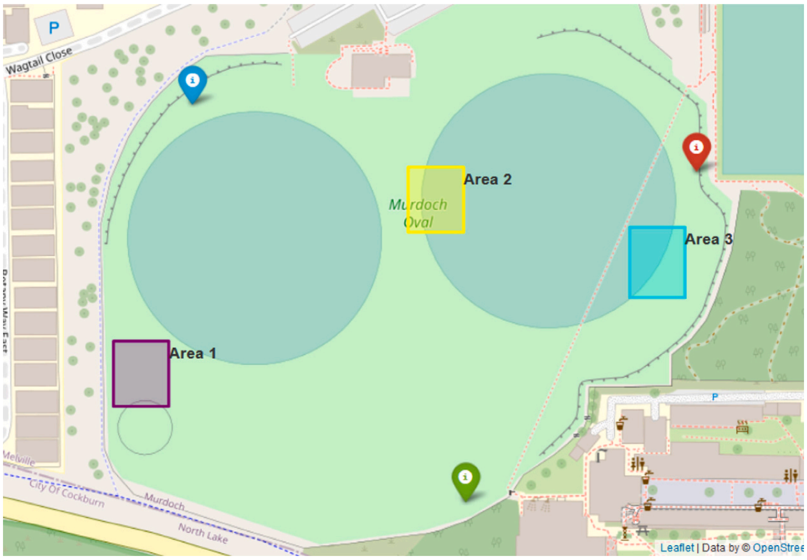

Fig. 4. Map of the experimental site, showcasing three distinct and designated areas, each possessing unique signal properties.

RSSI value for LoRa is −120 dBm, which denotes the weakest signal. The spreading factor (SF) employed in LoRa modulation affects the value of RSSI. Therefore, we carefully filtered the data for SF7 during the data-gathering process.

3.3.2. Path loss

The loss or attenuation an electromagnetic signal experiences as it travels from the transmitter to the receiver is referred to as path loss. Antenna gain depends upon its physical size compared to wavelength [52]. A half-wavelength LoRa-915 MHz Antenna Kit was used as a transmit target node antenna, which is expected to have around 2dBi antenna gain. As a receiver end gateway, the RAK2245 was used with a LoRa iPEX 2dBi antenna. The typical transmit power for the target node is 20dBm.

For each message of the dataset, the path loss (PL) was calculated using Eq. (4) [53], which takes into account the SNR and RSSI values from the corresponding message, as well as the antenna gains and transmit power from the hardware used.

$$PL_{\text{measured}} = P_T + G_T + G_R + 10 \times \log_{10}\left(1 + \frac{1}{\text{SNR}}\right) - \text{RSSI} \tag{4}$$

Here, P_T is transmit power, and G_T and G_R are the transmit and receive antenna gains. Next, the path loss at the reference distance of 1 m (d_0), denoted as $PL(d_0)$, is calculated using Eq. (4).

3.4. Performance evaluation parameters

We use mean average error for distance mapping and localization estimation to evaluate the performance. Here, we provide mathematical equations used to assess localization accuracy.

We used the Haversine formula to measure the geodesic distance between two locations on a curved surface like Earth.

$$d = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{5}$$

Here, the geodesic distance (or any other unit of measurement) between the two sites is d in kilometers. The radius of the sphere or ellipsoid approximately representing the curved surface is r. The latitudes of the two points are ϕ_1 and ϕ_2 in radians, respectively. The two points' radian longitudes are λ_1 and λ_2.

4. Result analysis

To evaluate the proposed method's real-time localization capability, we selected a sports oval at Murdoch University with an estimated coverage area of 30,000 square meters. Our experimental setup consisted of three RAK2245 Pi HAT modules with Raspberry Pi 3 Model B+ form factor SX1301, which served as the complete RF front-end for the LoRa gateway. We deployed a single Pycom LoPy4 device as the target node, positioned at a height of approximately 0.5 m, equivalent to the average height of a goat or a sheep. For cloud-based access from any location, we set up a ChirpStack server as the network and application server, with each ChirpStack LoRa gateway configured on Raspberry Pi to connect with the server. The selection of parameters for our setup was based on an analysis of the testbed region and experimental observations.

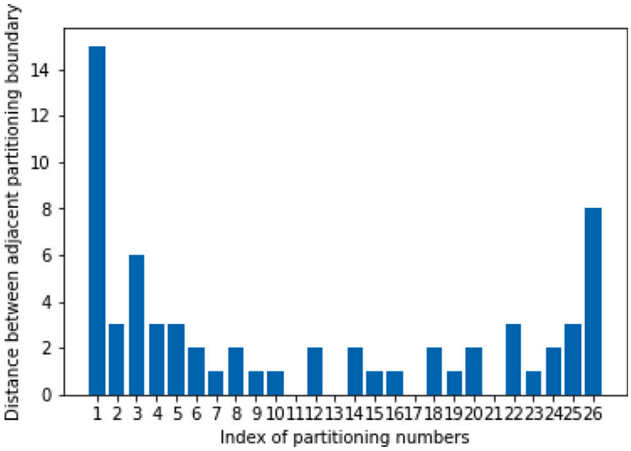

Fig. 5. Maximum Entropy Partitioning on the dataset (density of the partitions).

4.1. Case study 1

Our operational area is an LoS rangeland area to evaluate the proposed method for target node tracking. The target node moves/covers the whole area. The area covered by each gateway, distance-wise, it has divided into 15 ranges (each range of 20 m) according to distance. Each gateway test environment creates 15 areas, and the target LoRa node moves between these 15 areas 180 times. The objective is to identify the transition between the fifteen locations at each gateway. The target node returns to each location and the algorithm should identify the node's position at that specific time. The geodesic coordinates of the target nodes and gateways were marked using an iPhone 13 Pro. The distance is computed according to geodesic distance.

The test environment depicted in Fig. 3 consists of an outdoor layout where the network connectivity is challenging due to its remote rangeland characteristics. The layout includes selected positions and the placement of three LoRa gateway points. This environment was chosen to evaluate the robustness of the proposed method against environmental factors like multipath, considering the low connectivity prevalent in LoS locations. A Pycom LoPy4 device was utilized to evaluate a target node's movement. Positioned at a height of approximately 0.5 m, which corresponds to the average height of a goat or a sheep, the device was moved across the entire area. The objective was to estimate the device's location, treating it as a moving target. Data collection was carried out using the Chirpstack LoRa server. The experiments were conducted on a desktop computer with the following specifications: an Intel(R) Core(TM) i7-6800K processor, 64 GB of RAM, and an NVIDIA GeForce GTX 1080 Graphics Processing Unit (GPU). The operating system employed was the Professional Edition of Windows 10. For coordinate retrieval and geographical distance calculations, the GeoPy library was utilized.

We used a dataset where the RSSI range spanned from −48dBm to −113dBm, and the SNR range spanned from −9.75 dB to 10 dB. For the transmit target node antenna, a LoRa-915 MHz Antenna Kit with a half-wavelength design was utilized. This antenna kit is expected to provide an antenna gain of approximately 2dBi.[2] As for the receiver end gateway, a LoRa iPEX 2dBi antenna connected to RAK2245 was employed.[3] Regarding transmit power, the target node operated at the default level 20dBm. To divide the RSSI and path loss range into partitions, we selected 26 as the number of partitions. The partitioning process was carried out based on the MEP approach outlined in Algorithm 1. Fig. 5 illustrates the calculated partitions. The partitioning results indicate a higher frequency of occurrence in the middle range of RSSI/path loss values compared to the lower and significantly higher ranges. Consequently, estimating the localization class for middle-ranged RSSI/path loss values entails a higher error probability. Leveraging this characteristic, we update the typical trilateration algorithm. Each partition was assigned a distinct symbol from the English alphabet, which consisted of 26 letters. MEP enabled us to represent the 26 partitions conveniently. For simplicity, we set the depth of the D-Markov machine to 1, ensuring that the size of the alphabet matched the number of elements in the state set. After calculating the morph matrix $\mathcal{M}$, it becomes possible to determine the probability of each sample data being assigned to a specific area using Eqs. (2) and (3). This probability calculation aids in identifying the most likely area for each sample data point based on the calculated Morph matrix. To mitigate computational complexity, the probabilities were computed on a logarithmic scale. The maximum probability of a sample belonging to any of the areas is then identified as the resulting localization output, representing the estimated range of the sample's assigned area. The target node traversed 15 areas/classes associated with each gateway, with a CRP parameter set to 400 for the calculation. When estimating a specific area class, the mean point was used as the circle's radius around the gateway, enabling subsequent trilateration calculations.

[2] https://pycom.io/product/lora-868mhz-915mhz-sigfox-antenna-kit/

[3] https://docs.rakwireless.com/Product-Categories/WisLink/RAK2245-Pi-HAT/Datasheet/

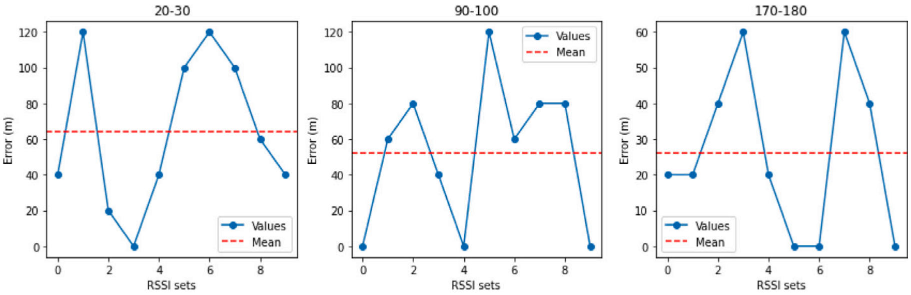

Fig. 6. Mean average error in ranging accuracy at a different stage of continuous learning from positively skewed RSSI dataset.

Table 3
Comparison in mean average ranging accuracy for RSSI and path loss data on different dataset orientations.

RSSI			PL		
Positive	Negative	Middle	Positive	Negative	Middle
28	38	30	34	34	36

In this method, each time series value is assigned a symbol according to the splitting range done by MEP. Consequently, this time series values to symbols mapping disregards the time series data's uneven fluctuations or noise. Each row of the matrix includes transition probabilities from a specific state to other available states. Statically, both columns and rows of the morph matrix are independent according to Markovian property. Therefore, there will be no cumulative effects when constructing the morph matrix. This approach primarily captures the statistical pattern changes in the time series data via symbol sequences. Then, the model parameters for the classification decision encompass the count of occurrences of each symbol from the alphabet set alongside each state, both on an individual state basis and cumulatively across all states within the symbol string derived from the test sample. These model parameters are updated in a database under each class during the localization estimation process. Moreover, the database is updated with all model parameters when a new class is generated based on CRP calculations. Thus, the RSSI/PL data collected over time is considered learning data for testing the present and future samples. Consequently, the total error rate will decrease over time.

We worked on six sets of data: positive, negative, and middle-skewed data for both RSSI and PL. To exhibit consistent improvement in accuracy, we exclusively present the positively skewed RSSI dataset. This self-correcting behavior can be observed visually in the results depicted in Fig. 6, demonstrating continuous improvement through continuous learning. The results showcase gradual improvement over time, with initial samples (20–30) having a mean error of 64 m, which reduces to around 26 m for mature-stage samples (170–180). However, the mean average ranging accuracy results from all six sets of data have accumulated in Table 3. Table 3 presents the mean average error rates for different dataset orientations in both RSSI and path loss. The results show the mean error obtained from the last ten samples output. Each area is defined by boundaries with a 20-m incremental radius, meaning that a one-step mistake in estimation corresponds to a 20-m error. Initially, our goal is to estimate the target node's possible localization inside the boundaries of each gateway within a certain region range. The limits of each region are 20-m increments starting from the centers of the circles, which stand in for the locations of the gateways. We refer to the precision of estimating the correct range as 'ranging accuracy'. The impact of time-weighting is examined using three distinct datasets, positive, negative, and middle-skewed. The experimental results do not provide sufficient evidence to reject the null hypothesis that time weighting of the feature space yields equivalent expected mean average errors. Furthermore, the correlation coefficient between the average values of each position in the dataset is calculated to be 0.347 for path loss and 0.350 for RSSI. Since the variability in the RSSI and path loss values concerning distance is similar in the dataset, the proposed model performs comparably on both features, yielding similar results. Fig. 7 displays the actual coordinates of the 10 test points along with the measured coordinates obtained through distance estimation followed by trilateration based on partitioning range accuracy. The blue round circles represent the actual coordinates, while the green round circles represent the measured coordinates. The computation for obtaining symbols from the continuous time series is O(T), with T as the number of data points. The observable Markov model estimation with fixed structure efficiently manages computational complexity. Likelihood computation models are the most time-consuming step, but Stirling's approximation prevents exponential growth with T. Probabilities were computed on a logarithmic scale, and D=1 is used for reduced computational complexity.

Range-based distance estimation findings were employed as input parameters for trilateration, a method for estimating a target node's location [54]. To properly estimate the target's position using trilateration, three fixed, non-collinear reference points are needed. In our work, it is employed to determine the transmitter's location based on estimated distances between the device and receiving gateways. Typically, the three circles drawn from the estimated distances do not intersect at a single point due to estimation errors. If they do intersect, the solution is straightforward. However, if they do not intersect, we can equally increment the estimated radius of all three circles. This approach may have limitations, as the distances/radii might not be equally erroneous. Here, the

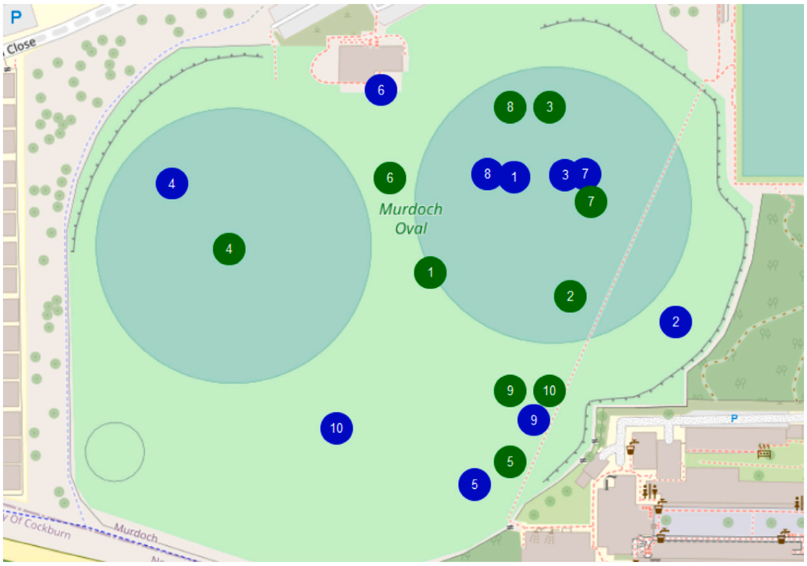

Fig. 7. Actual coordinates (blue round-circles) and measured coordinates (green round circles) from range estimation (using positively skewed RSSI dataset) followed by adaptive trilateration.

Table 4
Comparison in mean average localization accuracy with state-of-the art-approach.

Trilateration (Equal increment)	Proposed adaptive trilateration (partitioning range accuracy)	Range based (RSSI only) [24,25]	Range based (multi features) [25]	Fingerprint based [18,25]
38.52 m	32.48 m	58.25 m	42.77 m	84.63 m

importance of dataset characteristics becomes evident. The findings of the partitioning show that the intermediate range of RSSI/path loss values occurs more frequently than the lower and much higher ranges. As a result, there is a larger chance of inaccuracy when calculating the localization class for middle-ranged RSSI/path loss values. Understanding this characteristic helps identify the most error-prone areas of RSSI/path loss value estimation. Based on this hypothesis, we modify the well-known trilateration strategy by adjusting the radius depending on partitioning density. This adaptation considers the dataset's unique characteristics and aims to improve the accuracy of the localization process.

4.1.1. *Comparison with fingerprinting and range-based approach*

In this work, we also analyzed the performance of two existing localization approaches [18,24] on our collected dataset. Based on fingerprints, the first approach achieved a mean average localization accuracy of 84.63 m. The second approach involved range-based distance estimation followed by trilateration, with two variations: RSSI-only and multi-features-based distance mapping. The results are shown in Table 4. Results from the proposed approach in [25] show that the RSSI-only approach achieved a mean average localization accuracy of 58.25 m. The multi-features-based distance mapping approach achieved a significantly improved mean average localization accuracy of 42.77 m. Table 4 further illustrates the localization accuracy improvements achieved by our proposed technique. The proposed method showed a 32.48-m mean average localization accuracy when the trilateration was based on partitioning range accuracy according to Algorithm 2 and a 38.52-m mean average localization accuracy when the trilateration was based on the equal increment from the estimated radius made circles. We called it adaptive trilateration. These results indicate the superiority of the proposed model over the existing approach.

The proposed method is computationally fast. One of the investigations in this work focused on the computation time needed for location estimation based on a single LoRaWAN uplink message. This average was calculated on all six data sets' last ten test samples for both ranging and adaptive trilateration work. This method's computation time is proportional to the number of data points, exhibiting a linear [O(data points)] type of computational complexity. The minimal time demands of this method make it well-suited for utilization in real-time navigation systems.

4.2. *Case study 2*

The second case study concentrates on three designated areas, utilizing only RSSI values from the particular region. As a hypothetical approach to identifying unique cattle behaviors or patterns from a gathered dataset with coordinates, we employed the k-means clustering method, illustrated in Appendix. This approach allowed us to discover distinctive trends in our collected data

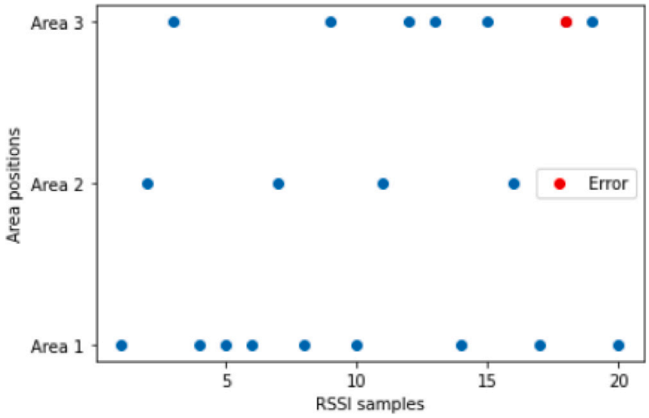

Fig. 8. Tracking demonstration between three designated areas (blue dot-tracked, red dot-error in tracking).

density based on their localization information, where we assumed cluster size was 3. In this study, we considered specific alphabets present in the RSSI values for these three regions. The D-Markov machine's depth was set to 1, and the CRP parameter was chosen as 75. In this case study, we performed range estimation followed by trilateration, focusing solely on three specific regions. If the estimated range extends beyond these designated areas, the method identifies the nearest area from that point and estimates its position accordingly. The proposed algorithm achieved a very good tracking capability of the target node with minimum error. Fig. 8 presents the results, where out of 20 positions, 19 were successfully tracked.

We employ sensor nodes powered by single-use batteries and three LoRaWAN gateways, indicating limited hardware requirements. This is an unsupervised method, so there is no need for an offline training phase and storing reference coordinates. It combines learning and estimation processes, making it easy to implement. It does not require re-collection of data or re-configuration given a new operating condition. Additionally, it avoids subscription fees for communication protocols and simplifies the process compared to traditional methods with large training data and complex infrastructure.

5. Conclusion

We propose a novel unsupervised symbolization technique for range-based outdoor localization using LoRa technology. The main contribution of the method is the simultaneous learning and estimation process. Our proposed unsupervised symbolization algorithm for outdoor localization and tracking utilizes maximum entropy partitioning to achieve finer partitioning in information-rich regions and coarser partitioning in sparse regions. The D-Markov machine constructs nondeterministic finite-state automata, and real-time tracking is facilitated by incorporating the Chinese restaurant process. We then employ an adaptive trilateration technique for localizing the target node. To demonstrate the performance of the proposed method, we collected a dataset and tested that. We used a low-power LoRaWAN network to estimate the location of a target node under the coverage of three gateways. Our proposed method demonstrates a higher ranging and localization accuracy over state-of-the-art approaches. Additionally, the proposed method showcases its tracking capability by aligning the localization results in specific designated small areas with the time frame of received signal features. Our approach for LoRa-based localization offers several advantages over traditional fingerprinting and range-based methods, eliminating the need for an offline phase, integrating learning and estimating operations, and enhancing resilience to noise and multipath effects. Outdoor testbed experiments demonstrate its effectiveness in real-time localization and tracking. The research also highlights the potential for further improvements in precision by combining LoRa with other monitoring sensors. In summary, our work contributes to the field of IoT-based object monitoring and offers insights for enhanced management in diverse agricultural settings.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix. K-means clustering

An approach used to partition a dataset into a predetermined number of clusters (denoted as 'K') is the k-means clustering algorithm. It is a flat clustering technique that calculates and adjusts centroids until the best ones are discovered. In the second case study, the K-means clustering method can be utilized to identify distinctive cattle behaviors or patterns based on their localization data. By considering the centroids of designated areas as clustering centroids and using estimated livestock positions, it becomes possible to determine the presence of target nodes within specific designated areas. The k-means clustering algorithm is described in the following way:

Algorithm 3 K-Means Clustering

```
Initialize randomly the number of clusters K and the centroids $C = \{c_1, c_2, \dots, c_K\}$.
Assign each data point ($x_i$) to the nearest centroid: $j \leftarrow \arg\min_j ||x_i - c_j||_2$
Calculate the arithmetic means of each cluster formed in the data $c \leftarrow \frac{1}{|\text{cluster}|} \sum_{x_i \in \text{cluster}} x_i$.
K-means assigns each record in the dataset to only one of the initial clusters (nearest cluster according to Euclidean distance).
Until convergence, K-means re-assigns each record and re-calculate the arithmetic mean.
```

References

[1] D. Mancuso, G. Castagnolo, M.C. Parlato, F. Valenti, S.M. Porto, Low-power networks and GIS analyses for monitoring the site use of grazing cattle, Comput. Electron. Agric. 210 (2023) 107897.

[2] K. Hu, C. Gu, J. Chen, Ltrack: A lora-based indoor tracking system for mobile robots, IEEE Trans. Veh. Technol. 71 (4) (2022) 4264–4276.

[3] S.K. Mohammed, S. Singh, R. Mizouni, H. Otrok, A deep learning framework for target localization in error-prone environment, Internet Things 22 (2023) 100713, http://dx.doi.org/10.1016/j.iot.2023.100713, URL https://www.sciencedirect.com/science/article/pii/S2542660523000367.

[4] M. Shurrab, R. Mizouni, S. Singh, H. Otrok, Reinforcement learning framework for UAV-based target localization applications, Internet Things 23 (2023) 100867, http://dx.doi.org/10.1016/j.iot.2023.100867, URL https://www.sciencedirect.com/science/article/pii/S2542660523001907.

[5] Y. Kawamoto, R. Sasazawa, B. Mao, N. Kato, Multilayer virtual cell-based resource allocation in low-power wide-area networks, IEEE Internet Things J. 6 (6) (2019) 10665–10674.

[6] B. Xie, J. Xiong, Combating interference for long range LoRa sensing, in: Proceedings of the 18th Conference on Embedded Networked Sensor Systems, 2020, pp. 69–81.

[7] F. Zhang, Z. Chang, K. Niu, J. Xiong, B. Jin, Q. Lv, D. Zhang, Exploring lora for long-range through-wall sensing, Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol. 4 (2) (2020) 1–27.

[8] K.-H. Lam, C.-C. Cheung, W.-C. Lee, RSSI-based LoRa localization systems for large-scale indoor and outdoor environments, IEEE Trans. Veh. Technol. 68 (12) (2019) 11778–11791.

[9] M. Aernouts, N. BniLam, R. Berkvens, M. Weyn, TDAoA: A combination of tdoa and AoA localization with lorawan, Internet Things 11 (2020) 100236, http://dx.doi.org/10.1016/j.iot.2020.100236, URL https://www.sciencedirect.com/science/article/pii/S254266052030069X.

[10] C. Gu, L. Jiang, R. Tan, Lora-based localization: Opportunities and challenges, 2018, arXiv preprint arXiv:1812.11481.

[11] A. Bansal, A. Gadre, V. Singh, A. Rowe, B. Iannucci, S. Kumar, Owll: Accurate lora localization using the tv whitespaces, in: Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021), 2021, pp. 148–162.

[12] Z. Shi, X. Chang, C. Yang, Z. Wu, J. Wu, An acoustic-based surveillance system for amateur drones detection and localization, IEEE Trans. Veh. Technol. 69 (3) (2020) 2731–2739.

[13] H. Huang, J. Yang, H. Huang, Y. Song, G. Gui, Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system, IEEE Trans. Veh. Technol. 67 (9) (2018) 8549–8560.

[14] C. Zhang, F. Li, J. Luo, Y. He, ILocScan: Harnessing multipath for simultaneous indoor source localization and space scanning, in: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, 2014, pp. 91–104.

[15] S.R. Jondhale, M. Sharma, R. Maheswar, R. Shubair, A. Shelke, Comparison of neural network training functions for rssi based indoor localization problem in WSN, Handb. Wirel. Sens. Netw.: Issues Chall. Curr. Scenario's (2020) 112–133.

[16] S.R. Jondhale, V. Mohan, B.B. Sharma, J. Lloret, S.V. Athawale, Support vector regression for mobile target localization in indoor environments, Sensors 22 (1) (2022) 358.

[17] S.M. Asaad, H.S. Maghdid, A comprehensive review of indoor/outdoor localization solutions in IoT era: Research challenges and future perspectives, Comput. Netw. 212 (2022) 109041, http://dx.doi.org/10.1016/j.comnet.2022.109041, URL https://www.sciencedirect.com/science/article/pii/S1389128622001918.

[18] T. Janssen, R. Berkvens, M. Weyn, Benchmarking RSS-based localization algorithms with lorawan, Internet Things 11 (2020) 100235, http://dx.doi.org/10.1016/j.iot.2020.100235, URL https://www.sciencedirect.com/science/article/pii/S2542660520300688.

[19] T. Janssen, R. Berkvens, M. Weyn, Comparing machine learning algorithms for RSS-based localization in LPWAN, in: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Springer, 2019, pp. 726–735, http://dx.doi.org/10.1007/978-3-030-33509-0_68.

[20] F.E. Oryad, H. Amindavar, Wireless positioning based on hierarchical symbolic dynamic filtering of RSSI time series, Signal Process. 206 (2023) 108903.

[21] R.K. Pallasena, M. Sharma, V. Krishnaswamy, Context-sensitive smart devices-definition and a functional taxonomy, Int. J. Soc. Humanist. Comput. 3 (2) (2019) 108–134, http://dx.doi.org/10.1504/IJSHC.2019.101593.

[22] P. Ssekidde, O. Steven Eyobu, D.S. Han, T.J. Oyana, Augmented CWT features for deep learning-based indoor localization using WiFi RSSI data, Appl. Sci. 11 (4) (2021) 1806, http://dx.doi.org/10.3390/app11041806.

[23] M. Ahmed Ouameur, M. Caza-Szoka, D. Massicotte, Machine learning enabled tools and methods for indoor localization using low power wireless network, Internet Things 12 (2020) 100300, http://dx.doi.org/10.1016/j.iot.2020.100300, URL https://www.sciencedirect.com/science/article/pii/S2542660520301323.

[24] M. Anjum, M. Abdullah Khan, S.A. Hassan, H. Jung, K. Dev, Analysis of time-weighted lora-based positioning using machine learning, Comput. Commun. 193 (2022) 266–278, http://dx.doi.org/10.1016/j.comcom.2022.07.010, URL https://www.sciencedirect.com/science/article/pii/S0140366422002572.

[25] K.Z. Islam, D. Murray, D. Diepeveen, M.G. Jones, F. Sohel, Machine learning-based LoRa localisation using multiple received signal features, IET Wirel. Sens. Syst. (2023) http://dx.doi.org/10.1049/wss2.12063.

[26] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent advances in indoor localization: A survey on theoretical approaches and applications, IEEE Commun. Surv. Tutor. 19 (2) (2016) 1327–1346.

[27] C.E. Shannon, A mathematical theory of communication, Bell Syst. Techn. J. 27 (3) (1948) 379–423.

[28] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, pp. 2–11.

[29] C.S. Daw, C.E.A. Finney, E.R. Tracy, A review of symbolic analysis of experimental data, Rev. Sci. Instrum. 74 (2) (2003) 915–930.

[30] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, 2014.

[31] A. Ray, Symbolic dynamic analysis of complex systems for anomaly detection, Signal Process. 84 (7) (2004) 1115–1130.

[32] K. Mukherjee, A. Ray, State splitting and merging in probabilistic finite state automata for signal representation and analysis, Signal Process. 104 (2014) 105–119.

[33] P. Beim Graben, Estimating and improving the signal-to-noise ratio of time series by symbolic dynamics, Phys. Rev. E 64 (5) (2001) 051104.

[34] M. Makki Alamdari, B. Samali, J. Li, Damage localization based on symbolic time series analysis, Struct. Control Health Monit. 22 (2) (2015) 374–393.

[35] S. Sarkar, P. Chattopdhyay, A. Ray, Symbolization of dynamic data-driven systems for signal representation, Signal, Image Video Process. 10 (2016) 1535–1542.

[36] J. Rissanen, A universal data compression system, IEEE Trans. Inform. Theory 29 (5) (1983) 656–664.

[37] Z. Zhu, J. Cheng, P. Wang, J. Wang, X. Kang, Y. Yang, A novel fault diagnosis framework for rotating machinery with hierarchical multiscale symbolic diversity entropy and robust twin hyperdisk-based tensor machine, Reliab. Eng. Syst. Saf. 231 (2023) 109037.

[38] S. Sarkar, K. Mukherjee, S. Sarkar, A. Ray, Symbolic dynamic analysis of transient time series for fault detection in gas turbine engines, J. Dyn. Syst. Meas. Control 135 (1) (2013) 014506.

[39] A. Akintayo, S. Sarkar, A symbolic dynamic filtering approach to unsupervised hierarchical feature extraction from time-series data, in: 2015 American Control Conference, (ACC), 2015, pp. 5824–5829, http://dx.doi.org/10.1109/ACC.2015.7172252.

[40] Y. Li, A. Ray, Unsupervised symbolization of signal time series for extraction of the year=2017, embedded information, Entropy 19 (4) http://dx.doi.org/10.3390/e19040148, URL https://www.mdpi.com/1099-4300/19/4/148.

[41] D.M. Blei, P.I. Frazier, Distance dependent Chinese restaurant processes., J. Mach. Learn. Res. 12 (8) (2011).

[42] A. Akintayo, S. Sarkar, Hierarchical symbolic dynamic filtering of streaming non-stationary time series data, Signal Process. 151 (2018) 76–88, http://dx.doi.org/10.1016/j.sigpro.2018.04.025, URL https://www.sciencedirect.com/science/article/pii/S0165168418301506.

[43] M. Bor, U. Roedig, Lora transmission parameter selection, in: 2017 13th International Conference on Distributed Computing in Sensor Systems, (DCOSS), 2017, pp. 27–34, http://dx.doi.org/10.1109/DCOSS.2017.10.

[44] D. Lind, B. Marcus, An Introduction to Symbolic Dynamics and Coding, Cambridge University Press, 2021.

[45] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, Vol. 7, Cambridge University Press, 2004.

[46] G. Pola, P. Tabuada, Symbolic models for nonlinear control systems: Alternating approximate bisimulations, SIAM J. Control Optim. 48 (2) (2009) 719–733.

[47] K. Deng, P.G. Mehta, S.P. Meyn, Optimal Kullback-Leibler aggregation via spectral theory of Markov chains, IEEE Trans. Automat. Control 56 (12) (2011) 2793–2808.

[48] P. Dupont, F. Denis, Y. Esposito, Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms, Pattern Recognit. 38 (9) (2005) 1349–1371.

[49] E. Vidal, F. Thollard, C. De La Higuera, F. Casacuberta, R.C. Carrasco, Probabilistic finite-state machines-Part II, IEEE Trans. Pattern Anal. Mach. Intell. 27 (7) (2005) 1026–1039.

[50] M. Vidyasagar, The complete realization problem for hidden Markov models: A survey and some new results, Math. Control Signals Systems 23 (1–3) (2011) 1–65.

[51] P. Adenis, Y. Wen, A. Ray, An inner product space on irreducible and synchronizable probabilistic finite state automata, Math. Control Signals Systems 23 (4) (2012) 281–310.

[52] A. Lai, K.M. Leong, T. Itoh, Infinite wavelength resonant antennas with monopolar radiation pattern based on periodic structures, IEEE Trans. Antennas Propag. 55 (3) (2007) 868–876, http://dx.doi.org/10.1109/TAP.2007.891845.

[53] G.M. Bianco, R. Giuliano, G. Marrocco, F. Mazzenga, A. Mejia-Aguilar, Lora system for search and rescue: Path-loss models and procedures in mountain scenarios, IEEE Internet Things J. 8 (3) (2021) 1985–1999, http://dx.doi.org/10.1109/JIOT.2020.3017044.

[54] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, IEEE Signal Process. Mag. 22 (4) (2005) 54–69, http://dx.doi.org/10.1109/MSP.2005.1458287.