Constrained Machine Learning for LoRa Gateway Location Optimisation

Abdullahi Abubakar Kutiriko*
University of Surrey
United Kingdom
a.abubakar@surrey.ac.uk

Tavis Shore*
University of Surrey
United Kingdom
t.shore@surrey.ac.uk

Nishanth Sastry
University of Surrey
United Kingdom
n.sastry@surrey.ac.uk

ABSTRACT

Low Power Wide Area Networks (LPWANs) are a subset of IoT transmission technologies that have gained traction in recent years with the number of such devices exceeding 200 million. This paper considers the scalability of one such LPWAN, LoRaWAN, as the number of devices in a network increases. Various existing optimisation techniques target LoRa characteristics such as collision rate, fairness, and power consumption. This paper proposes a machine learning ensemble to reduce the total distance between devices and improve the average received signal strength, resulting in improved network throughput, the scalability of LoRaWAN, and the cost of networks. The ensemble consists of a constrained K-Means clustering algorithm, a regression model to validate new gateway locations and a Neural network to estimate signal strength based on the location of the devices. Results show a mean distance reduction of 51% with an RSSI improvement of 3% when maintaining the number of gateways, also achieving a distance reduction of 27% and predicting an RSSI increase of 1% after clustering with 50% of the number of gateways.

CCS CONCEPTS

• **Computing methodologies** → **Ensemble methods**; • **Networks** → **Cyber-physical networks**; Sensor networks.

KEYWORDS

LoRa, Machine Learning, Neural Networks, Constrained Clustering, K-Means, RSSI, Optimisation

ACM Reference Format:
Abdullahi Abubakar Kutiriko*, Tavis Shore*, and Nishanth Sastry. 2022. Constrained Machine Learning for LoRa Gateway Location Optimisation. In *The 17th Asian Internet Engineering Conference (AINTEC'22), December 19–21, 2022, Hiroshima, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3570748.3570751

* The first two authors contributed equally to this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

AINTEC'22, December 19–21, 2022, Hiroshima, Japan

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9981-4/22/12…$15.00
https://doi.org/10.1145/3570748.3570751

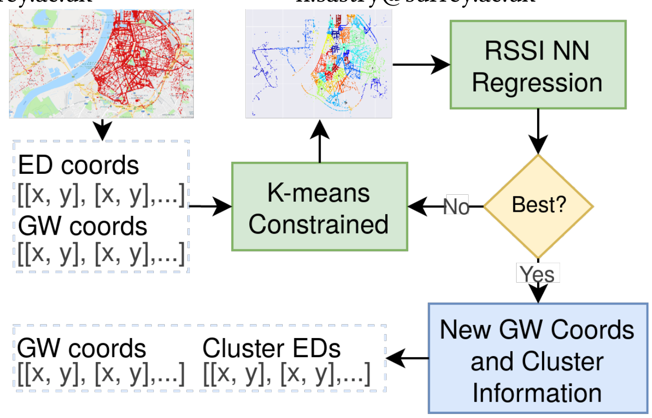

Figure 1: Ensemble's cluster assignment and verification order of execution. Constrained K-means redefines GW clusters, validating with regressor - halting after predetermined number of iterations, finally returning the optimal configuration.

1 INTRODUCTION

With the future of device connectivity and data sources appearing to be converging towards more decentralised systems, the importance of scalability and reliability have become important. The number of IoT devices in use is estimated to increase to 43 billion by 2023 [8]. As the density of IoT devices grows, the probability of communicational interference and collisions increases greatly, leading to the need to optimise in order to prevent these phenomena. This is especially becoming important in IoT mobility, where devices need to send data while in motion and can be in different locations within a short time. Our focus is on LoRa, a long-range radio frequency (RF) technology operating within the sub-GHz ISM (Industrial, Scientific, and Medical) band, initially developed in 2009 [21]. Since its inception, LoRa has grown rapidly, now used in millions of IoT devices worldwide. This is due to its ability to transmit data over large distances while requiring very little power. Given optimal conditions, LoRa end-devices (EDs) have achieved communication over distances of more than 800km [23].

As LoRa operates within ISM bands, there are minimal barriers for users to make use of these networks. This has led to large number of LoRa devices operating within the same frequency bands, increasing packet collisions and leading to deteriorating data completeness and throughput. Several features have been implemented to decrease the probability of collisions including Duty-cycling and Adaptive Data Rate (ADR). Duty cycles (DCs) determine the regularity at which transmissions can occur, the most common being 1%, meaning that many LoRa applications can communicate approximately every 2 minutes. To maximise transmission efficiency

LoRaWAN implements an adaptive data rate (ADR). This alters transmission parameters, aiming to increase the likelihood that a data frame is received by a gateway (GW) while attempting to keep power consumption to a minimum. There are two ADR methods in use: device-side and network-side. With device-side ADR, the device (using confirmed communication) initially transmits packets with the lowest power consumption possible – slowly increasing spreading factor (SF) until a confirmation acknowledgement is received. This method uses trial and error to reach the lowest power consuming characteristics available. With network-side ADR, the network calculates what it believes to be the most optimal set of parameters, and sends the information to the end-device for configuration. As the number of end-devices per GW channel reaches a certain level, approximately 5000 devices when using SF7, network efficacy severely deteriorates and Packet Delivery Ratios (PDR) can drop far below 50% [25]. This is due to a network being heavily utilised, with signals often overpowered and not optimised interfering with one other. Due to the impact that interfering packets have on the Received Signal Strength Indicator (RSSI), the PDR of LoRa transmissions decreases. This has led to the development of several optimisation techniques, [24] [4] [18] [17] [5] [1], in order to improve PDR in LoRa networks.

The push to open network technologies like The Things Network (TTN) and Helium Network has enabled the rapid development of low-cost and highly scalable applications that rely on the LoRa network. These applications include asset tracking, cattle tracking, building management, smart irrigation, and gateway coordination [7]. However, the increasing dependence on LoRa-based mobility services and the network's open nature makes it more challenging to predict and manage. Since LoRa operates within ISM bands, it is straightforward for users to utilise the network with little to no barriers. Similarly, LoRa is an ALOHA-based system, and end-devices are unaware when other end-devices transmit data. This makes it challenging to predict network utilisation, leading to competition for available space.

To address these challenges, this paper focuses on optimising GW placement, ensuring more end-devices can be covered. Due to the configurable nature of end-devices, effective network utilisation does not just rely on the GW placement but also on end-devices and their transmission characteristics. We model this as a LoRa network optimisation problem, in which constraints are placed on a machine learning (ML) algorithm to optimise the location of GWs. This method aids network congestion as the placement of GWs is calculated according to the density of end-devices within a region. By optimising the GW position, significant improvements are demonstrated in terms of increased packet delivery ratio and decreased collision probabilities, improving the overall viability of the network's scalability. This led to the development of an ML model to predict RSSI based on the location of the end-devices and GWs, and the distance between them.

The main contributions of this paper are:

- A novel ML algorithm for LoRa network architecture, producing optimised GW placement solutions.
- Configurable system for network engineers to optimise LoRa networks according to one of two different priorities – cost or function.
- A neural network mapping to estimate RSSI based on the location of end-devices and GW.

Adding an RSSI estimation algorithm is a development beyond previous studies [15][**?**], which generally use simulations to validate performance. Creating a machine learning model from real-world data allows the system to predict RSSI changes more accurately as ground truths are extracted from each region. Previous works [18][17][15][**?**] do not explicitly state such a combination of algorithms. This cyclical combination is designed to generalise well, improving RSSI estimation each time new data is fed through. It is also more flexible, enabling network managers to select a compromise between network cost and performance. Similarly, the end-device RSSI estimation model allows end-devices to estimate the best location to send data since both the location of end-device and GWs are known. The end-devices are able to estimate the probability of data reception at the GWs, thereby limiting the need for ADR and reducing the likelihood of collisions. As a consequence, end-device power efficiency is improved.

2 RELATED WORK

Since LoRa & LoRaWAN proposals have been mooted and standardised, there have been numerous studies attempting to improve network conditions and long-term prospects of the technology. Each research article included in the review has a primary objective, optimising network energy efficiency, scalability, or throughput.

2.1 GW Placement Optimisation Techniques

Optimising the location of GWs within a network is a good approach as relatively few devices require relocation to achieve potential improvements.

2.1.1 Non-ML Techniques. In [24], the author proposes two greedy algorithms to improve GW locations within a multiple GW network. Greedy algorithms always take the best immediate solution when finding an answer. They determine the global optimal for some problems but may find sub-optimal solutions for instances of other problems [4].

2.1.2 ML & Evolutionary Algorithmic Techniques. In [18], Nyirenda investigates the use of a modified Particle Swarm Optimisation (PSO) for optimising GW placement in LoRaWAN networks. The modification introduces GW distancing measures to the algorithm both during initialisation and flight time, aiming to maximise the PDR. Constraints placed upon the algorithm were to restrict GW placement into separate regions to ensure they aren't initialised too close to one another. Simulating the network resulted in the algorithm achieving a higher PDR than a traditional Particle Swarm Optimisation for a varying number of GWs, achieving higher rates than deterministic approaches when there are fewer than 49 GWs within a network.
In [17], a Fuzzy C-Means GW placement algorithm called PLACE is developed. The optimisation targets packet delivery ratio and the network's cost structure. Adding the probability that an end-device communicates with a GW other than its closest is helpful given the probability that a GW is not ready to receive a packet. A disadvantage of this method is that it does not consider real-world characteristics of LoRa, leaving the possibility of choosing too low a

cluster count. With simulations, it is discovered that the algorithm yields a decrease in capital and operational expenditure of 36%.

Researchers in [15] categorise GW deployment methods into two categories: network-aware - in which the location of end-devices is known, and network-agnostic - where the exact location of end-devices is unknown. The research aims to maximise RSSI values, using the result to determine which GW an end-device should aim to communicate with. Four phases of research were completed: GW location computation, propagation loss modelling, link assignment, and evaluation. A combination of K-Means and a grid method were used for computing GW locations in both aforementioned categories. Modelling propagation loss with the Longley-Rice path loss prediction model [**?**], terrain profiles were defined as: flat, plains, hills, mountains, and rugged mountains. These profiles determined how end-devices were assigned to GWs, employing either maximum RSSI, minimal distance, or an ILP approach. They determined that applying K-means strategies may result in competitive network performance along with reducing financial network costs.

2.2 Transmission Parameters Optimisation Techniques

2.2.1 ML & Evolutionary Algorithmic Techniques. In [5], a genetic algorithm (GA) is used to select end-device transmission parameters to optimise Quality of Service (QoS) criteria and form separable clusters of similar end-device parameters. A follow up paper *Reconfiguration of LoRa Network Parameters using Fuzzy C-Means Clustering* [1], aims to optimise within the collective QoS term: Bit Error Rate (BER), Time on Air (ToA), and RSSI. Randomly generating sets of transmission settings, a Fuzzy C-Means (FCM) clustering algorithm is applied to attain aforementioned metrics. Results were evaluated using simulations with a range of bandwidths, spreading factors (SF) and Signal-to-Noise (SNR) values resulting in well defined clusters. Showing that Fuzzy C-Means is efficient in representing clusters of QoS requirements.

[20] aims to improve LoRa efficiency, creating a closed-form formula that characterises end-device performance in terms of transmission policy before maximising the objective of this formula to increase throughput while limiting energy consumption. An optimisation problem is modelled in terms of the Packet Reception Rate with variable transmission parameters. Optimising with a GA and a simulated annealing algorithm. The proposed algorithm's throughput outperforms adaptive data rate (ADR) by 33.20%, a conservative policy by 91.81%, and the random policy by 238.8%.

DeepLoRa [14] aims to accurately estimate path loss of long-distance LoRa links over complex environments. They design a Bi-Directional Long Short Term Memory (Bi-LSTM), that learns the path loss of certain surroundings. The model has an average error of 3.56dB with a standard deviation of 3.17dB and the authors conclude that the model can accurately predict a general path loss estimation with less error than previously proposed methods.

3 TECHNIQUE DEVELOPMENT

The proposed technique is an ensemble of a constrained clustering algorithm and a regression neural network to validate newly determined clusters. We divided development into three phases: creating a clustering technique to determine optimal cells with constraints, model development to achieve accurate RSSI estimates before finally merging the models, operating iteratively to locate a potential global maximum for RSSI values against GW locations and their respective end-devices.

3.1 Problem Definition

A solution to LoRa's scalability issue is investigated here by setting an optimisation problem as the average distance between LoRa end-devices and GWs. Minimising this distance enables end-devices to use less power as they are able to lower spreading factor and transmission power. Reducing spreading factor (SF) leads to shorter Time on Air (ToA), allowing for more data uplinks (UL) as duty cycles can be obeyed with shorter transmission times. Shorter transmission path distances lead to stronger RSSI values, in turn leading to stronger Signal-to-Noise (SNR) values, enabling packets to be more reliably received. This reduces the need for retransmissions, leading to a reduction in overall traffic and the probability of collisions. Reducing collisions is of utmost importance for LoRa as packet reception failure is a strong factor in determining if it is truly scalable and suitable for long-term application. The developed application is intended to be combined with a LoRa transmission parameter optimisation technique, adjusting end-device parameters in line with new GW coordinates and path distances.

3.2 Data

We selected the dataset *Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas* [3] containing over 100,000 examples of LoRa uplinks. The data was collected around Antwerp city centre by attaching end-devices to postal vans as they delivered across the region. The end-devices used on each of the vehicles were identical, allowing for analysis under the assumption that all characteristics are equal. Distances between end-device and GW ranged from under 100m to over 4km. The area of collection was approximately 6km by 8km which is assumed to be a 2D plane in this research. Each uplink packet is assumed to be from an individual end-device for this analysis. As data was collected from moving end-devices, different reception characteristics were experienced from each location. The assumption is made that the transmissions are equal to if they had been received by a static end-device in each location. end-device transmission locations are displayed in Figure 2.

3.3 Clustering Analysis

As LoRa clusters are defined by their distance to the nearest GW, cluster analysis is an ideal method for re-configuring a GW location. Figure 3 shows transmissions, colour coded into the GWs that originally received them. These clusters are ill-defined and many packets were not received by their nearest GW.

K-means is employed for a baseline GW cell redefinition. It can readily scale to larger datasets, ideal for making the ensemble applicable to large-scale networks. Convergences are guaranteed, although not necessarily producing optimal clusters. The maximum numbers of end-devices per LoRa GW are contested within research, and different implementations have different optimums. end-devices have different hardware characteristics and transmit at different rates, with different payloads and different modulation

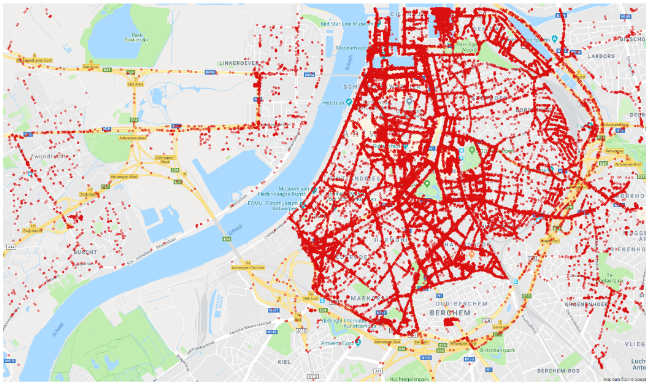

Figure 2: Location of each transmission in Antwerp. All transmissions occur on the road or car park as the end-devices were attached to postal vans.

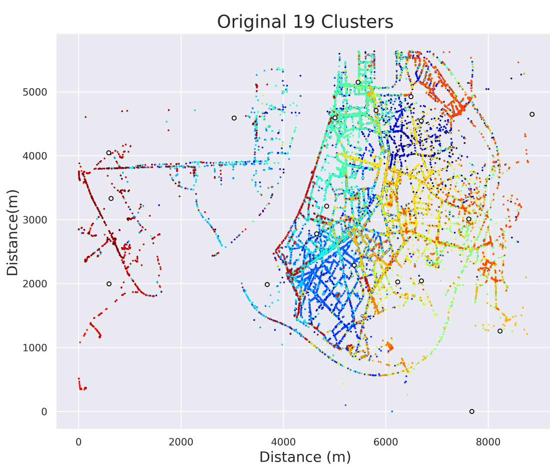

Figure 3: Original GWs white dots with a black outline, their respective end-devices separated into colour coded groups.

chirps. This leads to each network having different characteristics. However, this research does not consider the real-world differences between device hardware characteristics. This assumption does not affect the applicability of results since we are using a real-world dataset to train the model with over 100,000 unique data samples and over 19 GWs.

We used the elbow method to determine an optimal number of centroids for clustering, not considering LoRa network constraints. Seven GWs were deemed optimal for the minimal number of clusters that gain a good minimal overall distance between centroids and end-devices. Figure 4 displays the outcome of implementing 7 centroids, resulting in well-defined clusters containing nearly 16,000 end-devices per GW, possible for LoRa networks but would result in a weak Packet Delivery Ratios (PDR) due to high probability of collisions [2].

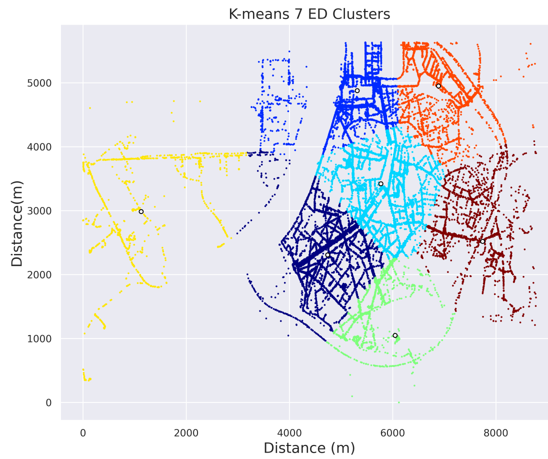

Figure 4: Outcome of redefining clusters using elbow method determined number of centroids

Figure 5 displays the outcome of selecting 19 clusters, the original number of GWs - selected to display the improvement, and how clusters may appear after transmission parameters are reduced accordingly.

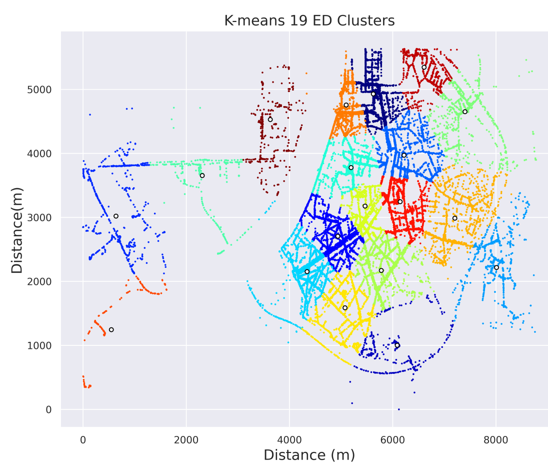

Figure 5: K-means clusters with equal number of GWs to the original configuration

3.4 Clustering Constraints

A constraint is placed on the clustering, setting limits for the maximum number of end-devices per cluster in order to reduce the likelihood of collisions. Also setting a minimum number of end-devices per cluster to prevents the algorithm from placing excess GWs, incurring unnecessary costs.

Table 1: K-Means Clustering Improvements

		End-device per GW Decrease		Distance Decrease	
GWs	Technique	Med	Mean	Med	Mean
20	K-Means	2.0%	20%	55%	60%
20	K-Means C	-16%	20%	47%	51%
40	K-Means	59%	60%	75%	74%
40	K-Means C	54%	60%	74%	71%
60	K-Means	72%	73%	82%	80%
60	K-Means C	67%	73%	82%	76%
80	K-Means	83%	80%	83%	82%
80	K-Means C	77%	80%	83%	78%
100	K-Means	84%	84%	88%	85%
100	K-Means C	82 %	84%	84%	77%

We used the python package *k-means-constrained* to add these constraints [13]. This package builds on sklearn's clustering algorithm, adding a modified version of the Minimum Cost Flow (MCF) Network proposed by Bradley et al. [6], originally proposed to improve k-means on datasets with a large number of dimensions, it is used here due to its multi-functionality.

Existing literature, [16] [2], determines initial upper limits on end-devices per GW used in this research. Through physical experimentation, [16] determines that with 1000 end-devices, a 3-channel GW can receive nearly 100 packets per hour from each end-device, more than sufficient for the majority of LoRa applications. 1000 end-devices per GW is used as a baseline in this research, going on to assign more per cluster as ubiquitous 8-channel GWs are capable of communicating with proportionally more end-devices than 3-channel GWs. Furthermore, various duty cycles do not allow such high rates of reception, leading to even more end-devices per GW being viable.

Various clustering parameters were tested using the measures: average distances between end-devices and GWs, and the average number of end-devices per GW. A sample of clustering improvements are displayed in Table 1. An improvement is not seen for the median number of end-devices per GW in the K-means constraints until the number of GWs increases above 27 due to one GW (car park GW) in the original configuration being responsible for a large proportion of end-devices, skewing the median in comparison to the mean.

Applying a maximum number of end-devices per GW as 1000 produced clusters shown in Figure 6, where the minimum number of end-devices per GW was 200. The graph shows how the density of end-devices affects clustering, with sparse end-devices in the left-hand side being served by fewer GWs, and clusters further towards the city centre covering much smaller areas. This resulted in a decrease in mean end-device per GW of 87% and a decrease in mean distance of 82%.

Figure 7, shows 51 clusters with a 2500 end-device maximum limit for comparison. Clusters are larger and less dense. This network would cost considerably less than that with 126 GWs. This experiment resulted in a decrease of mean end-device per GW of 69% and a mean distance decrease of 70% in comparison to the original network.

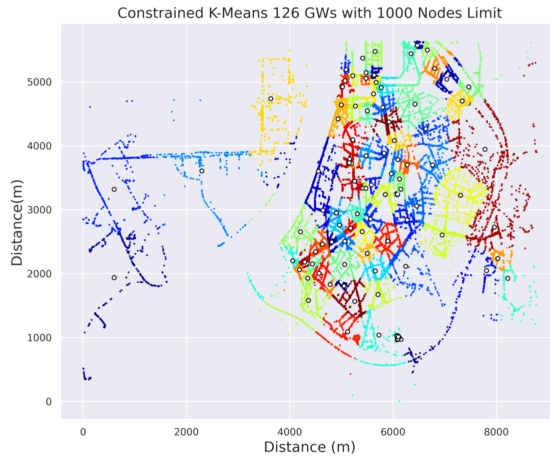

Figure 6: 1000 end-device Limit clusters, set with researched limits

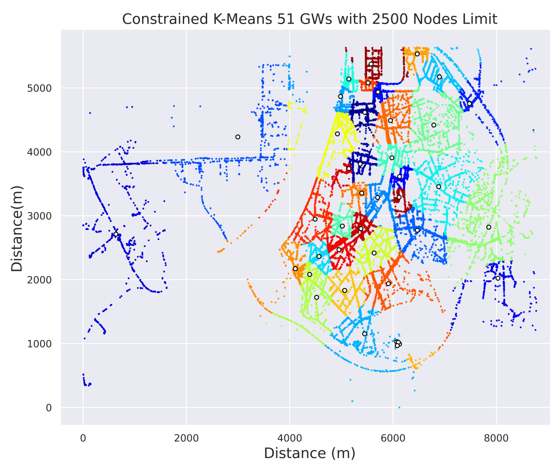

Figure 7: 2500 end-device limit clusters, for comparison due to GW cost implications

The standard K-Means algorithm performs well, reducing mean and maximum distances. However, the number of end-devices per GW can reach unfeasibly high values. Using the same number of GWs as the original network, constrained K-means determined clusters with a maximum number of end-devices per GW peaking at 5,752. Imposing limits on the number of end-devices that can be serviced by a GW is shown here to be significantly important during clustering. This is due to the affect it has on positioning, ensuring that regions with dense end-devices have more GWs to meet demand.

3.5 Clustering Validation

To estimate values of RSSI between end-devices and new GW coordinates, a regression neural network model was developed. According to the universal approximation theorem [10], neural networks with as little as one hidden layer, given enough neurons in this layer, can approximate any function.

Various architectures were tested, starting with a simple 3 layer neural network consisting of an input layer with 5 neurons - matching the number of features, a hidden layer with 12 neurons, and an output layer with one neuron. For hidden layers, *ReLU* was used as the activation function to solve the vanishing gradient problem. However, neurons can become inactivated due to negative values becoming zero. Different widths and depths for the network were tested. Wide networks are able to train well to data whereas deep networks are able to extract information from between features and form more abstract insights. Existing research was considered to determine tuning techniques. The paper [12] analyses deep regression, comparing the effects that altering hyper-parameter tuning has on various aspects of architectures - covering batch normalisation, drop ratios, and regression layers. [9] Follows the iterative design of NNs displaying stages taken to improve models, and [19] studies different architectures for a regression model, further testing the dropout function.

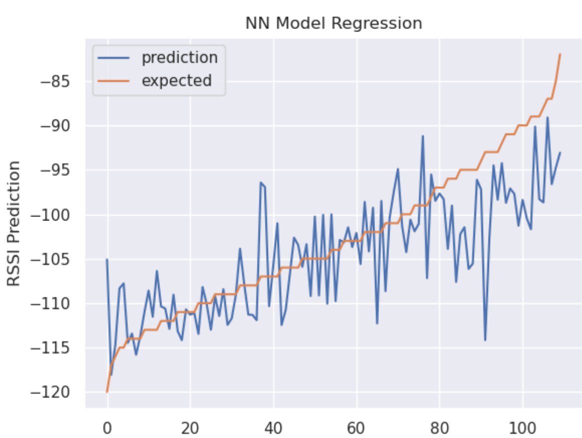

Figure 8: Neural Network's conformity to validation data

A network with 5 full-connected layers of widths 5-10-20-20-10-5-1 was created after extensive experimentation, achieving the best results capable within the research time-frame. Trained over 200 epochs with the *Adam* optimiser, a batch size of 32 and initial learning rate of 0.001. This model achieved an Root Mean Square Error (RMSE) of 1.21, Mean Absolute Error (MAE) of 0.71 and an R-squared (R^2) value of 0.96. The resulting regression and distribution of estimates are shown in Figures 8 and 9.

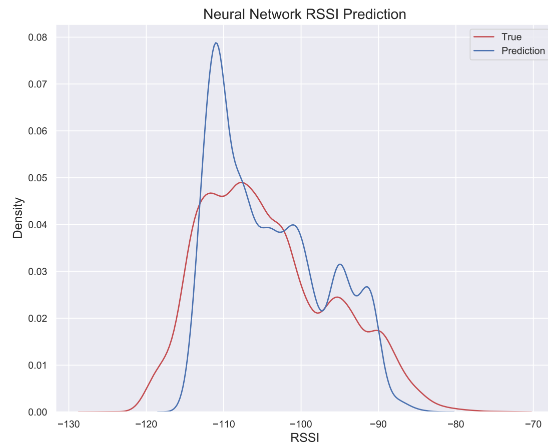

Figure 9: Validation distribution of RSSI estimates relative to true values

3.6 Ensemble Collation

The constrained K-means and regression models were combined to form an algorithm set inside a looping function taking the arguments: min number of end-devices per GW, max number of end-devices per GW, and the max number of iterations. This formed the brute-force algorithm which cycles through parameter combinations, determining the improvements of both distance and RSSI values. Algorithm 1 outlines the method.

3.7 End-device RSSI estimation

Effective network utilisation relies not only on the GW placement but also on the end-devices and how they transmit data. In the LoRa network, adaptive data rate (ADR) optimise data rates, airtime, and energy consumption while ensuring that messages are still received at gateways. end-devices close to the GWs use a lower spreading factor and higher data rate, while end-devices further away use a high spreading factor because they require a higher link budget.

But adaptive data rate is primarily applicable when the end-devices are placed in a fixed location since adaptive data rate is calculated based on the received signal at a particular location. When the end-device moves to a new location, adaptive data rate will have to be recalculated for the new location, which makes adaptive data rate inefficient for mobile end-devices. This means mobile devices usually send data without utilising the benefits of adaptive data rate. But thanks to the GPS sensor attached to many mobile applications and open access to GW data, the data required for prediction are becoming easily accessible. Using neural network with GW and end-device location as input, we were able to predict the RSSI at the GW with an accuracy score of 70%. Figure 9 shows the results with the RSSI values falling within the acceptable transmission range.

Algorithm 1 GW Location Ensemble Algorithm

Input: Packet reception data D, Max iterations I, Min/Max end-devices per GW ED_{min}, ED_{max}, Improvement Threshold T
while $i < I$ or $t < T$ **do**
  Determine iterative parameters, obtain new locations and evaluate
  $clusters =$
    $range(length(D)/ED_{max}, length(D)/ED_{min})$
  $min_{ed} = range(ED_{min}, (ED_{max} - ED_{min}))$
  for $c = 1, ..., clusters$ **do**
    for $m = 1, ..., min_{ed}$ **do**
      $c_{data} = newclusters(D, c, m, ED_{max})$
      Determine distance improvement
      $R_{mean}, R_{med} = RSSI_{model}(c_{data})$
      $data_{out} \leftarrow [c, ED_{min}, ED_{max}, C_{mean},$
        $C_{med}, R_{mean}, R_{med}]$
      $metric = max(R_{mean}, R_{med}, C_{mean}, C_{med})$
      if $metric > t$ **then**
        $t = metric$
      end if
    end for
  end for
end while
Output: $data_{out}$ Dataset containing new GW information from experiments

4 RESULTS

Results of experiments with differing numbers of GWs are given in Table 2. The neural network's performance is effective where the number of GWs is similar to that in the original data, with further model tuning required to improve results. A reason for lower than expected performance is that modelling focuses heavily on coordinates given in training data and less on distance between points which has a larger affect on RSSI. Current system results are functional for real-world application due to the known relation between RSSI and distance [22].

The ensemble will enable network operators to improve GW placement, ensuring that overall distance between EDs decreases. Figure 10 displays improvements as the number of clusters is altered. This may be used for determining the optimal number of GWs for a specific network, selecting a trade-off point between distance decrease / RSSI increase and the number of GWs installed, increasing network coverage while minimising total cost of GWs. RSSI improvements can be analysed to consider reducing spreading factor values. Lower values lead to reduced transmission time and power consumption - strengthening the argument for optimising GW placement to ensure LoRa's long-term success.

Similarly, placing the NN model on end-devices enables route planning within a predefined location based on the data. This allows end-devices to send a payload with high probability of success leveraging the NN . Again, a generalised model is achievable with access to more data. But since most end-devices are made up of constrained devices. We believe it is more applicable to build miniature models for specific locations leveraging models like the tiny ML [1].

[1]https://www.tinyml.org/

Table 2: Ensemble Experiment Results

	Distance (m)			Dist Decrease		RSSI Increase	
GWs	Med	Max	Mean	Mean	Med	Mean	Med
20	213	2365	266	51%	47%	3%	4%
30	149	2302	197	64%	63%	4%	6%
40	104	2181	160	71%	74%	5%	7%
50	93	2178	145	73%	77%	5%	7%
60	74	2272	130	76%	82%	5%	8%
70	66	2128	123	77%	83%	5%	8%

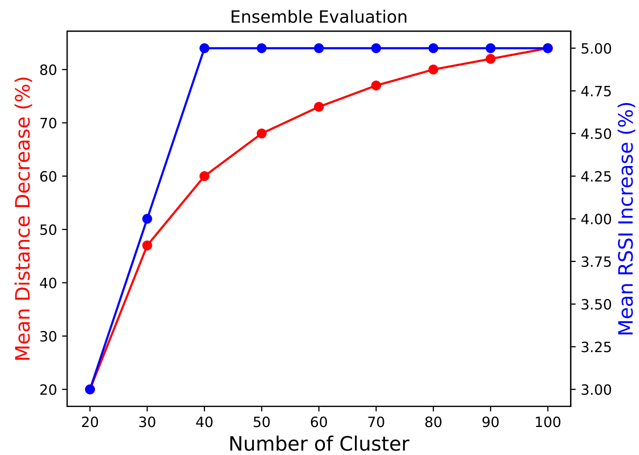

Figure 10: Ensembles optimisation evaluation across different numbers of clusters

5 CONCLUSION & FUTURE WORK

The proposed system successfully shows that the ensemble of a clustering algorithm and an evaluation algorithm may achieve improvements in LoRa performance. The work viably adds practical and applicable knowledge to the field of IoT transmission optimisation, proposing a self-contained ensemble of machine learning techniques to iteratively improve network performance. Although we would like to test the generalisation of the developed model, we do not currently have access to further localised datasets for research. The end-device RSSI estimation simplifies optimising LoRa network by informing route planning for applications, thereby increasing the network's overall efficiency.

5.1 Conclusion

While maintaining the same number of GWs as in the original data, the proposed algorithm was able to reduce the mean distance between end-devices and GWs by 51%, also reducing maximum and minimum distances. For this specific LoRaWAN network, if the number of GWs may be increased from 19 to 40, the algorithm is able to reduce the median number of end-devices per GW by 60%, resulting in a large improvement to the PDR. Some GWs in the original dataset were heavily under utilised, resulting in an unequal strain across the network. This algorithm is able to redistribute the load into a more balanced mapping, increasing reception fairness. end-device transmission parameters would then be tuned to communicate with the allocated GW. In the original dataset, many uplinks were consistently received by GWs other than the nearest. Reducing spreading factor and transmission power of an end-device will reduce the chance of this occurring. Further reducing the probability of packet collisions, reducing the overall power consumption requirements, increasing the possibility of LoRa being scalable and a long-term IoT solution. These values will be reducible as signals do not have to reach as far.

5.2 Future Work

There are various avenues of research that will be taken to continue developing this technique. The first stage will be to collect more diverse, accurate, and extensive data. With more examples of uplinks, the system may generalise further, producing better estimates for RSSI values. Another alteration will be the addition of transmission parameter values within data frames. After regrouping end-devices to near GW locations, this would allow for quantitative predictions for power consumption reduction as the power usage depends heavily on the spreading factor. By collecting data specifically for this research, failed uplinks will be stored, enabling GW placement to be computed with another constraint which will prevent end-devices with failed uplinks to be beyond a determined distance from a GW. This research considered the dataset as if it were a specific instance in time, not temporally different packets. With a more extensive collection of uplinks, allocation of end-devices to GWs will consider end-devices differing uplink intervals. This will further improve clustering as areas with frequent uplinks require a more dense GW architecture than areas with irregular uplinks. Another research interest we are considering is adding constraints for the GW locations. Currently, the GWs are placed in the centroid, which can be any location within the location. However, since some of the locations might not be feasible to place a GW, we are looking at adding geo constraints to ensure the GW are not placed, for instance, on the road and buildings.

REFERENCES

[1] Abdullahi Kutiriko Abubakar, Nishanth Sastry, and Mohamed Kassem. 2021. *Internet islands: first class networked communities in isolated regions.* In Proceedings of the 16th ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch '21.

[2] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. 2017. Understanding the limits of lorawan. (2017).

[3] M. Aernouts, R. Berkvens, K. V. Vlaenderen, and M. Weyn. 2019. Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas. (2019). https://doi.org/10.5281/zenodo.3904158

[4] Paul E. Black. 2005. greedy algorithm. (2005). https://xlinux.nist.gov/dads/HTML/greedyalgo.html

[5] M. Bor and U. Roedig. 2017. Lora transmission parameter selection. (2017).

[6] P. Bradley, K. Bennett, and A. Demiriz. 2000. Constrained k-means clustering. (2000). https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/

[7] A. Djoudi, R. Zitouni, N. Zangar, and L. George. 2020. Reconfiguration of lora networks parameters using fuzzy c-means clustering. In *2020 International Symposium on Networks.* Computers and Communications (ISNCC, 1–6.

[8] Dahlqvist Fredrik, Patel Mark, Rajko Alexander, and Shulman Jonathan. 2022. Growing opportunities in the Internet of Things. (2022). https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things

[9] R. Gore, J. Li, M. Manry, L. Liu, and C. Yu. 2005. Iterative design of neural network classifiers through regression. *International Journal of Artificial Intelligence Tools* 14 (2005), 281–302.

[10] K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. (1989).

[11]]b13 G. Hufford, A. Longley, and W. Kissick. [n. d.]. A guide to the use of the its irregular terrain model in the area prediction mode. 16 ([n. d.]). https://www.ntia.doc.gov/files/ntia/publications/ntia_82-100_20121129145031_555510.pdf

[12] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud. 2020. A comprehensive analysis of deep regression. (2020).

[13] Josh Levy-Kramer. 2022. k-means-constrained. (2022). https://pypi.org/project/k-means-constrained/

[14] L. Liu, Y. Yao, Z. Cao, and M. Zhang. 2021. Deeplora: Learning accurate path loss model for long distance links in lpwan. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications.* 1–10.

[15] R. Mina, H. Maryam, R. Zaidi, and S. Ali. 2019. Computational methods for network-aware and network-agnostic iot low power wide area networks (lpwans). *IEEE Internet of Things Journal* 6, 3 (2019), 5732–5744.

[16] J. H. I. Moerman and J. Hoebeke. 2017. Lora scalability: A simulation model based on interference measurements. *Sensors* 17 (2017), 6.

[17] N. C. M. Neto. 2020. *Fuzzy c-means based GW placement algorithm for lorawan.* Ph. D. Dissertation. Guam a University Campus.

[18] C. N. Nyirenda. 2021. *On the efficacy of particle swarm optimization for GW placement in lorawan networks.* in intechopen.

[19] A. Perkov, T. Zhangirov, A. Liss, and N. Grigoryeva. 2020. Peculiarities of regression model design based on neural networks. (2020).

[20] R. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro. 2019. *Performance optimization of lora nodes for the future smart city/industry.* in EURASIP Journal on Wireless Communications and Networking.

[21] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent. 2016. *Lorawan specification v1.0.1.* Tech. Rep.

[22] Z. Tan, M. Rahman, A. Rahman, A. Hamid, N. Amin, H. Munir, and M. Zabidi. 2019. Analysis on LoRa RSSI in Urban, Suburban, and Rural Area for Handover Signal Strength-Based Algorithm. (2019).

[23] The Things Network Global Team. 2020. LoRa World Record Broken: 832km/517mi using 25mW. (2020). https://www.thethingsnetwork.org/article/lorawan-world-record-broken-twice-in-single-experiment-1

[24] H. Tian, M. Weitnauer, and G. Nyengele. 2018. Optimized GW placement for interference cancellation in transmit-only lpwa networks. *in sensors* 18 (2018), 11.

[25] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi. 2018. Throughput, Coverage and Scalability of LoRa LPWAN for Internet of Things. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS.* 1–10.