Delay Optimization In Smart Health Systems By Employing Dynamic Scheduling Approach With Gaussian Mixture Model

Aftab Ali[1], Muhammad Javed[2], Muhammad Asad Khan[1], Nosheen Jelani[1], Muhammad Ijaz Khan[2]

[1]Gomal Research institute of computing (GRIC), Faculty of Computing, Gomal University, Dera Ismail Khan, Pakistan.
[2]Department of Computing and Information Technology, Faculty of Computing, Gomal University, Dera Ismail Khan, Pakistan.
* Corresponding author; Aftab Khan Email Address: aftab.ali01@nbp.com.pk

Abstract

With the increase in the number of Internet of Things (IoT) smart devices drastically, Low Power Wide Area Network (LPWAN) technologies have become an overwhelming choice worldwide. The researchers have used a variety of LPWAN technologies to solve difficulties like higher collision rates, retransmissions, delays, and energy usage. In contrast, the most appealing and appropriate technology in terms of energy efficiency, cheap cost, and delay optimization is Sigfox. The primary problem with Sigfox is the high percentage of packet drops caused by collisions. The Pure Aloha MAC technique, which Sigfox uses to transmit frames or data readings, is the main cause of this packet drop rate and ultimately retransmissions. Many retransmissions result from communication between Sigfox smart devices and Pure Aloha. The delay in Sigfox network has increased even further, with the increase in the number of retransmissions. This work uses the Gaussian Mixture Model (GMM), an unsupervised probabilistic technique, in conjunction with a Dynamic Scheduling Approach (DSA) to optimize the delay in Sigfox network. Retransmissions are decreased when DSA is used in conjunction with GMM, optimizing the latency in Sigfox. According to the findings, our method reduces Frame Collision Rate (FCR) by 15% when compared to traditional Sigfox. Furthermore, a 39% increase in Frame Success Ratio (FSR) is seen when comparing the traditional Sigfox. Moreover, 79% of the delay is optimized. This study may be useful in situations when patients' vital information must be transmitted to gateways with the least amount of delay and with optimal retransmissions.

Keywords: LPWAN, Sigfox, Gaussian Mixture Model, Fuzzy logic, Profiling, Collision, Adaptive Data Rate, End Device, Smart Node, Quality of Service, Frame Success Ratio, IoT, Machine Learning, Unsupervised Learning

I Introduction

The Internet of Things (IoTs) has given rise to a broad range of protocols and technologies that address its connectivity requirements. WiFi has been used for consumer networking, but not one of those technologies designed for IoT applications. Using gateways, LPWAN distributes radio coverage across a wide region while modifying transmission power, rates, modulation schemes, and other factors to reduce the amount of power that Smart Nodes (SNs) need to use to connect to their connections. Several LPWAN standards are used and deployed by authors to solve different kinds of problems. In terms of cost and energy usage, however, Sigfox is the most appealing and appropriate. Due to high frame drop rates, from collisions in extremely congested situations, Sigfox is susceptible to higher delay times [1]. The MAC technique, or Pure Aloha, that Sigfox uses to send data frames is the primary cause of this packet failure rate. Retransmissions are common when SNs in Sigfox establish connectivity with Pure Aloha. Retransmissions like these cause issues with increased latency and power usage. The limited lifespan of SNs and their high density in the LPWAN pose a major obstacle to effective packet delivery. The current LPWAN architecture has several problems, including a high real-time Frame Error Ratio (FER), degraded performance because of collisions, increased number of retransmissions, and a mitigated throughput. It has been utilized for large-scale data shared smart health applications. The transmission latency rises because of this. This issue, which directly affects the performance of such networks, is simplified using the Dynamic Scheduling Approach (DSA) in conjunction with the Gaussian Mixture Model (GMM), an unsupervised learning technique [2]. Moreover, the gateway might have a DSA installed to give priority to frames with varying profiles. The gateway oversees establishing the transmission intervals for various profiles. Due to the adaptive learning algorithm's notable improvement in the success percentage of frames from SNs, there are fewer retransmissions, which lowers delay.

The goal of the unique LPWAN protocol proposed in [3] was to maximize the potential of smart energy distribution devices. An uplink multi-hop communications technique, which uses less energy, has been used in this investigation. A multi-hop method has been reported to save up to 15% of energy consumption while preserving the same degree of network reliability on an actual testbed. The authors of [4] proposed a procedure for handling LoRaWAN packets and doing out IoT SN characterization. Specifically, SNs are categorized using the k-means technique according to their radio characteristics. ADR and BADR (Blind Adaptive Data Rate) were studied in a mobile environment in [5], and their shortcomings were noted along with the introduction of a novel ADR Retransmission algorithm. To complete their separate tasks, the network server and SN's algorithms operate simultaneously.

Critical data from patients with serious diseases is gathered in a smart health monitoring scenario to account for any delays. This is examined in [6], where a substantial number of SNs make up a LoRa network that has been tested and proven for use in a medical monitoring system. SNs are linked to a variety of individuals within this network, including staff, medical physicians, and patients. Researchers working with intelligent learning in LoRaWAN and Sigfox have significant methodological concerns. An intelligent learning system must be built since the number of SNs in a LoRaWAN increases exponentially the severity of performance related difficulties [7]. However, the usage of intelligent learning in Sigfox is fast growing since it is so effective for Internet of Things applications such as smart health monitoring.

This article describes how Sigfox has been configured in residential areas and how the Sigfox network enhances performance of smart health systems. With the goal of optimizing delay by lowering the number of retransmissions, a novel algorithm is introduced. To profile the SNs, an unsupervised technique called GMM with K-means is employed. Based on information received through the gateway, an ED is given a probability using a Probability Density Function (PDF). SNs classification into the Critical Patients Profile (CPP), Semi-Critical Patients Profile (SCPP), and Normal Patients Profile (NPP) is determined by this probability. GMM is a soft profiling technique for unsupervised learning. Soft profiling has the advantage of basing judgements on probability regarding SNs. GMM's elliptical form, which covers a larger region than other unsupervised techniques, is another advantage. Furthermore, DSA arranges traffic flow amongst every profile. Patients with critical readings will be assigned precedence in the CPP SNs. SNs with normal readings are included in NPP, whereas SNs with semi-critical readings are included in SCPP.

The main contribution of this manuscript is as follows:

1. To evaluate the performance of traditional Sigfox by using Pure Aloha MAC Scheme in terms of Frame Collision Rate (FCR), Frame Success Rate (FSR) and Transmission Delay (TD).
2. To enhance the performance of traditional Sigfox by introducing unsupervised intelligent learning approach Gaussian Mixture Model (GMM), to design profiles for patients.
3. A novel approach Dynamic Scheduling Algorithm (DSA) with GMM, is proposed to intelligently schedule traffic, which ultimately enhances performance in terms of retransmissions and delay.

II Literature Review

To improve the latency specifically in traditional Sigfox, several articles are highlighted in this section. Moreover, it investigates how Sigfox uses unsupervised learning or intelligent learning. Based on the Okumura-Hata propagation model, the authors in [8] examine the path-loss factor between the measured and anticipated readings. In addition, they also calculate Mean Square Error (MSE) for both projected and measured path-loss values. Article [9], specifically considers traffic that has multiple gateway deployments, with and without acknowledgements. Additionally, authors also examine, how Sigfox can improve performance in terms of latency and serve end devices with a 90% packet success rate when configured properly.

Authors in [11], performs several experiments to analyze end-to-end delay in Sigfox networks. Overall, the measurements are taken from the transmitter to the server into account. Authors worked on two cases in this article. The first was a static experiment with a 100% delivery rate and a typical delivery duration of 2.5 to 4.5 seconds. In the second, mobility is taken into consideration. An end-to-end latency of up to 8 seconds and a delivery rate of 20% is observed. The behaviour of the LPWA network in terms of retransmissions was examined in article [12]. Even so, LoRa networks support up to eight packet retransmissions, which aid in the recovery of dropped frames. However, as EDs are compelled to send more packets, the network lifetime also drastically reduces. To strike a balance between the expected network lifetime and the number of packet retransmissions, it would be imperative to foresee the consequences of each retransmission in a network.

In [14], a decoding technique that permits superposed packets was put forth. The packet success ratio is further decreased by collisions, and high latency is also experienced because of retransmissions. In general, the suggested decoding strategy works better together than the traditional Sigfox performance. The impact of machine learning approaches was examined in [16] to enhance performance. A mechanism for processing LoRa network packets was used to accomplish device profiling and frame arrival prediction to meet this goal. The examination focusses on unsupervised learning techniques, such as decision trees, k-means, and Long Short-Term Memory (LSTM) neural networks. The study demonstrated how profiling approaches allow a machine learning prediction algorithm to function even in situations where certain devices perceive high error rates, making training impossible.

An unsupervised learning strategy was presented by researchers in [17] to prioritize packets at various levels. A thousand smart nodes deliver data to the gateway on average. Several clusters are extracted using K-Means, an unsupervised learning technique, based on data obtained from smart apps such as humidity and weather temperature. Depending on the reading that was received, different weights were determined. The placement of smart nodes in various clusters is influenced by these weights. All things considered, this strategy improves performance in terms of energy and latency. Utilizing the Priority Scheduling Technique (PST), the outcome demonstrates that it significantly reduces consumption and delays. While the gateway can set the nodes' transmission intervals based on the transmission priorities of the dynamic PST. Different simulations are performed to show the behavior of throughput, error rate, delay and energy consumption.

For Internet of Things applications such as smart gas and energy meters and smart health monitoring, Sigfox is a better fit. Sigfox is widely used because of its ease of use, great range, low battery consumption, and minimal amount of end devices calculations required for data transmission. Sigfox is a must for Internet of Things applications where low latency is not as important. However, most of the time, high collision rates, poor throughput, a high number of retransmissions, and excessive delays are the main challenges that researchers encounter with Sigfox. Additionally, authors in [19] presented the EXPLoRa-SF and EXPLoRa-TA SF allocation systems. With improved ToA, these techniques offer minimal interference in cluster-based environments. Additionally, EXPLoRa-SF assigns the same SF and successfully completes gearbox without encountering any collisions. According to the simulation, large SF values offer long coverage, but they can also occasionally lead to a significant number of collisions, which increases delay.

Table 1 provides analyses of these schemes with single-hop LoRaWAN arhitecture.

Table 1. Research Studies Focuses On Delay

Ref.	System Architecture	Techniques	Achievements	Limitations
[11]	Single-Hop System Architecture	Pure Aloha MAC Scheme.	Confirmed and Un-Confirmed traffic are both consider having multi-gateway scenario. Enhanced performance in terms of PSR and mitigate delay.	In case of mobile devices, the losses are on a higher side which ultimately enhance delay.
[12]	Single-Hop System Architecture	CSS Modulation Scheme, Pure Aloha MAC Scheme	A novel mathematical model is introduced to analyze the effect of re-transmissions which ultimately enhance delay	Re-transmissions drastically effect the performance of LoRa network in terms of delay
[13]	Single-Hop System Architecture	CSS Modulation Scheme, Pure Aloha Scheme Used	Perform experimentation in real environment.	Lack of Mathematical expressions.
[14]	LoRa Single-Hop System Architecture	Pure Aloha Scheme	Investigates the feasibility of using the LPWAN protocol LoRaWAN with an event-triggered control scheme.	Path Loss and propagation delay are not addressed.
[15]	Single-Hop System Architecture	Pure Aloha with decoding algorithm	In the event of collisions, throughput is reduced due to packet loss and retransmissions occur. Overall, the energy consumption and delay is minimized through reinforcement learning adaptive approach.	In case of movable end devices, the losses are on a higher side, ultimately enhance retransmissions and delay.
[22]	Single-Hop System Architecture	CSS Modulation Scheme, Proposed MAC Scheme with ADR	Analyze the expected delay and the expected energy required to join (OTAA) the network. -The behavior of collision is thoroughly observed which ultimately affects the delay as well. The proposed MAC scheme overall outperforms the performance in terms of delay by mitigating the collision factor.	Targets delay and energy consumption only at the time of joining the LoRa Network.

The authors of [20] suggest using Reinforcement Learning to increase end device throughput. The Sigfox network's performance is significantly influenced by the configuration options set for end devices. The mentioned scheme analyses the optimal parameters and updates them accordingly. To determine the optimal end devices settings based on the Q function, the Reinforcement Learning algorithm is applied on the gateway.

III Impact of Un-Supervised Learning Techniques on LPWAN

Because of its low power consumption and long-range communication capabilities, Low Power Wide Area Networks (LPWAN) are critical for enabling large-scale Internet of Things applications. The following are some major ways that integrating unsupervised learning techniques might improve LPWAN system performance, efficiency, and capabilities:

A K-Means

Low Power Wide Area Networks (LPWAN) can greatly improve network performance and efficiency by implementing K-Means clustering. Devices are grouped using K-Means clustering according to parameters like geographic proximity or signal intensity, creating clusters led by selected cluster heads. By serving as middlemen, these cluster heads reduce the number of direct transfers to the central gateway, saving bandwidth and energy. Cluster heads also lessen the quantity of data delivered by preprocessing and aggregating data locally, which aids in ensuring good resource utilization and network traffic management. Because of its hierarchical structure, the network is more scalable and can accommodate the addition of new devices with ease. Additionally, by spotting anomalies and unusual patterns inside clusters, K-Means clustering improves security and fault detection and helps with anomaly identification. Improved data quality, balanced load distribution, and extended device lifespan are just a few advantages of implementing K-Means clustering in LPWAN, which makes it a potent tool for optimizing LPWAN systems despite obstacles like figuring out the ideal number of clusters and managing dynamic network topologies.

The objective function for the K-means profiling algorithm is the squared error function:

$$K\_M = \sum_{p=1}^{k} \sum_{q=1}^{n} (|x_p - y_q|)^2$$

where,

$|x_p - y_q|$ is the Eucledian distance between a point, x_p, and a centroid, y_q, iterated over all k points in the i^{th} cluster, for all n clusters.

B Fuzzy Probabilistic Model

An extension of probability theory for handling mixed probabilistic and non-probabilistic uncertainty is fuzzy probability theory. It offers a theoretical foundation for modelling uncertainty, which is only partially defined by randomness and resists a purely probabilistic modelling of certainty because of the unreliability, imprecision, or absence of relevant data. The fuzzy probabilistic model is a compromise between non-probabilistic uncertainty models and the probabilistic model. Fuzzy probability theory is significant because it treats population elements as set-valued quantities or uncertain granules rather than as crisp quantities, which is more realistic in most real-world scenarios. Thus, uncertainty that is both probabilistic and non-probabilistic can be appropriately and independently transferred to the outcomes of a later investigation.

There are four components to its architecture:

1. Based on linguistic data, the experts have created a set of rules and IF-THEN conditions to guide the decision-making mechanism. Fuzzy controllers may be designed and tuned using a variety of efficient techniques thanks to recent advances in fuzzy theory. Most of these advancements result in fewer

ambiguous rules.

2. This process turns inputs, such as discrete numbers, into fuzzy sets. Crisp inputs are essentially the precise inputs—temperature, pressure, rpms, etc.—that are measured by sensors and sent to the control system for processing.

3. The INFERENCE ENGINE selects which rules should be executed based on the input field by calculating the degree to which the current fuzzy input matches each rule. The control actions are then created by combining the fired rules.

4. This process turns the fuzzy sets that the inference engine produces into a precise value. To lower the error, the most appropriate defuzzification technique is combined with a particular expert system.

The authors in this manuscript use Gaussian Mixture Model (GMM), to assign probabilities to SNs. This is because of its flexibility and robustness specifically in smart health environments. Below are the details of GMM with Dynamic Scheduling Approach.

IV Dynamic Scheduling Approach with Gaussian Mixture Model

The suggested system paradigm for smart health monitoring is depicted in Figure 1. Each patient in this system is fully equipped with 3 wearable devices to monitor blood pressure, pulse oximeter and heart rate. In this scenario, a 5 km^2 area has two Gateways or Relay Devices (RDs) installed, and all the SNs are dispersed randomly. All the SN_js are first configured using the Sigfox device named "BioT Medical IoT Device" (The device is responsible to monitor heart rate, blood pressure, and pulse oximeter readings, where j varies from 1 to 3000. To interact with the RD, all the SN_js are static [23]. On gateway or RD, the GMM approach is used.

Once the best profiles are chosen by the GMM algorithm, each profile is assigned a SN based on the PDF. Based on DSA, data readings from different geographical regions or profiles are prioritized in terms of CPP, SCPP, and NPP. The notion of adding two RDs to our Sigfox network is necessary to handle sensitive data containing patient measurements.

We must install these RDs where SN interference is as minimal as feasible because RDs have the potential to occasionally enhance interference [24]. Utilizing multiple RDs also has the advantage of not requiring energy usage because Pure Aloha is utilized for SN to RD frame transport. Furthermore, thousands of nodes [25], can be served by Sigfox RDs using the differential binary phase shift keying (D-BPSK). The key difficulty is to achieve a greater frame acceptance rate with this enormous No. of SN broadcasting towards RD. Furthermore, achieving faster throughput results in fewer retransmissions, which significantly lowers transmitting nodes' transmission delays.

Each patient (SN_j) generates 3 data readings consisting of Heart Rate (HR), Blood Pressure (BP), and Pulse Rate (PR) using a random uniform distribution. SN_j will provide these values to RDs so that profiling can be done on them. RDs designate distinct profiles for smart nodes SNj based on these variables. Prioritizing traffic from profiles with critical HR, BP, and PR readings in the designated area is the major goal of this. In addition to all of this, the SN_js in Sigfox uses Ultra Narrow Band (UNB) modulation to send frames to the RDs.

Additionally, the system model developed for this manuscript is designated within a 5 km^2 radius in densely inhabited locations. The system architecture for the Sigfox environment smart health monitoring scenario is presented in Figure 1.

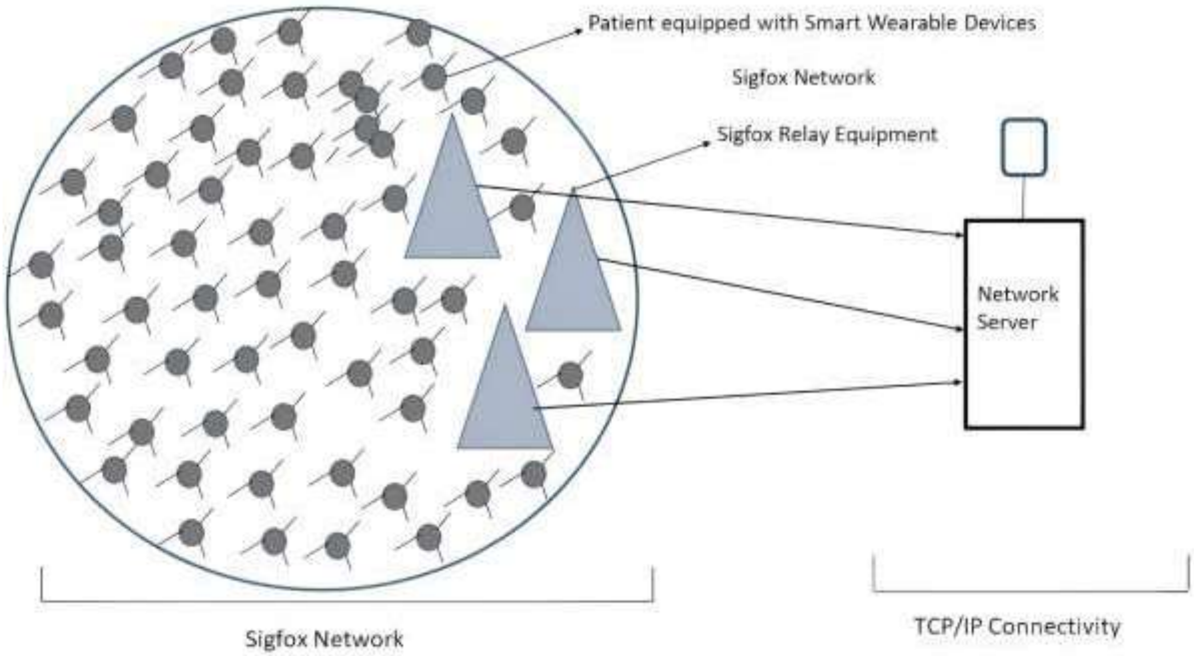

Figure 1 Sigfox System Architecture for Smart Health Systems.

This study's only goal is to use intelligent learning algorithms to reduce transmission delays in Sigfox, which will ultimately reduce collisions and retransmissions. When smart nodes transmit at the same time and same channel frequency, they collide in Sigfox [16], [26]. Sigfox is a single-hop wireless system that transmits frames from SNs to RDs using Pure Aloha. We utilize the Poisson distribution to express this Sigfox behavior, where Prob is the collided frames probability and is given as;

$$\mathrm{Prob} = e^{-2f}$$

where f is the number of frames transmitted per SN. Moreover, this probability will grow exponentially with the addition of SNs, causing QoS problems in the Sigfox network. One of the key components in improving QoS in Sigfox is the Latency factor.

The bit rate or frame bits transferred determines the transmission latency in a direct proportion [23]. As stated below, the delay in a Sigfox environment is dependent on bit rate and the quantity of bits in the broadcast frame.

$$\mathrm{Delay_{trans}} = \left({}^{Ps}/_{R}\right) + H$$

Where, P_s is the Payload size in bits, R is the rate of data measured in bits per second (bps), H is the overall overhead of Protocol in seconds. In case retransmissions are considered, the effective delay, or effective delay $\mathrm{Delay_{eff-trans}}$, becomes:

$$\mathrm{Delay_{eff-trans}} = N\,.\,\left({}^{Ps}/_{R} + H\right)$$

Where N is the number of retransmissions (typically Sigfox transmits each message three times for redundancy).

$$\mathrm{Delay}_{\mathrm{trans-prof}} = \sum_{z=1}^{k}(\mathrm{Delay}_{\mathrm{profile(1)}}, \mathrm{Delay}_{\mathrm{profile(2)}},\ldots\ldots, \mathrm{Delay}_{\mathrm{profile(k)}})$$

where k is the number of specific regions or profiles. Additionally, Z_j = Delay ($\mathrm{Delay}_{\mathrm{init}}$ + $\mathrm{Delay}_{\mathrm{Rc}}$ + $\mathrm{Delay}_{\mathrm{Rch}}$); $\mathrm{Delay}_{\mathrm{init}}$ is the initial delay of each node SN_j; $\mathrm{Delay}_{\mathrm{Rc}}$ is the retransmission delay produced by SN_j; and $\mathrm{Delay}_{\mathrm{Rch}}$ is normally introduced due to various impairments in the channel. Terminologies like beginning transmission, collision, and unfavourable channel circumstances must be understood by GW. There will be a back-off in transmission from SN_js in an SCPP until the transmission from SN_js in an CPP is finished.

The delay expression of SCPP becomes $\mathrm{Delay}_{\mathrm{Profi\_SCPP}}$.

$$\mathrm{Delay}_{\mathrm{Profi\_SCPP}} = \mathrm{Delay}_{\mathrm{Profi\_CPP}} + Z(j)$$

where j= 1,2, 3,….., continues to SN_j, $\mathrm{Delay}_{\mathrm{Profi\_SCPP}}$, and $\mathrm{Delay}_{\mathrm{Profi\_CPP}}$ represent the profiles delay.

Researchers from all around the world are turning to the GMM profiling approach as an alternative to traditional profiling methods. The GMM method uses the EM model to achieve convergence. The way that GMM's decision limits are shaped is an additional advantage. GMM generates more sort of elliptical profile shapes as compared to circular boundaries in K-Means. But one fascinating feature of GMM is that it gives each object a probability. We can rapidly ascertain the degree to which we think an SN matches a specific profile by giving each thing a likelihood. GMM is much more flexible and robust, in terms of profiles. Nevertheless, because the EM algorithm requires more iterations to reach convergence, GMM sometimes operates more slowly than K-Means. It is not the greatest course of action to let them reach a local minimum, which they might do very soon. Usually, K-Means is used to initialize GMM to prevent this issue. This generally improves the profiles (clusters) produced by K-Means well. KMeans enables the creation of GMM.

The following is a presentation of the generic Gaussian distribution as a Probability Distribution Function (PDF).

$$\mathrm{PDF}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

Patients that are lying in different locations depending on readings are represented by the probability density function $\mathrm{PDF}(x)$. The variance, standard deviation, and average (mean) of the bell-shaped curve are denoted by the words σ, σ^2, and μ, respectively.

Mathematically, for 3-dimensional data readings from patients (SNs), a vector is designed as, Y= $\{y_1, y_2, y_3, \ldots\ldots, y_n\} \in R^3$.

$$P(y/d) = \frac{1}{\sqrt{2\pi.\,|\sum d|}}.\, e^{-\frac{1}{2}(y_i-\mu_d)^T \sum_d^{-1} (y_i-\mu_d)}$$

$P(y/d)$ is the probability density function of i^{th} node with respect to center point of profiles c. After this in exponential component we are subtracting mean component from the i^{th} instance of EDs $(y_i - \mu_d)^T$ and in the middle we are multiplying it by inverse of co-variance $\sum_{d}^{-1}(y_i - \mu_d)$. The co-variance component describes the shape of Gaussian distribution.

In the proposed algorithm DSA, the decision of which SN is to transmit data is decided by RD. RD designs all these profiles with the help of probabilistic approach called GMM. Initially, all the SNs transmit data readings toward RDs, and these data readings are denoted by $G.SN_j$. Now GMM algorithm assign probabilities to all these SNs, according to the data readings received. Based on these probabilities, GMM assign SNs to specific profiles (CPP, SCPP, NPP) respectively. Further DSA schedule traffic from all these profiles like SN_j from CPP are allowed to transmit data for 15 minutes. After 15 minutes of data transmission, now SN_j from SCPP are allowed for 5 minutes. Once the first data readings are transmitted towards RD, now all the SNs use Enhanced Pure Aloha to transmit data. Enhanced Pure Aloha demonstrates that devices in CPP or SCPP can only transmit their 2nd reading, if and only if, the 1st reading is strayed from 2nd reading by 5 to 8 %. Following this rule, a lot of unwanted traffic will be differed and ultimately bandwidth will be efficiently utilized. The flow diagram of DSA with GMM is presented below.

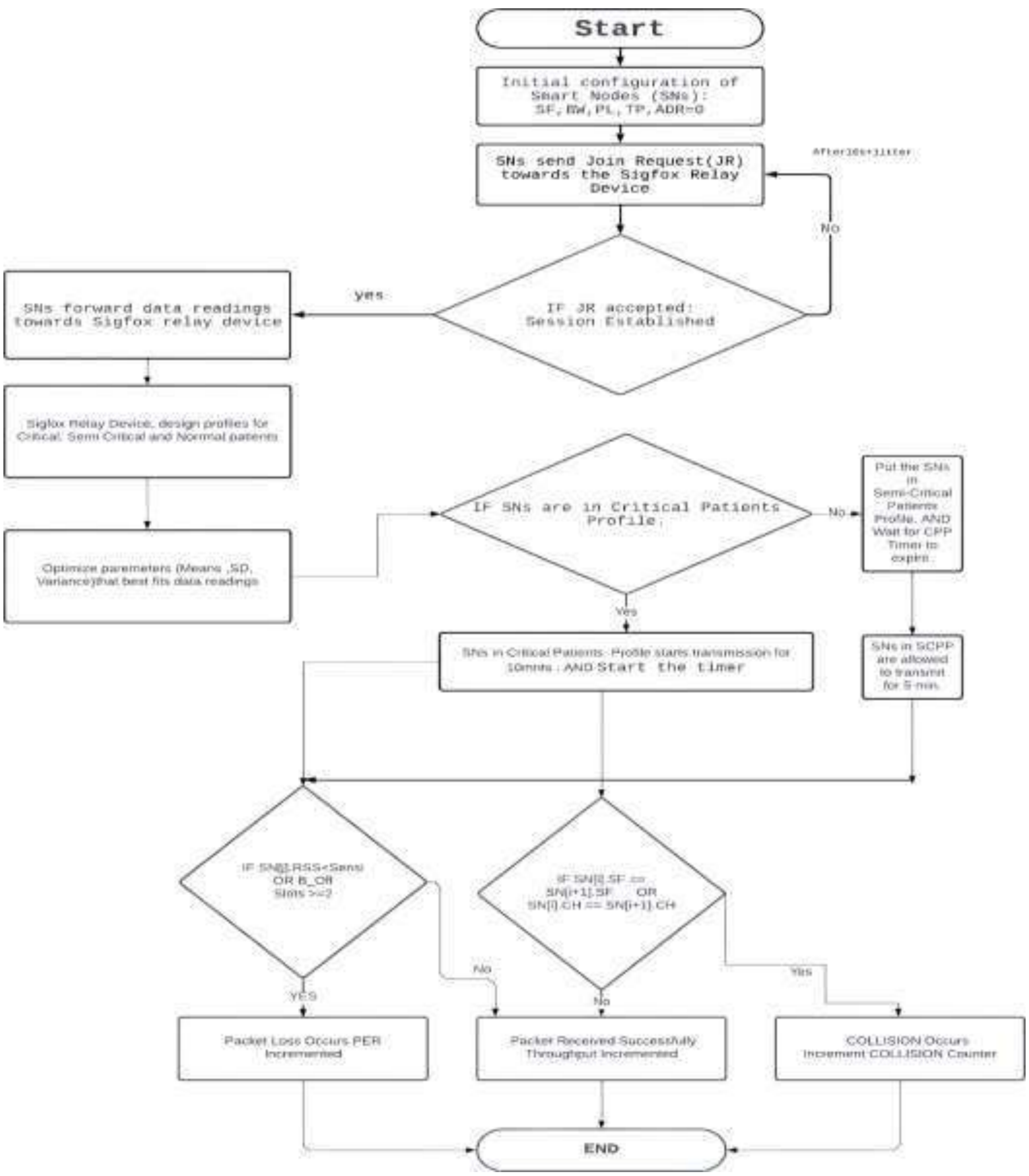

Figure 2 Flow diagram of DSA with GMM

The step-by-step algorithm of DSA with GMM approach is presented as follow.

Algorithm 1: Dynamic Scheduling Algorithm with Gaussian Mixture Model:

Initial Configuration: BW, T_p, CH.

if SN[j] send JR towards SERVER
SERVER accepted JR and assign Channel CH[i]
If Session == ESTB **Then**
SN[j] generates data frames towards RD
GMM of RD assign probabilities to SN[j].
According to assigned probabilities, SN[j] is assigned to Profiles [CPP, SCPP, NPP]
if $SN_j \in$ **CPP**
SN_j **:= $(SN_j)_{CPP}$ & $(SN_j)_{CPP}$** starts sending data for 15 Min
else if SNj $\in$ SC**PP & $(SN_j)_{CPP}$ = 0**
SN_j **:= $(SN_j)_{SCPP}$ & $(SN_j)_{SCPP}$** starts sending data for 5 Min.
if N_Reading changes by 5% **Than** Curr_Reading
OR
if Last_Ack Not Received
SNs in SCPP can transmit data several times during this 5 Min.
else if $SN_j \in$ **NPP & [$(SN_j)_{CPP}$, $(SN_j)_{SCPP}$] = 0 THEN**
SN_j **:= $(SN_j)_{NPP}$ & $(SN_j)_{NPP}$ &** ignore transmission.
else
$(SN_j)_{Pr}$ **= 0**
end if
PERFORM GMM with Dynamic Scheduling **AND goto** START LOOP.

V Results and Discussions

This section covers the simulation results for the FCR, FSR, and transmission delay. To determine the performance of Smart Node SN_i, in terms of collision, FSR, and FER, many simulations are performed, once the gateway or Relay Devices RD_j schedules the data generated from SN_i, among three mentioned profiles. Profiles like Critical Patients Profile (CPP), Semi-Critical Patients Profile (SCPP), and Normal Patients Profile (NPP) are discussed in detail earlier in previous section. Normalized FSR, FER, and collision values will be determined for each profile accordingly. Two RD_j were used in the simulation, producing two CPP concurrently. Since patients (SN_{is}) may also have critical readings, we should select the appropriate CPP for each RD_j independently. In a scenario where smart health monitoring is required, we want as many SN_i as possible to deliver data simultaneously while also considering the quality of service. The use of RD_{js} is a significant additional component in the previously described area. Given that we have two RD_j, the distances between them are carefully calibrated to minimize interference. The list of parameters used in the simulation is displayed in Table 1 [1] [4] [5].

Table 1: Initial Configurable Parameters for Smart Nodes (SN) or Patients

Required Parameters	Initial Configuration
Covered Geographical Area	5 - 6 Km^2
Bandwidth Used	125 KHz
Frequency	868 MHz

No. of patients	1000 (Which means 3000 data readings)
Tx pow.	2 - 17dBm
No. of Profiles	3 (CPP, SCPP, NPP)
Frame Size	20 Bytes

It is now allowed for the SNs in CPP to send frame packets to RD_j. The behavior of FSR is methodically examined with a range of nodes. The behavior of the FCR in percentage (%) with different numbers of SNs is shown in Figure 3. Each of the 1000 patients has three wearable smart devices that provide data to RD_j. SN_i first transmits data at BW 125 KHz and T_p 17 dBm. The payload size is limited to 20 bytes since SN_{is} generates a small amount of data in Kbits. The trend for traditional Sigfox is shown in Figure 3, which also shows an increase in FCR along with an increase in SN_{is}.

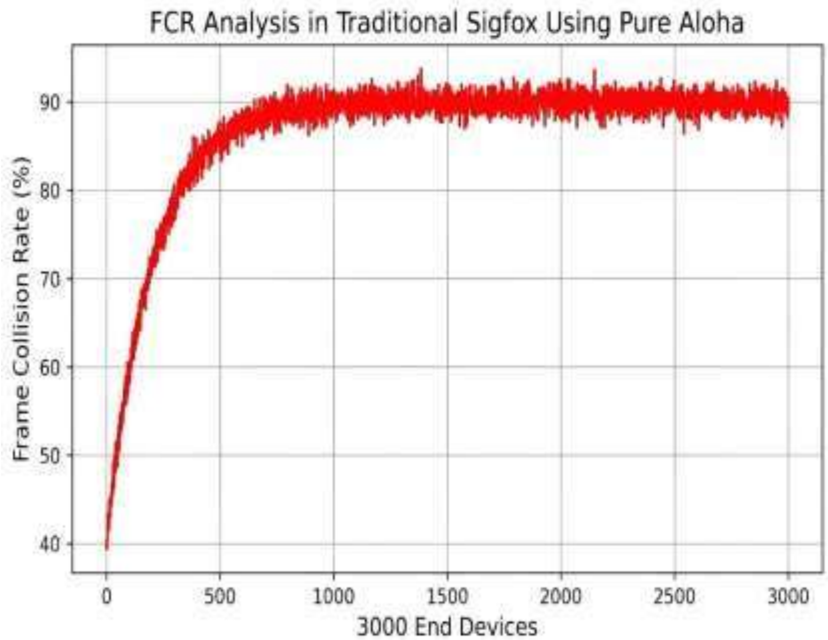

Figure 3 Analysis of FCR for Traditional Sigfox using Pure Aloha.

Findings in Figure 3 show that SN_i transmits data to RD_j via trailing Pure Aloha. It is evident that FCR is almost 85% when all SN_i transmit data simultaneously. Therefore, the goal of profile and real time scheduling on RD_j. is to lower the quantity of SN_i transmitted and, consequently, retransmissions. Enhanced Pure Aloha is carefully presented in Algorithm 1 below. By doing this, we can effectively control the network capacity and block superfluous traffic.

Results of the FCR for CPP through RD1 are shown below in Figure 4. Based on the probabilities assigned to SN_i, profiling is carried out using GMM. Following the completion of the simulation, the SN is distributed among the three profiles (CPP, SCPP, and NPP) by the GMM. 700 SN_i are initially included in CPP by GMM technique. Out of the entire 3000 SN_i, about 700 had critical readings. Profiles are prioritized by RD_j according to DSA. Once priorities have been determined, all SN_{is} in CPP are now allowed to send data to the given RD_j. The FCR behaviour with different numbers of SN_i is displayed in Figure 4. By restricting the quantity of SN_i, we can effortlessly transmit vital patient information to RD_j.

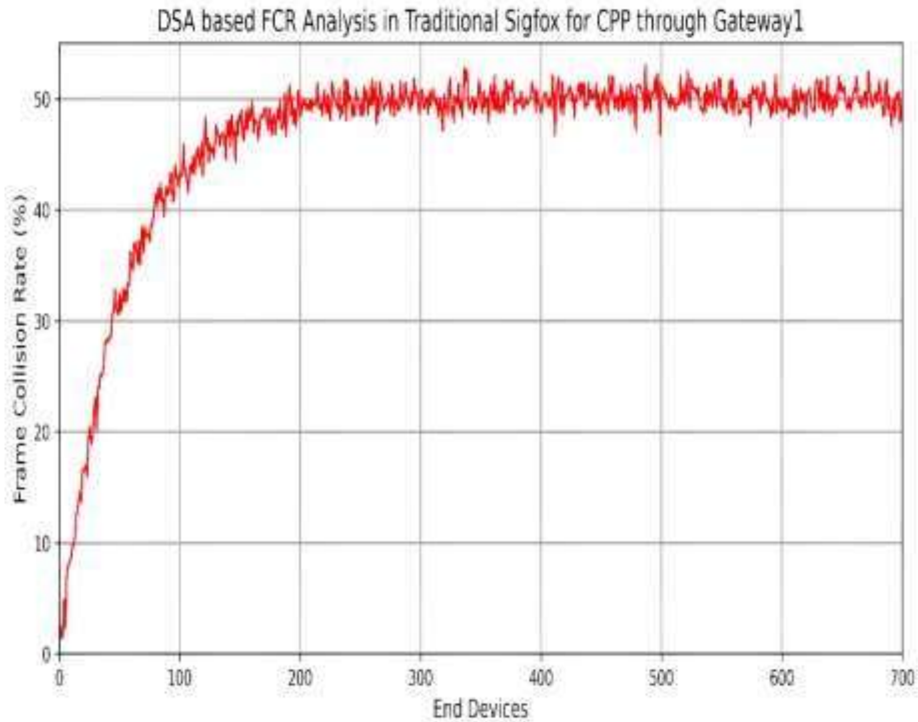

Figure 4 Analysis of FCR using DSA for Relay Device 1.

The findings of the second RD2, which serves 400 SNs (using the GMM technique to choose 400 SNs in CPP based on readings), are shown in Figure 5. Currently, RD2 is the primary destination for these 400 SNs frame transmissions. In comparison to CPP serviced by RD1, notably we have 300 less transmitting SN, the FCR ratio is slightly less in this CPP served by RD2 due to the decrease in the number of SNs. In the simulation, data for each of these patients is generated using the Gaussian or normal distribution. Gaussian distribution is used because critically ill patients tend to have lower blood pressure, pulse oximeter values, and heart rates in real-world settings. The peak's Probability Density Function (PDF), like that of the Gaussian distribution, indicates that the patients are normal. There are fewer of these patients on the right or left side who have critical levels.

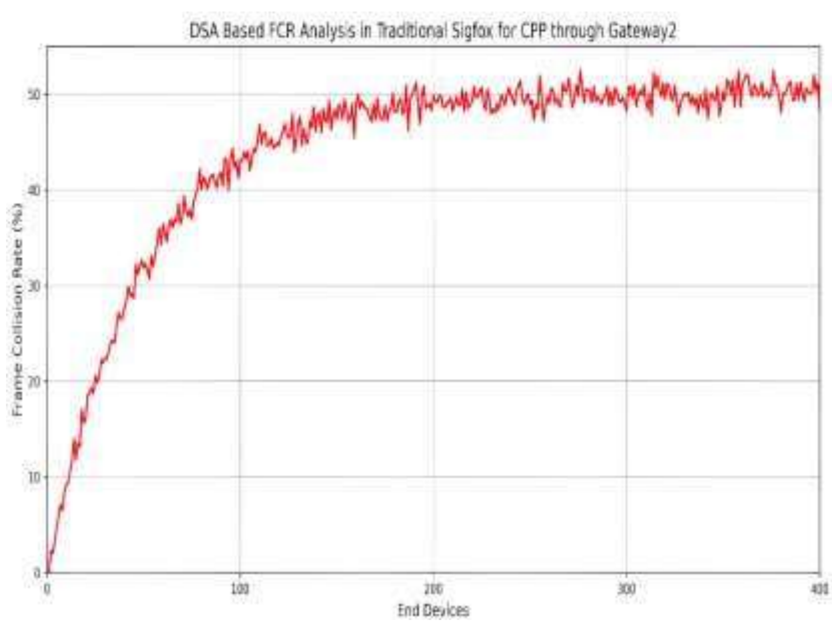

Figure 5 Analysis of FCR using DSA for Relay Device 2.

The behavior of FSR through gateway 1 or RD1 is depicted in Figure 6. Our goal is to maximise FSR, and this requires smart RD_j deployment. In terms of latency, DSA performs better than both traditional Sigfox. Retransmission only takes place in the event of a packet loss or collision. By lowering the number of SN_is, DSA cleverly reduces traffic so that more data can be transmitted at once. By using this method, the likelihood of a collision and subsequent frame retransmission is greatly decreased.

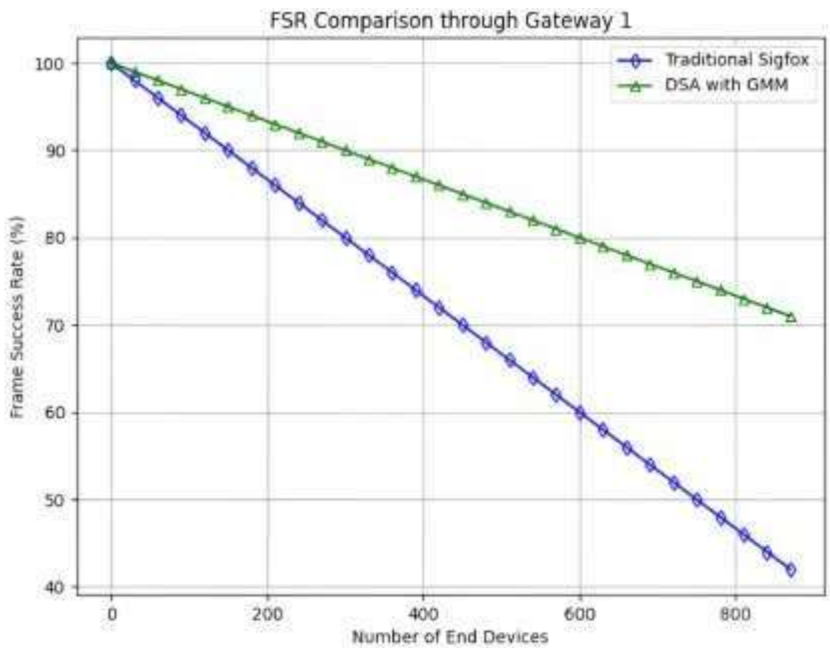

Figure 6 FSR comparison of Traditional Sigfox and DSA with GMM through Relay Device 1.

Figure 7 depicts the behavior of FSR through gateway 2 or RD2. The performance of DSA with GMM is much improved as compared to traditional Sigfox in terms of FSR.

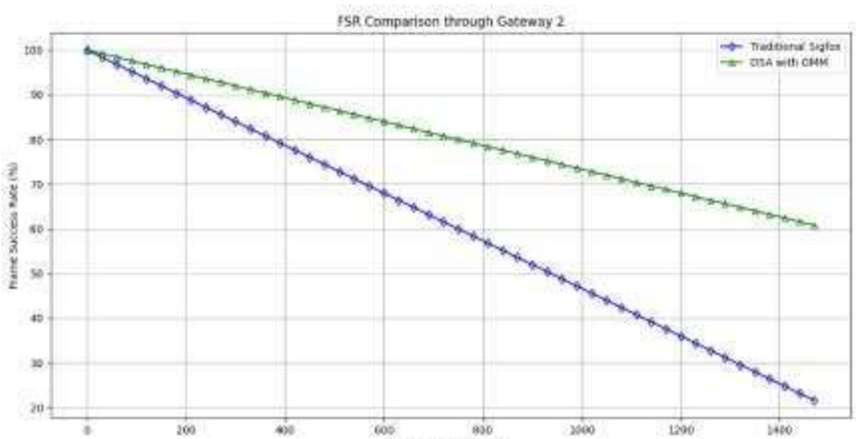

Figure 7 FSR comparison of Traditional Sigfox and DSA with GMM through Relay Device 2.

Figure 8 illustrates how different numbers of SNs might cause delays in the Sigfox network. In terms of latency, DSA performs better than conventional Sigfox. Using two RDjs, the outcomes of the DSA and conventional Sigfox networks are simulated. Retransmission only takes place in the event of a frame loss or collision. By lowering the number of SN_is, DSA cleverly reduces traffic so that more data can be transmitted at once. This method dramatically lowers the likelihood of a collision and, thus, frame retransmission. In terms of latency, DSA with GMM performs 73% better than the traditional Sigfox network.

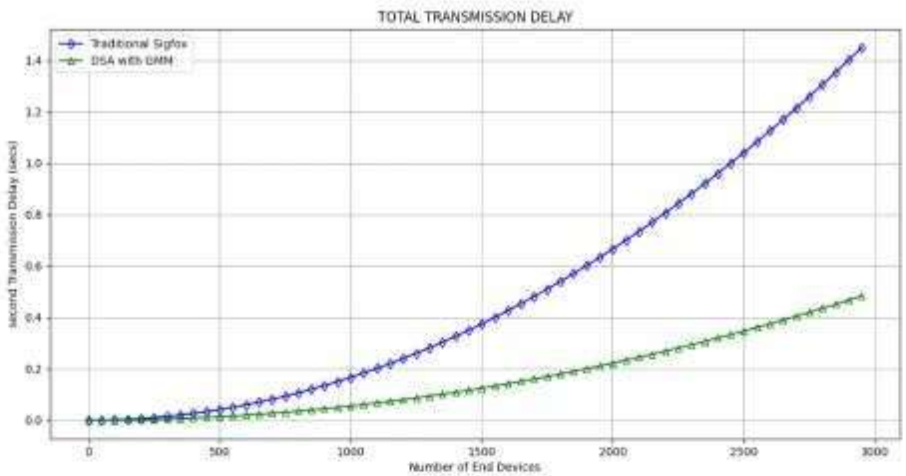

Figure 8 Comparison of delay performance between traditional Sigfox and DSA with GMM.

VI Conclusion

LPWAN is by far the most widely used choice for several smart IoT applications. In this article, we analyze the effect of delay in traditional Sigfox and optimize it's performance by using a novel DSA with GMM. The GMM is used to assign probabilities to SN and DSA technique schedules the traffic from SN. Utilizing GMM in conjunction with DSA lowers retransmissions and generates the ideal latency for Sigfox. Sigfox might be a good option because of its improved performance in terms of FSR, throughput, collision, and the quantity of retransmissions required. This is mainly because of profiles created for each SN, which reduces the number of simultaneous transmissions because the DSA is used to configure the SNs with different transmission intervals according to the priority of the profiles. In our investigation, compared to traditional Sigfox, the FCR is lowered by 39%. Additionally, FSR has increased by 39% when compared to traditional Sigfox. Moreover, the delay is simultaneously optimized by 79%. This study may be used in settings where it is necessary for patients' vital information to be transferred effectively and as soon as possible to gateways.

References

[1] J. B. Rydell, O. Otterlind, and I. Butun, "Delay considerations for reliable communications in lorawan," in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2022, pp. 1–6.
[2] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1472–1514, 2020.

[3] T. Adame Vázquez, S. Barrachina-Muñoz, B. Bellalta, and A. Bel, “Hare:Supporting efficient uplink multi-hop communications in self-organizing lpwans,” Sensors, vol. 18, no. 1, p. 115, 2018.
[4] J. M. Valtorta, A. Martino, F. Cuomo, and D. Garlisi, “A clustering approach for profiling lorawan iot devices,” in European conference on ambient intelligence. Springer, 2019, pp. 58–74.
[5] A. Farhad, D.-H. Kim, and J.-Y. Pyun, “R-arm: Retransmission-assisted resource management in lorawan for the internet of things,” IEEE Internet of Things Journal, vol. 9, no. 10, pp. 7347–7361, 2021.
[6] I. Bouazzi, M. Zaidi, M. Usman, M. Z. M. Shamim, V. K. Gunjan, and N. Singh, “Future trends for healthcare monitoring system in smart cities using lorawan-based wban,” Mobile Information Systems, vol. 2022, 2022.

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

[8] Isabona, J., & Imoize, A. L. (2021). Terrain-based adaption of propagation model loss parameters using non-linear square regression. Journal of Engineering and Applied Science, 68, 1-19..

[9] D. Magrin, M. Capuzzo, A. Zanella, L. Vangelista, and M. Zorzi, “Performance analysis of lorawan in industrial scenarios,” IEEE Transactions on Industrial Informatics, vol. 17, no. 9, pp. 6241–6250, 2020.

[10] Lopez, V., Kaushal, H., Prodanoff, Z., & Jiang, A. (2021). Evaluation of communication delays in LoRaWAN networks for indoor emergency scenarios. Journal of Communications Technology and Electronics, 66(Suppl 2), S149-S158.

Journal of Communications Technology and Electronics, vol. 66, no. 2, pp. S149–S158, 2021.

[11] Ikonen, J., Nelimarkka, N., Nardelli, P. H., Mattila, N., & Melgarejo, D. C. (2022). Experimental Evaluation of End-to-End Delay in a Sigfox Network. IEEE Networking Letters, 4(4), 194-198.

[12] B. Paul, “A novel mathematical model to evaluate the impact of packet retransmissions in lorawan,” IEEE Sensors Letters, vol. 4, no. 5, pp. 1–4, 2020.

[13] D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, “On the evaluation of application level delays in public lorawan networks,” in 2019 IEEE International Symposium on Measurements & Networking (M&N). IEEE, 2019, pp. 1–6.

[14] N. El Rachkidy, A. Guitton, and M. Kaneko, “Collision resolution protocol for delay and energy efficient lora networks,” IEEE Transactions on Green Communications and Networking, vol. 3, no. 2, pp. 535–551, 2019.

[15] Ali, Z., Qureshi, K. N., Mustafa, K., Bukhsh, R., Aslam, S., Mujlid, H., & Ghafoor, K. Z. (2022). Edge based priority-aware dynamic resource allocation for Internet of Things networks. Entropy, 24(11), 1607.

[16] D. Bankov, E. Khorov, and A. Lyakhov, “On the limits of lorawan channel access,” in 2016 International conference on engineering and telecommunication (EnT). IEEE, 2016, pp. 10–14.

[17] M. Alenezi, K. K. Chai, A. S. Alam, Y. Chen, and S. Jimaa, “Unsupervised learning clustering and dynamic transmission scheduling for efficient dense lorawan networks,” IEEE Access, vol. 8, pp. 191 495–191 509, 2020.

[18] S. Cui and I. Joe, “Collision prediction for a low power wide area network using deep learning methods,” Journal of Communications and Networks, vol. 22, no. 3, pp. 205–214, 2020.

[19] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, “Explora: Extending the performance of lora by suitable spreading factor allocations,” in 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2017, pp. 1–8.

[20] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Optimizing and updating lora communication parameters: A machine learning approach," IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 884–895, 2019.

[21] Ding, J., Nemati, M., Ranaweera, C., & Choi, J. (2020). IoT connectivity technologies and applications: A survey. IEEE Access, 8, 67646-67673.

[22] A. Mdhaffar, T. Chaari, K. Larbi, M. Jmaiel, and B. Freisleben, "Iotbased health monitoring via lorawan," in IEEE EUROCON 2017-17th international conference on smart technologies. IEEE, 2017, pp. 519–524.

[23] L. Alliance, "The lora alliance wide area networks for internet of things," www. lora-alliance. org, 2016.

[24] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in lora networks," arXiv preprint arXiv:1611.00688, 2016.

[25] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," ACM Transactions on Sensor Networks (TOSN), vol. 15, no. 2, pp. 1–35, 2019.

[26] Z. Ali, S. Henna, A. Akhunzada, M. Raza, and S. W. Kim, "Performance evaluation of lorawan for green internet of things," IEEE Access, vol. 7, pp. 164 102–164 112, 2019.