LoRa Mesh-5G Integrated Network for Trackside Smart Weather Monitoring

Yu Chen, *Student Member, IEEE,* Guo Shi, Mohammad Al-Quraan, *Graduate Student Member, IEEE,* Yusuf Sambo, *Senior Member, IEEE,* Oluwakayode Onireti, *Senior Member, IEEE,* and Muhammad Imran, *Fellow, IEEE*

***Abstract*—Monitoring of trackside weather is a critical aspect of railway operations, mainly for safety and efficiency reasons. Unfortunately, current cellular networks, including the fourth-generation and fifth-generation (5G) cellular networks, do not provide ubiquitous coverage for rail lines mainly due to an unfavorable cost-benefit realization. In this paper, we propose a Long Range (LoRa) mesh-5G integrated network that tackles this problem by utilizing a 5G network for backhaul, computing and storage, and LoRa mesh to extend coverage. We design a LoRa mesh server that runs on a private cloud of the 5G network to manage the LoRa mesh network. We integrate edge computing into the network and design a cloud-edge-terminal collaborative architecture with three algorithms for timely significant-change updates, packet loss detection, and adaptive thresholds to reduce the packet rate and data volume of the network. We validate the design by implementing a proof-of-concept on the 5G testbed at the University of Glasgow. The experimental results demonstrate the feasibility of the network and the cloud-edge-terminal collaborative architecture.**

***Index Terms*—LoRa mesh, 5G, hybrid network, weather monitoring, railway digitalization, cloud-edge-terminal collaboration.**

I. Introduction

Since the 19th century, the railway has become one of the main transportation methods all over the world. According to the International Union of Railways [1], the total length of railway tracks in the world was more than 855,726 km in 2022. The cost of maintaining such a vast infrastructure is very high, especially if traditional periodic maintenance methods are used, i.e., sending the crew and equipment to walk along railways to check the status of the infrastructure regularly. With the development of Internet of Things (IoT) and data analysis technologies, predictive maintenance has become a promising method to reduce costs. Due to the long length of railway tracks, a massive number of sensors deployed alongside railway tracks is needed to monitor the status of the infrastructure and transmit it to the cloud for making policies. The crew and equipment are sent out to specific locations only when maintenance status is triggered by the system. On the other hand, bad weather causes damage to the infrastructure and poses a threat to the safety of passengers and staff [2]. Specifically, a high temperature poses the risk of buckling to rails and a low temperature freezes rails. High wind can blow objects onto rails blocking the railway, while heavy rain makes rails slippery and continuous rain would result in landslides or flooding. To reduce the damage and threat, railway operators need to monitor the trackside weather and predict the weather in the future. With the information, they can take action before adverse weather happens, such as reducing the speed limit, rescheduling routes, and sending special fleets to maintain the rails. Both predictive maintenance and weather monitoring need a massive machine-type communication (mMTC) network to gather data from sensors throughout the whole railway network. The main requirements of the network are as follows.

1) Wide coverage: The communication network should cover the whole railway network which is characterized by several long linear tracks.
2) Supporting massive end devices: Given the long length of railway tracks, a massive number of sensors and end devices need to be deployed to monitor the whole railway network.
3) Low-cost communication infrastructure: High investment in the infrastructure is hard to be paid back as many railways are located at remote sites without other connectivity requirements.
4) Low-power and low-cost end devices: Given the massive number of end devices, their power consumption and cost have a great impact on the overall cost.

Unfortunately, current trackside networks do not satisfy the requirements. As a traditional mobile communication system of railways, the global system for mobile communications-railway (GSM-R) is widely used globally, especially in Europe [3]. Based on the second-generation cellular network, GSM-R is already outdated and suffers from many issues such as high interference, low capacity, and limited capability. Hence, GSM-R cannot support massive end devices for predictive maintenance or weather monitoring. With the sunset of the second and third-generation cellular networks, GSM-R is being replaced by long-term evolution for railways (LTE-R) which is based on the fourth-generation (4G) cellular network.

Manuscript received 26 June 2023; revised 22 January 2024; accepted 29 January 2024 (Corresponding author: Guo Shi).

Yu Chen, Mohammad Al-Quraan, Yusuf Sambo, Oluwakayode Onireti, and Muhammad Imran are with the James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK (e-mail: {y.chen.5, m.alquraan.1}@research.gla.ac.uk; {yusuf.sambo, oluwakayode.onireti, muhammad.imran}@glasgow.ac.uk).

Guo Shi is with the Department of Management Science, University of Strathclyde, Glasgow, G4 0QU, UK (e-mail: guo.shi@strath.ac.uk).

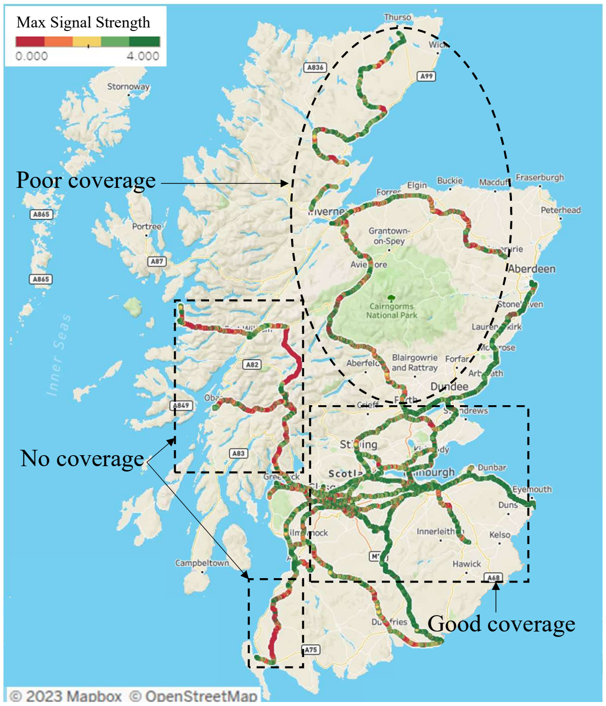


Fig. 1. 4G signal strength of Scotland railways (based on [5]).

Although operators have started deploying 4G since 2009, they do not cover the whole railway network due to the low rate of return in remote and rural areas. Taking Scotland as an example, Ofcom, the regulator for communication services in the United Kingdom, measured the 4G signal strength in the railways of Scotland [4]. As shown in Fig. 1, the railways near main cities have good signal strength, whereas coverage in rural areas is poor, especially the lines in the highland and west coast which are remote sites with low population density. Thus, the current 4G does not provide the coverage required by railway networks. With wide deployment in recent years, the fifth-generation (5G) cellular network is expected to facilitate railway digitalization. However, 5G faces the same challenges as 4G, and might even require a denser deployment based on their operating frequency bands. Therefore, the biggest issue of 4G and 5G for widespread railways is that they cannot provide wide coverage at a low cost.

Regarded as one of the most popular low-power wide-area technologies, long range wide area network (LoRaWAN), based on long range (LoRa), is an important supplement to cellular networks in rural areas [6]–[9]. With a star topology, the messages from LoRaWAN end devices aggregate at the LoRaWAN gateway which forwards them to the LoRaWAN server on the cloud by backhaul networks. Although LoRaWAN has wide coverage, it is not suitable for linear rail lines as many LoRaWAN gateways have to be deployed alongside the lines at a certain distance. Each gateway requires access to a backhaul connection, which significantly increases the deployment cost. Instead of using one-hop communication, LoRa mesh can extend the coverage of LoRaWAN with no need to densely deploy gateways. It allows trackside end devices to relay messages from other end devices until they arrive at a gateway that is deployed at locations with access to a backhaul network. Thus, LoRa mesh is a potential network providing wide coverage for railways.

Motivated by the trackside sensing requirements, the coverage status of existing trackside networks, and the wide coverage of LoRa mesh, in this paper, we propose a LoRa mesh-based network to monitor trackside weather. LoRa mesh is a self-organized network lacking a centralized server for management, e.g., node registration. To overcome this challenge, we propose integrating LoRa mesh into a 5G network, which provides the LoRa mesh with a private cloud for network management and intelligent data processing. Moreover, 5G networks can provide a reliable and flexible backhaul for LoRa mesh and reduce the deployment cost by utilizing existing cellular infrastructure. On the other hand, collaborating with the terminal and cloud, edge computing is integrated to reduce the volume of data transmitted in the network. The proposed network is suitable for trackside low data rate sensing applications including both predictive maintenance and weather monitoring. Moreover, due to its low cost and extensive coverage, the proposed network can be utilized in various scenarios, including infrastructure monitoring in remote areas, environmental monitoring in agricultural settings, and the development of IoT-based smart cities. In this paper, we implement it as a proof of concept for weather monitoring. Our main contributions in this paper are summarized as follows.

1) We integrate LoRa mesh into a 5G network. The 5G network provides a reliable backhaul and a centralized platform for deploying LoRa mesh servers. LoRa mesh significantly extends the coverage of the 5G network at a low cost.
2) We design a cloud-edge-terminal collaborative architecture for the LoRa mesh-5G integrated network. With timely significant-change updates, packet loss detection, and adaptive threshold algorithms, the architecture can significantly reduce the data volume of the network.
3) Using off-the-shelf components, we implement the LoRa mesh-5G integrated network at the University of Glasgow campus, proving its feasibility. Although it is used for weather monitoring, it acts as a prototype for trackside low data rate sensing.

The rest of this paper is structured as follows. We review the related works in Section II. In Section III, we present the network model and integration methods of the LoRa mesh-5G integrated network. The cloud-edge-terminal collaborative architecture is proposed in Section IV along with three algorithms about timely significant-change updates, packet loss detection, and adaptive thresholds. The implementation and experimental results are presented in Section V and Section VI, respectively. Finally, Section VII concludes the paper.

II. Related Works

There has been significant research progress on facilitating the evolution of GSM-R to IoT-enabled smart systems for railways. Paula *et al.* [3] reviewed the evolution of railway communication technologies from GSM-R to LTE-R. They observed that the legacy infrastructure is gradually being replaced by smart train management systems that provide many IoT-enabled services such as predictive maintenance, smart train control, and smart infrastructure. Singh *et al.* [10] investigated key IoT technologies that have the potential to be applied to the next-generation railway system such as fog computing, cloud computing, wireless sensor networks, big data analytics, and 5G technology. As current trends, IoT and big data-based methods are being used to collect and analyze data from railway systems. Moreover, artificial intelligence-integrated architectures are being used to make policies for smart railway systems. Ai *et al.* [11] presented a complete framework of 5G technologies for smart railways. However, due to the high cost of dense deployment of 5G base stations, current efforts to employ 5G technology in smart railways, such as [12]–[15], only focus on high-speed or urban railways.

With wide coverage and low power consumption, LoRa-based networks are being used for smart railways. Jo *et al.* [16] stated the necessity of combining IoT with condition-based maintenance for smart railways. To evaluate the suitability of existing IoT solutions, they compared long-term evolution (LTE), LoRa, and narrowband Internet of Things (NB-IoT). According to the results, LoRa is the optimal solution in terms of coverage, power consumption, and implementation complexity. As for electromagnetic compatibility, Deniau *et al.* [17], [18] analyzed the impact of transient electromagnetic interference on LoRa and LoRaWAN in a railway environment and proved that the interference can be detected and separated by support vector machines (SVM)-based methods. Based on LoRa, Naveen and Rayala [19] designed an automatic level-crossing system, and Ferretti *et al.* [20] designed a signaling system for smart railways.

To extend the coverage of LoRa, many efforts have been put into research on LoRa mesh networks. Lundell *et al.* [21] proposed a routing protocol for LoRa mesh networks and validated it in both laboratory and field tests. Berto *et al.* [22] implemented a LoRa mesh network based on RadioHead packet radio library [23] which is a popular open-source LoRa mesh library for embedded microprocessors with very limited resources. Lee and Ke [24] evaluated the performance of LoRa mesh networks via a real experiment of monitoring large-area IoT sensors. Huh and Kim [25] proposed a LoRa mesh protocol and discussed its use cases including fire pipe freeze monitoring, street light smart control, and toxic gas monitoring. Ebi *et al.* [26] used a LoRa mesh network to monitor underground infrastructure. By evaluating the performance of two field tests, it is proved that LoRa mesh networks have advantages over LoRa and LoRaWAN in terms of the coverage and reliability of packet delivery. Hong *et al.* [27] proposed a hierarchical-based energy-efficient routing protocol for LoRa mesh networks. It is demonstrated that the proposed protocol outperforms conventional ad hoc on-demand distance vector routing methods in terms of energy efficiency and transmission delay. Tian *et al.* [28] developed LoRaHop which is an add-on protocol compatible with LoRaWAN to extend the coverage by a multi-hop mesh network. They evaluated its performance on an outdoor testbed demonstrating that LoRaHop can extend the coverage of a LoRaWAN network with improved reliability and reduced power consumption. Wu and Liebeherr [29] proposed a self-organizing communication protocol, called CottonCandy, to mitigate packet collisions during data collection.

To reduce the deployment cost of LoRa-based networks and extend the coverage of 5G networks, LoRa/LoRaWAN-5G integration is regarded as a potential solution. Chen *et al.* [6] provided a survey on LoRa/LoRaWAN-5G integration in which the main challenges and potential solutions are discussed in detail. Yasmin *et al.* [7] proposed four integration methods. Three of them focus on enabling LoRaWAN to communicate with the 5G access network. The last one aims to deploy LoRaWAN servers as a part of the 5G core network. Navarro [8] proposed to incorporate the protocol stack of eNodeB into the LoRaWAN gateway, enabling the LoRaWAN gateway to access 4G/5G core network directly. Ksentini and Frangoudis [30] utilized the European Telecommunications Standards Institute (ETSI) multi-access edge computing server (MEC) [31] as the cloud for deploying LoRaWAN servers and related applications. To improve the performance of LoRaWAN-5G integrated networks, Torroglosa *et al.* [32] proposed a roaming method for the end devices with dual connectivity of LoRaWAN and 5G. Because of these efforts, LoRa/LoRaWAN-5G integrated network has been used for many applications. Chen *et al.* [33] implemented a LoRaWAN-5G integrated network with a cooperative access network and a converged core network, which is used to monitor the indoor temperature and human activity for smart buildings. Zhang *et al.* [34] integrated LoRa into the 5G network to extend coverage to the blind area of smart grids.

While 5G networks provide cloud computing and centralized management for LoRa-based networks, edge computing is being used to enhance their network performance. Sarker *et al.* [35] proposed a generic architecture to integrate edge computing into LoRa-based networks. Reducing the latency, edge computing makes LoRa-based networks suitable for some latency-sensitive applications. Liu *et al.* [36] designed and implemented a LoRa system with edge computing at the LoRa gateway to reduce the large latency of LoRa and balance the workloads between the LoRa gateway and the LoRa servers on the cloud. Sarker *et al.* [37] utilized an edge computing-enabled LoRa network for smart parking, achieving near real-time monitoring of vehicles. Reducing the volume of data sent to the cloud, edge computing eases the backhaul workload. Kumari *et al.* [38] implemented edge devices to compress the data from smart meters before sending it to the LoRa gateway. Sharofidinov *et al.* [39] deployed a machine learning approach at the edge of a LoRa network to analyze and control the state of a greenhouse, reducing the volume of data sent to cloud servers. In addition to reduced latency and data volume, edge computing can enhance network security. Hou *et al.* [40] used edge computing to enhance the security of LoRa by deploying blockchain at the LoRa gateway, i.e., the edge of the network.

To summarize, IoT technologies are improving railway digitalization and LoRa is a promising solution. However, as a single technology, LoRa is not sufficient for widespread railway systems. Integrating mesh networks, 5G networks, and edge computing into LoRa significantly improves its performance and capability. Therefore, in our approach, we

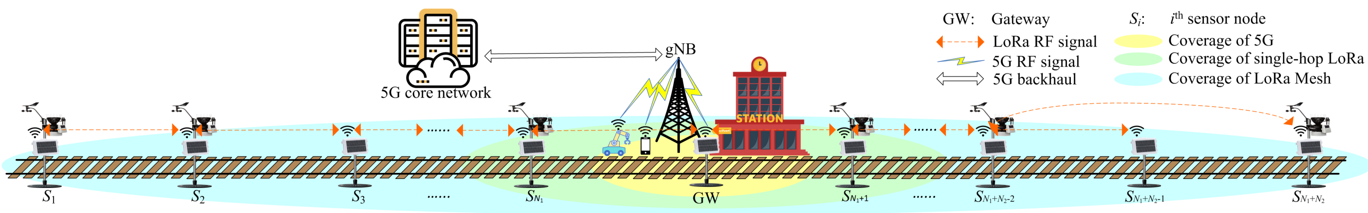


Fig. 2. System model of the LoRa mesh-5G integrated network.

design and implement a LoRa mesh-5G integrated network enabled by edge computing.

III. LoRa Mesh-5G Integrated Network

In this section, we will introduce a system model and an integration approach of the LoRa mesh-5G integrated network for trackside smart monitoring.

A. System Model

As shown in Fig. 2, a LoRa mesh network is deployed alongside a railway track to monitor the weather. It consists of two kinds of nodes, i.e., gateway (GW) and sensor nodes denoted as S_i where i is an integer. The gateway is deployed in the middle of the railway line, receiving packets from sensors deployed on either side of the railway line. The coverage area of the gateway is equivalent to that of a single-hop LoRa network, depicted as a light green ellipse in Fig. 2. We assume that there are N_1 and N_2 sensor nodes on the left side and the right side of the gateway, respectively, where N_1 and N_2 are integers. Some nodes, such as $S_1, S_2, S_{N_1}, S_{N_1+1}, S_{N_1+N_2-2}$ and $S_{N_1+N_2}$, are equipped with weather stations. They can read the weather station data and send it to the gateway in the form of LoRa mesh packets. Besides, they also relay packets from other nodes. Other nodes do not have weather stations, such as S_3 and $S_{N_1+N_2-1}$, and only relay packets. They are used to extend the coverage of the LoRa mesh network when network operators do not want to densely deploy weather stations. The relaying of sensor nodes significantly extends the coverage of the LoRa mesh network, as depicted by the light blue ellipse in Fig. 2. All the packets containing weather data are transmitted to the gateway via one-hop or multi-hop LoRa communication. Depending on the distances between nodes, one sensor node can have one or multiple routes to the gateway. When joining the network or finding that the existing route is no longer valid, a sensor node tries to discover a valid route to the gateway automatically, achieving self-organization and self-healing [22], [41]. Moreover, to achieve remote and smart control, downlink communication is enabled, allowing the gateway to send commands like resetting to sensor nodes.

After aggregating at the gateway, all the trackside data needs to be transmitted to the cloud through a backhaul connection. To leverage existing cellular infrastructure, the gateway is deployed in areas with 5G coverage, e.g., a train station. As shown in Fig. 2, a 5G base station (gNB) is deployed near a train station providing connections for various users, such as automated guided vehicles, smartphones, and the LoRa mesh network. Compared with LoRa and LoRa mesh, the 5G network offers relatively shorter coverage, illustrated by the yellow ellipse in Fig. 2. In the 5G network, the gateway acts as user equipment (UE) that can communicate with the gNB directly. Then, through the 5G backhaul, the weather data can securely arrive at the 5G core network in the cloud. It is clear from Fig. 2 that the LoRa mesh-5G integrated network has a significantly extended coverage compared with 5G networks and single hop LoRa-5G integrated networks.

B. Integration Approach

Although LoRa mesh networks benefit from flexible deployment and coverage extension, they lack a centralized server, resulting in difficulties in management. Leveraging the existing 5G network, we propose to deploy a LoRa mesh server within the 5G core network to provide efficient management. To allow the LoRa mesh nodes to access the 5G network, we integrate a 5G dongle into the LoRa gateway.

The functions of the LoRa mesh server are 1) sensor node registration, 2) network parameter management, e.g., the update frequency of sensor nodes, 3) data storage, 4) cloud data processing for application, and 5) providing a user interface. As shown in Fig. 3, the LoRa mesh server is deployed in one of the virtual machines within the 5G network cloud. Based on the 5G-MEC integration method proposed by ETSI [31], the cloud is deployed within the 5G core network as a MEC and connects to the user plane function (UPF) via N6 interface in the user plane. Based on hypertext transfer protocol secure (HTTPS), transmission control protocol (TCP), and Internet Protocol (IP), the user interface bridges the LoRa mesh server and external networks. Through the user interface, the network status and the processed application data are displayed to users in a dashboard. Moreover, the interface allows authorized users to change the parameters of the network and reset specific sensor nodes or the gateway. To enable the gateway to access the LoRa mesh server within the 5G core network, we install the protocol stack of 5G UE in the gateway. As shown in Fig. 3, the gateway has both the LoRa mesh protocol stack and the 5G UE protocol stack. When receiving LoRa mesh packets from sensor nodes, the gateway decapsulates them like a usual LoRa mesh node. Then, acting as a 5G UE, the gateway encapsulates the data as 5G packets and transmits them to gNB over the air via NR-Uu interface. Through the gNB and UPF, the packets arrive at the LoRa mesh server securely as the transmission is protected by 5G security mechanisms.

In terms of the application layer protocol for the session between the gateway and the LoRa mesh server, we choose message queuing telemetry transport (MQTT) [42] which

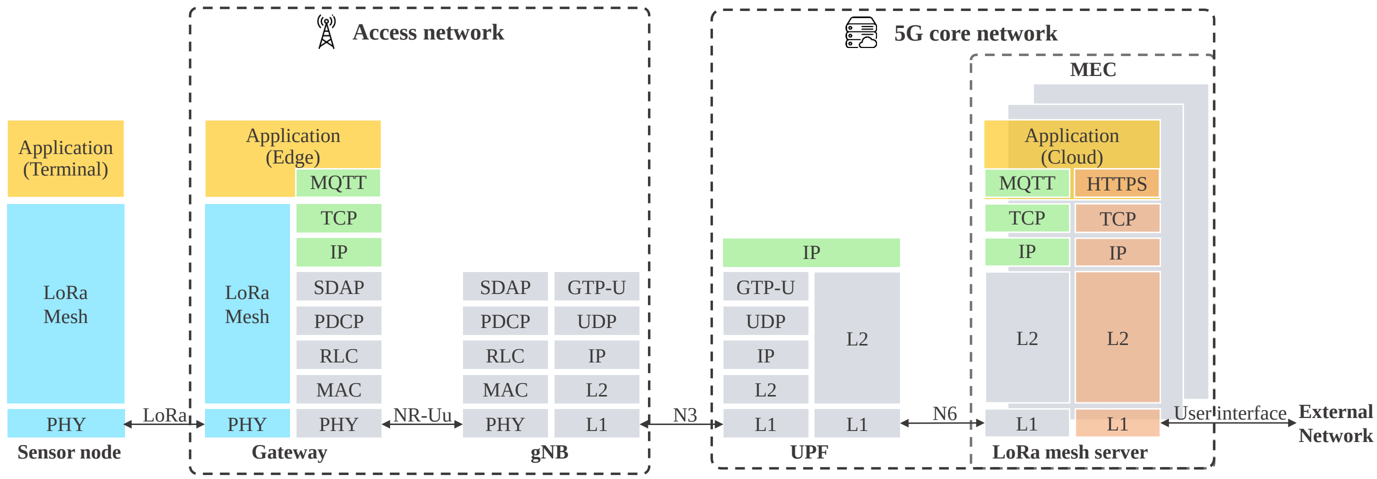


Fig. 3. Protocol stacks of LoRa mesh-5G integrated network.

is a lightweight open messaging transport protocol. Based on a publish-subscribe model, an MQTT broker listens to MQTT publishers and retransmits the received messages to specific MQTT subscribers. The messages are published on topics. In the LoRa mesh-5G integrated network, an MQTT broker is deployed in the LoRa mesh server to retransmit messages with two topics including uplink data and downlink commands. Regarding uplink data, an MQTT publisher and an MQTT subscriber are deployed in the gateway and the LoRa mesh server, respectively. Regarding downlink commands, an MQTT publisher and an MQTT subscriber are deployed in the LoRa mesh server and the gateway, respectively. By doing so, bidirectional communication between the gateway and the LoRa mesh server is achieved. Designed for restricted environments such as machine-to-machine communication and IoT, MQTT has many advantages including a small footprint, limited network bandwidth, fast message delivery, and easy deployment. In the LoRa mesh-5G integrated network, compared with hypertext transfer protocol (HTTP) [43], MQTT can reduce the consumption of the gateway resource, especially the required 5G radio frequency bandwidth. However, security is the main drawback of MQTT [44]. Without built-in encryption, MQTT usually uses transport layer security (TLS)/secure sockets layer (SSL) for security encryption, deviating from its design aim as TLS/SSL is not a lightweight protocol. By virtue of the 5G security mechanisms, there is no need to use TLS/SSL as MQTT data is encapsulated as a 5G payload protected by 5G authentication and encryption. Thus, the drawback of MQTT on security is overcome in the LoRa mesh-5G integrated network.

According to the above description, the integration of LoRa mesh and 5G is achieved at both the core network level and access network level. The core-level integration helps manage the LoRa mesh network and provides a user interface for observation and control. The access-level integration keeps the deployment flexibility of the LoRa mesh network. Collecting data from a mesh network requires one or more nodes to have a backhaul connection to upload the data to the cloud. The backhaul connection is usually via wire-based networks like Ethernet or short-range wireless networks like WiFi, posing restrictions on the deployment location of one or more nodes. With the capability of communicating with gNB, the gateway is required to be deployed within the coverage of gNB which is a relatively large area and has no impact on the deployment flexibility of LoRa mesh networks. Thus, the LoRa mesh-5G integrated network benefits from both the management capability of the 5G network and the coverage extension of LoRa mesh networks with no impact on the deployment flexibility.

IV. Cloud-edge-terminal Collaboration

In addition to enhancing communication capability, the integration of LoRa mesh and 5G also provides a three-level computing architecture, i.e., terminal computing at sensor nodes, edge computing at the gateway, and cloud computing at the 5G core network. Through cloud-edge-terminal collaboration, the packet rate and data volume of the integrated network can be reduced significantly, which is beneficial to enhancing the scalability of the LoRa mesh network, reducing the required 5G radio frequency bandwidth, and easing the backhaul workload.

A. *Cloud-edge-terminal Collaboration Architecture*

As shown in Fig. 4, terminal devices, i.e., sensor nodes, measure air temperature and wind speed, and the observations are sent to the edge, i.e., the gateway, periodically. Despite operating in license-free bands, LoRa suffers from duty cycle limitation which results in lower packet rates to connect more end devices [45]. So, it is necessary to reduce the frequency of periodic updates to support more sensor nodes. However, a low update frequency increases the delay between a significant change in an observation happening and the railway track manager knowing it. The delay further increases the response time to adverse weather or extreme weather. To address the issue, besides sending periodic updates at a low frequency, terminal devices send significant-change updates to the gateway edge immediately when detecting a significant

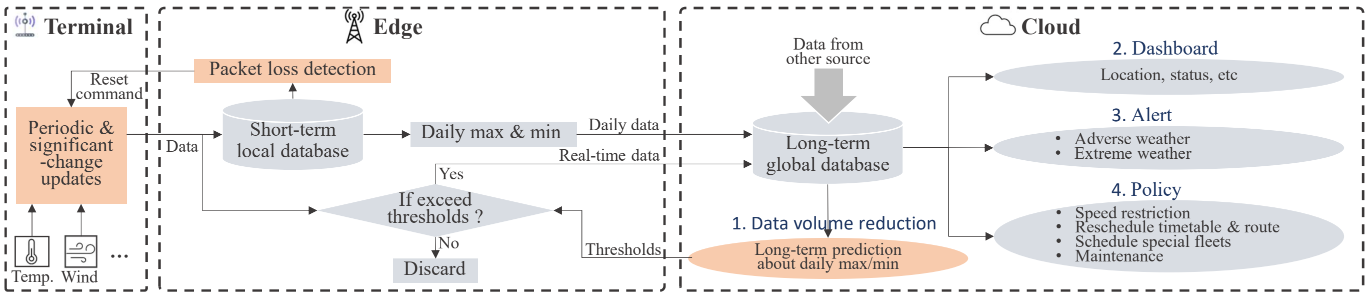


Fig. 4. Data flow chart of cloud-edge-terminal collaboration. "Temp." denotes air temperature.

change between the current observation and the last update. The detection is easy to achieve so that the complexity and resource consumption of terminal devices is not increased. As shown in Fig. 4, after receiving data from terminal devices, the gateway edge stores it on a local database. Given the storage capability of the gateway edge device, the retention time of the data is limited.

Operating in license-free bands, LoRa suffers from packet loss due to reasons such as signal collision [46]. With more hops and dynamic routes, LoRa mesh is more likely to lose packets. If packet loss continues for a specific sensor node, there is a high probability that the sensor node has malfunctioned or its route to the gateway has issues, e.g., some intermediate nodes are too busy to relay packets for it. Routing-related issues can be solved by remote resetting the sensor node as this will initiate route discovery again. If there are multiple potential routes to the gateway, the new route is very likely to be different from the old one as the busy intermediate nodes are also too busy to process the route discovery request from it. The issues with the sensor node can be classified into two categories. The first category is those that can be solved remotely, e.g., the periodical updates of two sensor nodes collide with each other as they transmit periodical updates at the same time due to the same period and start time. Remote resetting can solve the issue by changing the start time of the sensor node. It is also expected to solve many other issues of the first category. The second category is those that cannot be solved remotely such as hardware issues. Although they cannot be solved remotely, a remote resetting command is also helpful to verify this kind of issue as the sensor node cannot transmit an acknowledgment back to the gateway for the remote resetting command in this situation. Thus, it is necessary to design a packet loss detection algorithm to trigger remote reset commands to heal the network or verify the issues that cannot be solved remotely. In this paper, as shown in Fig. 4, a method of packet loss detection is designed for the gateway edge. It reads data from the local database and automatically sends a reset command to the specific sensor node when detecting continuous packet loss. On receiving a reset command, the sensor node resets itself, discovers a route to the gateway again, and sends an acknowledgment to the gateway. The reason why packet loss detection is deployed at the gateway edge instead of the cloud is that it is based on the most recently received data that is not necessarily fully sent to the cloud. To reduce the backhaul 5G network requirement, the edge filters data before sending it to the cloud. The filtering policy is based on thresholds calculated in the cloud. Besides filtering the real-time data, the edge also calculates the maximum air temperature, minimum air temperature, and maximum wind speed of a day, which are sent to the cloud on a daily basis for calculating the thresholds of the filtering.

The computation tasks of the cloud are application-oriented. In the case of trackside weather monitoring, track and/or train operators are concerned about bad weather that is likely to pose a danger to passengers, trains, or infrastructures. Network Rail [2], the biggest track manager in the United Kingdom, discloses their definitions of adverse weather and extreme weather in winter, i.e., temperature below -3 °C or wind speed above 60 miles per hour (mph) as adverse weather, and temperature below -7 °C or wind speed above 70 mph as extreme weather. In summer, high temperature also has an adverse impact on the track or train. In adverse weather or extreme weather, the operator must take action promptly such as imposing a speed restriction for trains, rescheduling timetables and/or routes, and scheduling special fleets like snowploughs. Thus, we propose to use historical daily maximum and/or minimum weather values to predict the maximum and/or minimum weather values of the next day as the thresholds of real-time data filtering. As shown in Fig. 4, the thresholds are sent to the edge. If weather values are within the thresholds, they are discarded in the edge. Otherwise, they are sent to the cloud immediately. There are two advantages of the adaptive thresholds. First, the operator can master the trend of the weather with reduced data volume from the edge to the cloud. Second, if weather values are outside of the predictions, the operator is informed of the unexpected values immediately which are likely to change to adverse or extreme weather. So, the operator can get an early warning of adverse weather and extreme weather. By employing the adaptive thresholds, the operator obtains the data with lower delays. In the meantime, the data volume from the edge to the cloud is significantly reduced.

As shown in Fig. 4, besides data volume reduction, the cloud runs a dashboard that shows the location of LoRa mesh nodes, the status of the LoRa mesh network, and the weather status and values. Moreover, the cloud sends alerts to corresponding staff when adverse or extreme weather is likely to happen. Combined with data from other sources such as atmosphere, passenger, train, and route information, weather data can also be used to make intelligent policies such as speed restrictions, timetables, routes, scheduling special fleets, and maintenance.

B. Periodic and Significant-change Updates

Sensor nodes send periodic updates with the period p. In addition, they send significant-change updates when the change from the last update is bigger than c^{t} for air temperature or c^{w} for wind speed. The parameters $t_{\mathrm{u}}^{\mathrm{t}}$ and $t_{\mathrm{u}}^{\mathrm{w}}$ denote the time of the last update for air temperature and wind speed, respectively. Let $T(t)$ and $W(t)$ denote temperature and wind speed at the time t, respectively. Assume that sensor nodes check observations at intervals of Δt, where $p/\Delta t \in \mathbb{N}$. Then, the time of sending periodic and significant-change updates can be described by Algorithm 1.

Algorithm 1 Periodic and significant-change updates

Input: c^{t}, c^{w}, p
Initialization: Time $t = 0$, $t_{\mathrm{u}}^{\mathrm{t}} = 0$, $t_{\mathrm{u}}^{\mathrm{w}} = 0$
Repeat:
IF $t/p \in \mathbb{N}$ THEN
send $T(t)$ and $W(t)$ in one packet
$t_{\mathrm{u}}^{\mathrm{t}} = t$, $t_{\mathrm{u}}^{\mathrm{w}} = t$
ELSE IF $|T(t) - T(t_{\mathrm{u}}^{\mathrm{t}})| \geq c^{\mathrm{t}}$ THEN
send $T(t)$
$t_{\mathrm{u}}^{\mathrm{t}} = t$
ELSE IF $|W(t) - W(t_{\mathrm{u}}^{\mathrm{w}})| \geq c^{\mathrm{w}}$ THEN
send $W(t)$
$t_{\mathrm{u}}^{\mathrm{w}} = t$
END IF
$t = t + \Delta t$

C. Packet Loss Detection

As mentioned earlier, there are two kinds of packets from the terminal to the edge, i.e., periodical updates and significant-change updates. Since packet loss is easy to detect for periodic updates, we only focus on the detection for significant-change updates. Given that the characteristics of air temperature and wind speed are different, we use different methods to detect packet loss for them.

Trackside air temperature varies continuously and is unlikely to fluctuate greatly within a short duration. Given that the period of the periodical updates is relatively short, we assume that air temperature varies monotonically within a period. So, when Δt approaches 0, i.e., the checking for significant changes is continuous, the estimated number of generated significant-change updates between time ip and time $(i+1)p$ where $i \in \mathbb{N}$ is

$$\hat{n}_i^{\mathrm{g}} = \begin{cases} 0, & \text{for } T(ip) = T((i+1)p) \\ \left\lceil \frac{|T((i+1)p) - T(ip)|}{c^{\mathrm{t}}} \right\rceil - 1, & \text{for } T(ip) \neq T((i+1)p) \end{cases} \quad (1)$$

where $\lceil \cdot \rceil$ is the ceiling function. Packet loss of significant-change updates on air temperature between time ip and time $(i+1)p$ happens when

$$n_i^{\mathrm{r}} < \hat{n}_i^{\mathrm{g}}, \quad (2)$$

where n_i^{r} is the number of received significant-change updates between time ip and time $(i+1)p$. As deterministic formulas, (1) and (2) can be used to detect packet loss for air temperature easily.

Unlike air temperature, wind speed fluctuates greatly. Moreover, even within a short time, it is likely to change so dramatically that we regard it as a discrete variable. Thus, it is difficult to derive a deterministic formula for packet loss detection on wind speed. Since packet loss is an abnormal behavior, data-driven anomaly detection methods are suitable for the detection. The received data on wind speed between time 0 and time $(n+1)p$ where $n \in \mathbb{N}$ can be denoted as

$$\begin{aligned} W_n = \{ & w_0(0), w_1(0), ..., w_{c_0}(0), \\ & w_0(p), w_1(p), ..., w_{c_1}(p), \\ & ..., \\ & w_0(np), w_1(np), ..., w_{c_n}(np)\}, \end{aligned} \quad (3)$$

where $w_0(ip)$ with $i \in [0, n] \cap \mathbb{N}$ is the value of the periodic update at time ip, $w_j(ip)$ with $j \in [1, c_i] \cap \mathbb{N}$ is the value of one significant-change update between time ip and time $(i+1)p$, and $c_i \in \mathbb{N}$ is the number of significant-change updates between time ip and time $(i+1)p$. Packet loss happens when the number of generated packets is not equal to the number of received packets. Since the number of generated packets is related to the changes in the readings, we select five features denoted by a feature vector

$$F_i = [c_i, STD_i, \max_{j\in[0,c_i]} w_j(ip), \min_{j\in[0,c_i]} w_j(ip), w_0(ip)]^{\mathrm{T}}, \quad (4)$$

where STD_i is the standard deviation of $\{w_0(ip), w_1(ip), ..., w_{c_i}(ip)\}$. By slicing the received data by windows with the length of αp where $\alpha \in \mathbb{Z}^+$, the data set can be structured as

$$W_n' = [(I_1', o_1), (I_2', o_2), ..., (I_{n-\alpha+1}', o_{n-\alpha+1})], \quad (5)$$

where $I_i' = [F_i^{\mathrm{T}}, F_{i+1}^{\mathrm{T}}, ..., F_{i+\alpha-1}^{\mathrm{T}}]^{\mathrm{T}}$, and the binary variable o_i is the output of the i^{th} window. The parameter $o_i = 1$ when there is packet loss during the time window, and 0 otherwise. Finally, we use SVM [47], a typical anomaly detection method, to process the data set for packet loss detection.

D. Long-term Prediction for Adaptive Thresholds

There are many methods based on historical data to predict the daily maximum temperature, daily minimum temperature, and daily maximum wind speed [48]–[50]. Given that the data we have collected is limited, we employ autoregressive integrated moving average (ARIMA) [51] which is a robust time-series prediction method. Processing the three kinds of values separately, we use ARIMA to predict the maximum air temperature, minimum air temperature, and maximum wind speed of the next day based on historical daily maximum air temperature, minimum air temperature, and maximum wind speed respectively.

Let $p^{\mathrm{t}}_{\min}$, $p^{\mathrm{t}}_{\max}$, and $p^{\mathrm{w}}_{\max}$ denote the predicted minimum air temperature, maximum air temperature, and maximum wind speed of the next day, respectively. The thresholds of minimum air temperature, maximum air temperature, and maximum wind speed are calculated respectively as

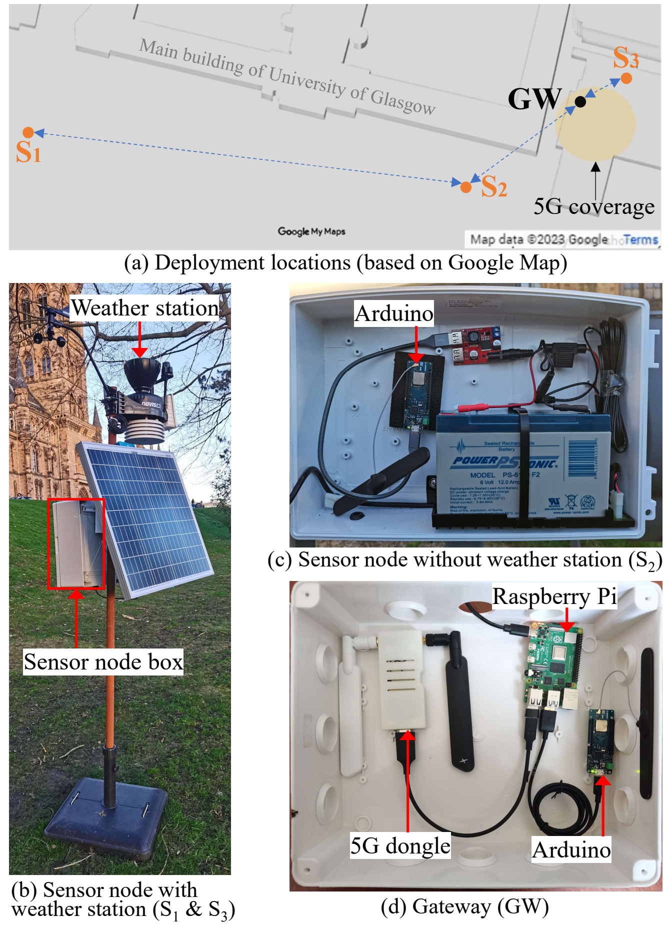
$$H^{t}_{\min} = \max(p^{\mathrm{t}}_{\min}, c^{t}_{\min} + v^{t}_{\min}), \tag{6a}$$
$$H^{t}_{\max} = \min(p^{\mathrm{t}}_{\max}, c^{t}_{\max} - v^{t}_{\max}), \tag{6b}$$
$$H^{\mathrm{w}}_{\max} = \min(p^{\mathrm{w}}_{\max}, c^{\mathrm{w}}_{\max} - v^{\mathrm{w}}_{\max}), \tag{6c}$$

where $c^{\mathrm{t}}_{\min}$, $c^{\mathrm{t}}_{\max}$, and $c^{\mathrm{w}}_{\max}$ are the criteria of adverse weather about low air temperature, high air temperature, and high wind speed, respectively, and $v^{\mathrm{t}}_{\min}$, $v^{\mathrm{t}}_{\max}$, and $v^{\mathrm{w}}_{\max}$ are the fixed values determined by the operator for early warning of adverse weather. $H^{\mathrm{t}}_{\min}$, $H^{\mathrm{t}}_{\max}$, and $H^{\mathrm{w}}_{\max}$ are sent back to the edge and only data outside of the thresholds is sent to the cloud.

V. Implementation

Given the safety and efficiency implications in railway operations as we have highlighted in Section I, we have implemented a proof of concept on the University of Glasgow campus for validation purposes. In this proof of concept, we deploy a LoRa mesh-5G integrated network utilizing the Glasgow 5G testbed. As shown in Fig. 5(a), a GW is deployed within the coverage of the 5G testbed and it connects to the network using its 5G module. In addition, we deploy three sensor nodes i.e., S_1, S_2, and S_3 such that sensors S_2 and S_3 can communicate with GW directly. However, due to blockage by the main building of the University of Glasgow, S_1 can communicate with neither GW nor S_3 directly. Instead, the packets of S_1 must be relayed by S_2 to arrive at GW. Thus, the topology of the LoRa mesh network is illustrated by blue dash lines in Fig. 5(a).

S_1 and S_3 are sensor nodes connected to a weather station. As shown in Fig. 5(b), they consist of a weather station produced by Davis Instruments[1] and a sensor node box that has a weather envoy, a Raspberry Pi, and an Arduino board inside. The weather envoy reads real-time data from the weather station using WiFi and transmits it to the Raspberry Pi using a data logger. The Raspberry Pi is connected to the Arduino board using a universal serial bus (USB) cable. Data processing is realized using the Raspberry Pi. When periodic updates or significant-change updates need to be sent to GW, the Raspberry Pi transmits the update to the Arduino board where a LoRa mesh client runs. Given the requirement of low power consumption and low costs, the LoRa mesh clients in the network are based on RadioHead packet radio library [23]. On receiving data from Raspberry Pi, the LoRa mesh client encapsulates the data as a LoRa mesh packet and sends it over the air to the LoRa mesh client of GW directly or through other sensor nodes. Unlike S_1 and S_3, as shown in Fig. 5(c),



(a) Deployment locations (based on Google Map)

(b) Sensor node with weather station (S_1 & S_3)

(c) Sensor node without weather station (S_2)

(d) Gateway (GW)

Fig. 5. Implementation at the campus of the University of Glasgow. S_1 and S_3 are sensor nodes with weather stations shown in (b). S_2 is a sensor node without a weather station shown in (c). GW is a gateway shown in (d).

S_2 consists of only an Arduino board and a power system. Without weather stations, S_2 only relays LoRa mesh packets.

As shown in Fig. 5(d), GW consists of an Arduino board, a Raspberry Pi, and a 5G dongle. Running on the Arduino board, the LoRa mesh client of GW receives LoRa mesh packets, decapsulates them, and forwards them to the Raspberry Pi. All the data processing tasks of the edge shown in Fig. 4 are realized in the Raspberry Pi. Moreover, an MQTT subscriber and an MQTT publisher are implemented in the Raspberry Pi using Node-RED[2] which is a programming tool providing a wide range of nodes with various functions. The 5G dongle is assembled at the University of Glasgow and can access the Glasgow 5G testbed. By the 5G dongle, the LoRa mesh gateway can communicate with the LoRa mesh server deployed in the 5G core network.

Glasgow 5G testbed is equipped with an on-premises private cloud. We create a virtue machine instance on the cloud to implement the LoRa mesh server. In the implementation, we only realize the functions of data volume reduction and dashboard as the functions of alert and policy are out of the scope of this paper. For bidirectional communication with

[1]https://www.davisinstruments.com/
[2]https://nodered.org/

GW, we implement an MQTT broker, an MQTT subscriber, and an MQTT publisher on the server also using Node-RED. Moreover, Node-RED is used to process the received data and store it in the long-term global database which is realized by InfluxDB[3]. For data volume reduction, we use Python to read data from InfluxDB, make predictions, and store the results. i.e., thresholds, in InfluxDB again. Node-RED monitors InfluxDB for the thresholds and sends them to GW. We use Grafana[4] to directly read from InfluxDB, process data, and display data on a dashboard. Grafana provides an HTTP-based user interface by which users can access the dashboard from external networks easily. Besides, we implement another user interface for changing the parameters of the network such as the update frequency of LoRa mesh sensor nodes. By deploying HTTP end-points on the Node-RED, the LoRa mesh server provides HTTP-based web services that users can call from external networks to change parameters. With the two user interfaces, authorized users can monitor the network and change parameters easily.

VI. Experimental Results

In this section, we will describe the experimental results of periodic and significant-change updates, packet loss detection, and adaptive thresholds.

A. Periodic and Significant-change Updates

To investigate the benefits of timely significant-change updates, we record data in a sensor node from May 01, 2023, to May 17, 2023, with Δt = 1 second. By doing so, the results of different p, c_{t}, and c_{w} can be calculated from one data set. To quantify the delay between a significant change in an observation happening and the track manager knowing it, we define the average delay between time 0 and time $(n+1)p$ where $n \in \mathbb{N}$ as

$$\hat{d} = 1440 \cdot \frac{\sum_{i=0}^{n} d_i}{(n+1)p}, \tag{7}$$

where the constant number 1440 is for transforming the unit of $\hat{d}$ to minutes per day and d_i is the delay occurring between ip and $(i+1)p$, such that $d_i = 0$ if $|T(t) - T(ip)| < c^{\mathrm{t}}$ and $|W(t) - W(ip)| < c^{\mathrm{w}}$ for $\forall t \in (ip, (i+1)p)$. Otherwise, $d_i = \max\left((i+1)p - t\right)$ where t satisfies $|T(t) - T(ip)| \geq c^{\mathrm{t}}$ or $|W(t) - W(ip)| \geq c^{\mathrm{w}}$. With the definition, the average delay is illustrated in Fig. 6. For sending only periodic updates, the average delay increases significantly with the increase of p, c^{t}, and c^{w}. However, as shown in Fig. 7, increasing p can significantly reduce the number of packets. Thus, periodic updates cannot achieve low packet rates and low average delay at the same time. Adding significant-change updates solved the problem. The green line in Fig. 6 illustrates the average delay in sending periodic and significant-change updates. Regardless of p, c^{t}, and c^{w}, it approaches zero as sensor nodes send updates once a significant change is detected. On the other hand, its number of packets also reduces significantly with the increase of p. Compared with periodic updates, adding significant-change updates increases the number of packets very slightly, e.g., using significant-change updates with $c^{\mathrm{t}} = 2$ °C and $c^{\mathrm{w}} = 20$ mph only increases about 3 packets per day when p is between 5 and 5.08 minutes. Therefore, it is beneficial to use periodic updates and significant-change updates to reduce packet rates and average delay at the same time.

[3]https://www.influxdata.com/
[4]https://grafana.com/

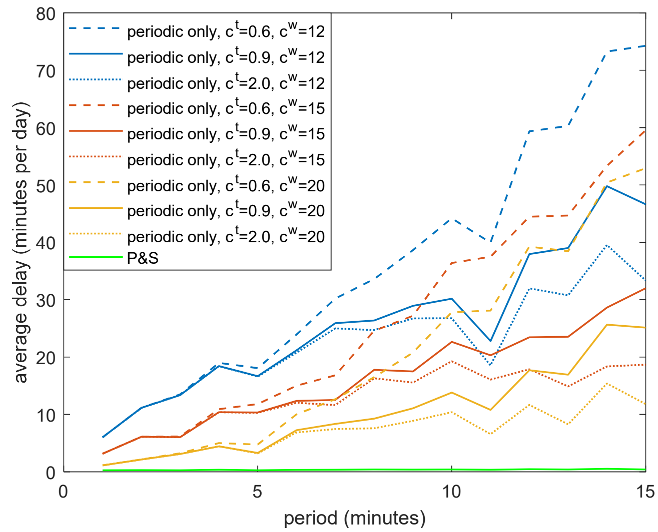


Fig. 6. Average delay with different periods and significant-change criteria. c^{t} and c^{w} are in °C and mph, respectively. "P&S" denotes that sensor nodes send periodic and significant-change packets.

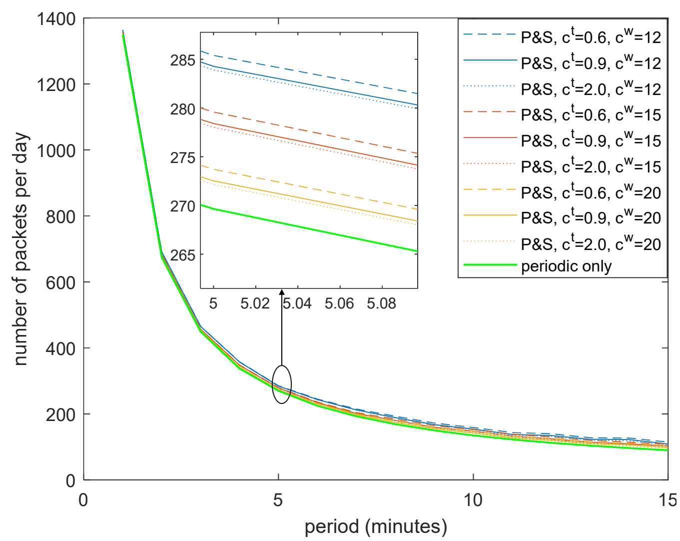


Fig. 7. The number of packets with different periods and significant-change criteria. c^{t} and c^{w} are in °C and mph, respectively. "P&S" denotes that sensor nodes send periodic and significant-change packets.

B. Packet Loss Detection

We collect data directly from sensor nodes, including generated periodical updates and significant-change updates, from December 14, 2022, to January 20, 2023, with $p = 15$ minutes, $c^{\mathrm{t}} = 2$ °C, and $c^{\mathrm{w}} = 12$ mph. Assume that all the periodical updates are received by GW.

In terms of air temperature, $n_{\mathrm{t}}^{\mathrm{g}} = \hat{n}_{\mathrm{t}}^{\mathrm{g}}$ for 99.97% of the data, which means the estimation of (1) has the accuracy of 99.97%. In this situation, the result of packet loss detection is correct no matter whether there is a packet loss or not.

In terms of wind speed, given that there are not enough lost packets during this time to do the training and test of SVM, we randomly delete 30% of the significant-change updates and then generate samples with features by the moving window method which are described in (4) and (5). The samples with deleted significant-change updates are labeled as abnormal, or otherwise normal. Then, 80% of the samples are randomly selected to train the SVM algorithm, and the remaining 20% of the samples are utilized to evaluate the performance. Since packet loss detection is a supervised anomaly detection problem, we select four common performance metrics including $precision$, $recall$, $F1score$, and $accuracy$ which are defined as

$$precision = \frac{TP}{TP+FP}, \tag{8a}$$
$$recall = \frac{TP}{TP+FN}, \tag{8b}$$
$$accuracy = \frac{TP+TN}{TP+FP+FN+TN}, \tag{8c}$$
$$F1score = \frac{2*precison*recall}{precison+recall}, \tag{8d}$$

where TP denotes true positive that the loss of significant-change updates is successfully classed as an abnormality, TN denotes true negative that a normal situation is identified correctly, FP denotes false positive that an abnormality is incorrectly classed as normality, and FN denotes false negative that a normal situation is incorrectly identified as an abnormality. To obtain the optimal window length, we repeat the experiments with different window lengths. As shown in Fig. 8, with the window length increasing, $recall$ and $accuracy$ drop when the window length is less than 1 hour and become stable when the window length is bigger than 2 hours. $precision$ and $F1score$ increase with the window length increasing and become stable when the window length is bigger than 15/4 hours. Therefore, the window length should be set bigger than 15/4 hours for high performance of the packet loss detection algorithm. Given computation complexity and storage requirement, the window length is set as 15/4 hours which also serves as the retention time of the local database in the edge.

C. Adaptive Thresholds

We also use the data collected from the implemented network between December 14, 2022, and January 20, 2023, for the adaptive threshold algorithm. ARIMA model is fit by the data from December 14, 2022, to January 10, 2023. The remaining data is used to evaluate the prediction performance by root mean square error (RMSE) and mean absolute error (MAE) which are defined as

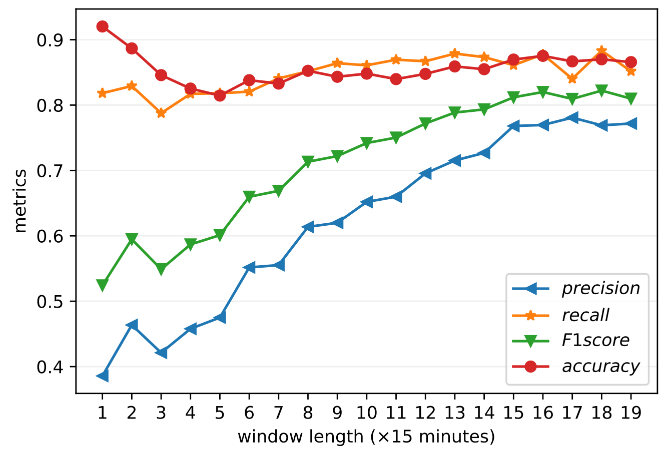


Fig. 8. Result of packet loss detection.

TABLE I
RESULTS OF ADAPTIVE THRESHOLDS.

Prediction objective	RMSE	MAE	Data reduction rate
Max air temperature	1.906 °C	1.544 °C	72.23%
Min air temperature	1.231 °C	0.983 °C	
Max wind speed	6.997 mph	6.097 mph	97.96%

$$RMSE = \sqrt{\frac{1}{n_{\mathrm{e}}}\sum_{i=1}^{n_{\mathrm{e}}}(y_i - \hat{y}_i)^2}, \tag{9a}$$
$$MAE = \frac{1}{n_{\mathrm{e}}}\sum_{i=1}^{n_{\mathrm{e}}}|y_i - \hat{y}_i|, \tag{9b}$$

where y_i is the i^{th} actual value, $\hat{y}_i$ is the corresponding predicted value, and n_{e} is the total number of the values to be evaluated. As shown in Table I, the RMSE of the prediction for the maximum air temperature, minimum air temperature, and maximum wind speed of the next day is 1.906 °C, 1.231 °C, and 6.997 mph, respectively. The MAE of the prediction about the maximum air temperature, minimum air temperature, and maximum wind speed of the next day is 1.544 °C, 0.983 °C, and 6.097 mph, respectively. With the prediction accuracy, the data reduction rates from edge to the cloud for air temperature and wind speed are 72.23% and 97.96%, respectively.

VII. Conclusion and Future Work

In this paper, we have proposed a LoRa mesh-5G integrated network for trackside smart weather monitoring. We first presented the network model and integrated the LoRa mesh network into a 5G network, which significantly reduced the cost of deploying communication infrastructure for wide and remote coverage. We employed the proposed network for weather monitoring and designed a cloud-edge-terminal collaborative architecture to bring artificial intelligence to the IoT network. Utilizing three intelligent algorithms, i.e., timely significant-change updates, packet loss detection, and adaptive thresholds, the proposed architecture reduced the data volume and achieved self-detection for packet loss. With reduced data volume, the network can support more end devices and provide wider coverage. Based on the Glasgow 5G testbed, a proof of concept was implemented using low-power and low-cost off-the-shelf hardware. The experimental results demonstrated the feasibility of the proposed integrated network and cloud-edge-terminal collaborative architecture. The proposed LoRa mesh-5G integrated network satisfies all the identified trackside sensing requirements, i.e., wide coverage, supporting massive end devices, low-cost communication infrastructure, and low-power and low-cost end devices.

However, the possibility of signal collision and interference could rise with an increase in the deployed end devices, posing a threat to network reliability. Moreover, duty cycle regulation [52] on the frequency bands utilized by LoRa could constrain its scalability. In our future work, we plan to analyze these aspects of the network and introduce novel LoRa mesh routing algorithms to address the potential challenges associated with reliability and scalability.

ACKNOWLEDGMENTS

This research was partly funded by 5G3i Ltd. corporation and reports work (partially) undertaken in the context of the project Satellites for Digitalization of Railways (SODOR). SODOR is a project that is led by CGI and it is part of a joint initiative by UK Department for DCMS, UK Space Agency and European Space Agency (ESA) in the UK to demonstrate the integrated use of 5G in the area of transport and logistics. The authors would like to thank Mr. Habib Manzoor from University of Glasgow, UK, for drafting the code about data encoding.

REFERENCES

[1] "Railway statistics synopsis 2022," International Union of Railways (UIC), Paris, France. Accessed: Jun. 14, 2023. [Online]. Available: https://uic.org/IMG/pdf/uic-railway-statistics-synopsis-2022.pdf

[2] "Winter weather can present some real challenges for the railway – here's how we respond." Network Rail. https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/delays-explained/snow-and-ice/ (accessed: Jun. 14, 2023).

[3] P. Fraga-Lamas, T. M. Fernández-Caramés, and L. Castedo, "Towards the Internet of smart trains: A review on industrial IoT-connected railways," *Sensors*, vol. 17, no. 6, p. 1457, 2017.

[4] "Mobile signal strength measurement data from Network Rail's engineering trains," Ofcom, London, UK. Accessed: Jun. 14, 2023. [Online]. Available: https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2019/data-downloads

[5] R. Rowson, "GB railway mobile phone network 4G coverage." Tableau Public. https://public.tableau.com/app/profile/richard.rowson/viz/lte/Railway4Gsignalstrength (accessed: Jun. 14, 2023).

[6] Y. Chen, Y. A. Sambo, O. Onireti, and M. A. Imran, "A survey on LPWAN-5G integration: Main challenges and potential solutions," *IEEE Access*, vol. 10, pp. 32 132–32 149, 2022.

[7] R. Yasmin, J. Petäjäjärvi, K. Mikhaylov, and A. Pouttu, "On the integration of LoRaWAN with the 5G test network," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–6.

[8] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, "Integration of LoRaWAN and 4G/5G for the industrial internet of things," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 60–67, 2018.

[9] H. Jradi, F. Nouvel, A. E. Samhat, J.-C. Prévotet, and M. Mroue, "A seamless integration solution for LoRaWAN into 5G system," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 16238–16252, 2023.

[10] P. Singh, Z. Elmi, V. K. Meriga, J. Pasha, and M. A. Dulebenets, "Internet of things for sustainable railway transportation: Past, present, and future," *Cleaner Logistics and Supply Chain*, vol. 4, p. 100065, 2022.

[11] B. Ai, A. F. Molisch, M. Rupp, and Z.-D. Zhong, "5G key technologies for smart railways," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 856–893, 2020.

[12] Y. Liu, C.-X. Wang, and J. Huang, "Recent developments and future challenges in channel measurements and models for 5G and beyond high-speed train communication systems," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 50–56, 2019.

[13] H. Song, X. Fang, and L. Yan, "Handover scheme for 5G C/U plane split heterogeneous network in high-speed railway," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4633–4646, 2014.

[14] D. He *et al.*, "Channel measurement, simulation, and analysis for high-speed railway communications in 5G millimeter-wave band," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3144–3158, 2017.

[15] K. Ko, W. Ahn, and W. Shin, "High-speed train positioning using deep Kalman filter with 5G NR signals," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 993–16 004, 2022.

[16] O. Jo, Y.-K. Kim, and J. Kim, "Internet of things for smart railway: feasibility and applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 482–490, 2017.

[17] V. Deniau *et al.*, "Analysis of the susceptibility of the LoRa communication protocol in the railway electromagnetic environment," in *2021 XXXIVth General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*. IEEE, 2021, pp. 1–4.

[18] J. Villain *et al.*, "Detection and classification of interference affecting LoRaWAN communications in railway environment," in *2022 3rd URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC)*. IEEE, 2022, pp. 1–4.

[19] K. Naveen and S. Rayala, "Automatic railway level crossing using LoRa technology," *Materials Today: Proceedings*, 2023.

[20] D. Ferretti, P. Lanci, B. Torun, D. Amato, and R. Verdone, "LoRa-based railway signalling system for secondary lines," in *2022 61st FITCE International Congress Future Telecommunications: Infrastructure and Sustainability (FITCE)*. IEEE, 2022, pp. 1–6.

[21] D. Lundell, A. Hedberg, C. Nyberg, and E. Fitzgerald, "A routing protocol for LoRa mesh networks," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2018, pp. 14–19.

[22] R. Berto, P. Napoletano, and M. Savi, "A LoRa-based mesh network for peer-to-peer long-range communication," *Sensors*, vol. 21, no. 13, p. 4314, 2021.

[23] *RadioHead.* (2023). AirSpayce. Accessed: Jun. 14, 2023. [Online]. Available: https://www.airspayce.com/mikem/arduino/RadioHead/

[24] H.-C. Lee and K.-H. Ke, "Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2177–2187, 2018.

[25] H. Huh and J. Y. Kim, "LoRa-based mesh network for IoT applications," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 524–527.

[26] C. Ebi, F. Schaltegger, A. Rüst, and F. Blumensaat, "Synchronous LoRa mesh network to monitor processes in underground infrastructure," *IEEE Access*, vol. 7, pp. 57 663–57 677, 2019.

[27] S. Hong, F. Yao, Y. Ding, and S.-H. Yang, "A hierarchy-based energy-efficient routing protocol for LoRa-mesh network," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 836–22 849, 2022.

[28] P. Tian, C. A. Boano, X. Ma, and J. Wei, "LoRaHop: Multi-hop support for LoRaWAN uplink and downlink messaging," *IEEE Internet of Things Journal*, 2023.

[29] D. Wu and J. Liebeherr, "A low-cost low-power LoRa mesh network for large-scale environmental sensing," *IEEE Internet of Things Journal*, vol. 10, no. 19, pp. 16700–16714, 2023.

[30] A. Ksentini and P. A. Frangoudis, "On extending ETSI MEC to support LoRa for efficient IoT application deployment at the edge," *IEEE Communications Standards Magazine*, vol. 4, no. 2, pp. 57–63, 2020.

[31] S. Kekki *et al.*, "MEC in 5G networks," ETSI, Sophia Antipolis, France, 2018. Accessed: Jun. 14, 2023. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[32] E. M. Torroglosa-Garcia, J. M. A. Calero, J. B. Bernabe, and A. Skarmeta, "Enabling roaming across heterogeneous IoT wireless networks: LoRaWAN meets 5G," *IEEE Access*, vol. 8, pp. 103 164–103 180, 2020.

[33] Y. Chen, Y. A. Sambo, O. Onireti, S. Ansari, and M. A. Imran, "LoRaWAN-5G integrated network with collaborative RAN and converged core network," in *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2022, pp. 1–5.

[34] J. Zhang *et al.*, "Enabling blind area coverage for the smart grids: Integrating energy-efficient LoRa technologies in the 5G," in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. IEEE, 2021, pp. 952–956.
[35] V. K. Sarker, J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "A survey on LoRa for IoT: Integrating edge computing," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2019, pp. 295–300.
[36] Z. Liu, Q. Zhou, L. Hou, R. Xu, and K. Zheng, "Design and implementation on a LoRa system with edge computing," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.
[37] V. K. Sarker, T. N. Gia, I. Ben Dhaou, and T. Westerlund, "Smart parking system with dynamic pricing, edge-cloud computing and LoRa," *Sensors*, vol. 20, no. 17, p. 4669, 2020.
[38] P. Kumari, R. Mishra, H. P. Gupta, T. Dutta, and S. K. Das, "An energy efficient smart metering system using edge computing in LoRa network," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 4, pp. 786–798, 2021.
[39] F. Sharofidinov, M. S. A. Muthanna, V. D. Pham, A. Khakimov, A. Muthanna, and K. Samouylov, "Agriculture management based on LoRa edge computing system," in *Distributed Computer and Communication Networks: 23rd International Conference, DCCN 2020, Moscow, Russia, September 14–18, 2020, Revised Selected Papers*. Springer, 2020, pp. 113–125.
[40] L. Hou, K. Zheng, Z. Liu, X. Xu, and T. Wu, "Design and prototype implementation of a blockchain-enabled LoRa system with edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2419–2430, 2020.
[41] A. W.-L. Wong, S. L. Goh, M. K. Hasan, and S. Fattah, "Multi-hop and mesh for LoRa networks: Recent advancements, issues, and recommended applications," *ACM Computing Surveys*, vol. 56, no. 6, article. 136, 2024.
[42] *Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1*, ISO/IEC 20922:2016, June 2016. [Online]. Available: https://www.iso.org/standard/69466.html
[43] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT," in *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*. IEEE, 2016, pp. 1–6.
[44] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Applied Sciences*, vol. 9, no. 5, p. 848, 2019.
[45] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.
[46] G. Ferré, "Collision and packet loss analysis in a LoRaWAN network," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2586–2590.
[47] M. Hosseinzadeh, A. M. Rahmani, B. Vo, M. Bidaki, M. Masdari, and M. Zangakani, "Improving security using SVM-based anomaly detection: issues and challenges," *Soft Computing*, vol. 25, pp. 3195–3223, 2021.
[48] Z. Karevan and J. A. Suykens, "Transductive LSTM for time-series prediction: An application to weather forecasting," *Neural Networks*, vol. 125, pp. 1–9, 2020.
[49] A. Paniagua-Tineo, S. Salcedo-Sanz, C. Casanova-Mateo, E. Ortiz-García, M. Cony, and E. Hernández-Martín, "Prediction of daily maximum temperature using a support vector regression algorithm," *Renewable Energy*, vol. 36, no. 11, pp. 3054–3060, 2011.
[50] T. L. Thorarinsdottir and T. Gneiting, "Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, no. 2, pp. 371–388, 2010.
[51] N. Shivhare, A. K. Rahul, S. B. Dwivedi, and P. K. S. Dikshit, "ARIMA based daily weather forecasting tool: A case study for Varanasi," *Mausam*, vol. 70, no. 1, pp. 133–140, 2019.
[52] "ETSI EN 300 220-2 v3.2.1 (2018-06)," ETSI, Sophia Antipolis, France, Rep. REN/ERM-TG28-535, 2018. Accessed: Jan. 22, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.02.01_60/en_30022002v030201p.pdf

Yu Chen (Student Member, IEEE) received the B.S. degree in measurement and control technology and instrumentation and the M.S. degree in instrument science and technology from the University of Electronic Science and Technology of China, in 2017 and 2020, respectively. Currently, he is working towards the Ph.D. degree in electronics and electrical engineering at the University of Glasgow, U.K. His research interests include self-organizing networks, hybrid network management, and compressive sampling.

Guo Shi received the B.S. degree in Industrial Engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018, and the M.S. degree in Industrial Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021. Currently, she is a Ph.D. student in the Department of Management Science at the University of Strathclyde, Glasgow, U.K. Her research interests include data-driven decision-making, machine learning, maintenance modeling, prognostic and health management.

Mohammad Al-Quraan (Graduate Student Member, IEEE) received the B.Sc. (Hons.) degree in telecommunications engineering and the M.Sc. (Excellence) degree in wireless telecommunications engineering from Yarmouk University, Irbid, Jordan, in 2011 and 2019, respectively. He is currently working toward the Ph.D. degree in electronics and electrical engineering with the University of Glasgow, Glasgow, U.K. From 2012 to 2018, he was a Senior Network and Telecommunications Engineer with the Jordan University of Science and Technology (JUST), Irbid. Then till 2020, he was the Head of the Network and Telecommunications Department, JUST. His research interests include machine learning, computer vision, cognitive radio, and beyond 5G wireless technologies.

Yusuf Sambo (Senior Member, IEEE) received the MSc degree (distinction) in Mobile and Satellite Communications, in 2011 and the Ph.D. degree in electronic engineering in 2016 from the Institute for Communication Systems (ICS, formally known as CCSR), University of Surrey. He was a Postdoctoral Research Associate in the Communications, Sensing and Imaging (CSI) research group at the University of Glasgow and since 2019, Lecturer at the James Watt School of Engineering, University of Glasgow and 5G Testbed Lead managing the Scotland 5G Centre testbed at the University of Glasgow. Prior to joining Glasgow, he was a Lecturer in Telecommunications Engineering at Baze University, Abuja between 2016 and 2017. His interests include self-organizing networks, cell-free massive MIMO, EMF exposure, and green communications.

Oluwakayode Onireti (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from the University of Ilorin, Ilorin, Nigeria, in 2005, and the M.Sc. degree (Hons.) in mobile and satellite communications, and the Ph.D. degree in electronics engineering from the University of Surrey, Guildford, U.K., in 2009 and 2012, respectively. He is currently a Lecturer with the University of Glasgow, Glasgow, U.K. He has been actively involved in projects, such as ROCKET, EARTH, Greencom, QSON, DARE, and Energy proportional EnodeB for LTEAdvanced and Beyond. His main research interests include self-organizing cellular networks, energy efficient networks, wireless blockchain networks, millimeter wave communications, and cooperative communications.

Muhammad Imran (Fellow, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees from Imperial College London, London, U.K., in 2002 and 2007, respectively. He is currently the Dean of University of Glasgow UESTC, Head of Autonomous Systems and Connectivity Division, and a Professor of communication systems with the James Watt School of Engineering, University of Glasgow, Glasgow, U.K. He is an Affiliate Professor with The University of Oklahoma, Norman, OK, USA, and 5G Innovation Centre, University of Surrey, Guildford, U.K. He is leading research with the University of Glasgow for Scotland 5G Centre. He has more than 20 years of combined academic and industry experience with several leading roles in multi-million pound funded projects, working primarily in the research areas of cellular communication systems. He was the recipient of an Award of Excellence in recognition of his academic achievements, conferred by the President of Pakistan. He was also the recipient of the IEEE Comsoc's Fred Ellersick Award 2014, FEPS Learning and Teaching Award 2014, Sentinel of Science Award 2016, and ten patents.