Automation in Construction 150 (2023) 104835

Contents lists available at ScienceDirect

Automation in Construction

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

AIoT-informed digital twin communication for bridge maintenance

Yan Gao [a], Haijiang Li [*,a], Guanyu Xiong [a], Honghong Song [a,b]

[a] BIM for Smart Engineering Centre, School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK
[b] Department of civil Engineering, Dalian Maritime University, Dalian 116026, China

ARTICLE INFO

Keywords:
Digital twin
Bridge maintenance
Communication complexity
Time delay
Resilience
Edge computing
LPWAN

ABSTRACT

Digital twin (DT) has been moving progressively from concept to practice for bridge operation and maintenance (O&M), but its issues of data synchronization and fault tolerance remain problematic. This paper investigates the time delay of bridge DT services according to communication and computation complexity, revealing the distinct impact of their sequence, and proposes an AIoT-informed DT communication framework to solve the above issues. The information hierarchy and two-way communication can be leveraged to minimize communication complexity in the framework. Meanwhile, the data flow and resilience of the proposed framework are demonstrated using a Petri net. Moreover, the framework is developed into a prototypical DT through cross-platform integration and validated with different cases. The results demonstrate that compared with other existing bridge DTs, the proposed framework has high efficiency, low-latency, and excellent fault tolerance, which can contribute to the efficiency and safety of bridge O&M, especially under communication-constraint circumstances. The framework is also promising for federated learning to protect the AI-model privacy of different stakeholders and has the potential to support agent-based intelligent bridge management in the future with little human intervention.

1. Introduction

Bridges play a critical role in transport systems, and their failure will result in traffic disruption, economic loss, or even severe casualties. According to the ASCE report in 2021, 6154 (over 7.5%) of the nation's bridges are considered structurally deficient in the US, and unfortunately, 178 million trips are taken across these bridges every day [1]. Similarly, over 3105 bridges (around 4.3%) are identified as substandard in the UK until Jan 2021 [2]. As the deterioration and failures of aging bridges increase every year, it is necessary to keep regular inspections, effective monitoring, timely condition assessment, and optimal maintenance for the safety of bridge operation, and prolong their service life. Currently, autonomous inspection with drones [3] or climbing robots [4], and real-time structural health monitoring (SHM) with various sensors [5] have been accepted as effective methods to indicate structural deficiency and the influence on bridge capability before maintenance. Meanwhile, artificial intelligence (AI), big-data analysis [6], knowledge-based reasoning [7], and multi-objective optimization [8] are explored for bridge operation and maintenance (O&M), showing the potential for intelligent bridge management in the future with very little human intervention.

Digital twin (DT) is promising to achieve smart bridge O&M through different DT services. It should be able to process big data from multiple sources in near real-time, such as physical bridges, traffic, weather forecast, inventory, and historical records, and make holistic decision-making for bridge O&M. For example, a bridge DT can release early warnings and quick response for physical bridges and public users when a disaster is happening or predicted to happen; it can also provide the optimized maintenance planning considering the benefits of different stakeholders, such as the minimal traffic disruption. However, when DT comes to practical application for bridges, most successful frameworks and systems from mature fields, e.g., intelligent manufacturing, cannot be employed directly due to bridge characteristics. For example, when DT implementation concerns bridge locations, it may have issues, such as restricted communication. At the same time, massive heterogeneous data from regular inspection and real-time SHM on physical bridges makes it difficult to design an appropriate bridge DT framework to support timely DT services in a communication-constrained environment such as a low-power wide area network (LPWAN). Currently, most existing bridge DTs [9–11] are cloud-based and rely on excellent communication without any consideration of fault tolerance, e.g., to endure a temporary loss of communication. In practice, such an issue

* Corresponding author.
E-mail address: lih@cardiff.ac.uk (H. Li).

https://doi.org/10.1016/j.autcon.2023.104835
Received 4 October 2022; Received in revised form 23 February 2023; Accepted 11 March 2023
Available online 22 March 2023

about resilience is critical for physical entities (PE), which can put physical bridges and public users in danger, especially in case of emergency. Moreover, the transmission of massive data can result in a significant time delay for DT synchronization as well, which will decrease the efficiency of DT services in bridge maintenance such as inspection. Hence, it is necessary to propose a highly efficient, low-latency, and resilient DT framework, which can work under communication-constraint circumstances to support bridge O&M.

This work proposes an AIoT-informed DT communication framework to solve the above issues. At first, the study indicates that the time delay in DT services consists of computation and communication time costs, depending on computational and communication complexity respectively. Edge computing can reduce the time delay of DT services significantly when communication time cost dominates in the process, which usually occurs under restricted communication. Moreover, the information hierarchy, also known as the DIKW pyramid, can be leveraged to indicate how to decrease the complexity of the transmission using AI-based edge computing. Two-way communication is recommended to minimize the communication complexity for big-data analysis in a communication-constraint environment involving different data sources owned by edge and cloud, such as holistic bridge assessment, thereby decreasing the time delay. A hierarchical architecture with high resilience based on the mesh network is designed to endure a temporary loss of communication at different levels. The bridge DT system is idealized mathematically in state-space representation with time delay and inequalities for hardware processing capability. Moreover, the proposed framework is modeled using a Petri net to demonstrate the data flow for DT services and framework resilience using tokens and conditional probability respectively. Furthermore, the proposed framework is developed into a cross-platform prototype step by step, integrating AIoT-based edge devices, LPWAN, MQTT protocols, cloud servers, and a web-based platform with both GIS and BIM. Finally, the framework is validated with different cases for bridge O&M, i.e., drone-enabled bridge inspection, vibration-based monitoring (VBM), and dynamic route planning for evacuation. The performance demonstrates the high efficiency, low latency, and excellent fault tolerance of the framework, which can contribute directly to enhancing the efficiency and safety of bridge O&M through DT.

The contribution of this research is three-fold:

1) An AIoT-informed DT communication framework is proposed to support smart bridge O&M under communication-constrained circumstances with high efficiency, low latency, and excellent fault tolerance to endure a temporary loss of communication. Furthermore, the framework is developed into a cross-platform prototype, integrating AIoT-based edge devices, LPWAN, MQTT protocols, cloud servers, and a web-based platform, and validated with different cases for bridge O&M.
2) The study investigates the time delay of bridge DT services according to communication and computation complexity, which reveals the distinct impact of their sequence. AI-based edge computing can decrease the time delay of DT services significantly when communication time dominates in the process. Moreover, the information hierarchy can be leveraged to reduce communication complexity, and two-way communication is recommended to satisfy restricted communication with minimal complexity for big-data analysis involving different sources owned by edge and cloud separately.
3) The bridge DT is idealized in state-space representation with time delay, which is beneficial to understanding the bridge control based on DT. Meanwhile, the data flow and resilience of the proposed framework are demonstrated using a Petri net with tokens and conditional probability respectively.

The rest of this paper is structured as follows: Section 2 overviews the related work and technologies with bridge DTs and introduces the research problems. Section 3 presents the theoretical foundation for the proposed framework. Section 4 is for framework design and prototype development. Section 5 is framework validation with different cases during bridge O&M. Section 6 concludes the work and discusses the further direction for this research.

2. Literature review

2.1. Bridge DT review

Although there is hardly a unified definition for DT in different disciplines and domains currently, the ongoing research exhibits a notable characteristic that DTs are designed and developed for specific purposes and circumstances [12,13]. For example, a bridge DT for SHM is defined as a virtual representation of the physical bridge, which does not only update as new data is collected in near real-time but also provides feedback into the physical bridge and performs 'what-if' scenarios for assessing asset risks and predicting asset performance [14]. A DT for bridge maintenance aims to be updated along with visual inspection and non-destructive test (NDT) continuously [15–17] and integrated with other multi-source data, such as original design, damage history, inventory, traffic, weather, disaster, to support holistic decision-making for maintenance planning [18,19]. Moreover, multiple bridges DTs can be considered as a bridge network and utilized for intelligent transport, which is usually represented topologically on a map [20]. A comprehensive and sophisticated DT system to support bridge O&M in this research aims to take all the above purposes (or services) into account. The bridge DT models can be created by different approaches technically, including building information modeling (BIM), physics-based approach (such as finite element modeling), data-driven approach (such as statistical modeling), and data-centric engineering approach (i.e., hybrid modeling) [14]. Their key features include digital replica (including geometry, materials, etc.); data composition; bidirectional connection (update and feedback) in near real-time; the lifecycle span of a physical bridge; common data environment (CDE); visualization; simulation; learning from actual measurement data [14].

Over the past decade, the interest in bridge DT has grown significantly, and a few prototypes and pilot projects have been proposed and demonstrated successfully. For example, a DT system for two pilot railway bridges [21] was developed, which integrates Fiber Bragg Gratings (FBG) sensors, laser rangefinders, and other additional sensors to achieve on-going monitoring for train-bridge coupling parameters (e.g., strains, accelerations, train axle positions). It can calculate the key indicators (e.g., curvature, end rotations, displacements, axle weights) for SHM in the cloud, as well as use a web-based platform embedded with Unity for human-machine interaction (HMI). Another cloud-based bridge DT for SHM [22] employs a finite element model with damaged states and data synthetically created, real-world monitoring data (such as vibration and strain) from multiple sensors, and a pre-trained surrogate model based on deep learning to detect damage existence, identify damage location, and quantify damage severity on a practical bridge, thereby achieving proactive maintenance. Moreover, a conceptual bridge DT [11,17,23] for preventative maintenance is developed using a surface model, BIM model, and simulation model. The surface model is generated continuously through reverse engineering (photograph mapping and 3D scanning) and aligned parametric modeling during the bridge lifecycle O&M. Damage information through image processing after visual inspection can be recorded in a code system and linked to specific BIM elements. The simulation model is achieved with FEM, wherein the detected deterioration of structural elements is evaluated and employed to update the structural parameters. Another exemplary openBIM-based bridge DT [1] was developed by industry, which can enable long-term monitoring of bridge condition and predictive maintenance (PM) with aggregated information, by combining traditional inspections with digital information (from structural diagnosis and monitoring) and injecting the derived semantic information into the BIM model. Furthermore, there are also many other frameworks or systems developed in the forms of the bridge management system (BMS) or bridge information modeling (BrIM) for bridge SHM [5,24–27], and O&M [19,28–30]. They are very instructive due to the close relationship to bridge DT, which can be taken as resorts and further developed to bridge DT.

Although there are already some successful designs and implementations for bridge DT, most of them relied on excellent communication [21,31,32], such as Ethernet, 4G, and 5G, or even did not reveal communication approaches such as conceptual designs [11,17,23]. However, in practice, when bridge DT implementation concerns bridge location, many bridges are working under communication-constraint circumstances, which can be attributed to either geographic or economic reasons. For example, there are thousands of bridges in the UK like the one in the pilot project, which has successful DT implementation based on excellent communication [21], but widely spreading such exemplars will lead to large expenditures on middleware and data plans. Meanwhile, storing such a huge amount of inspection and monitoring data will also become a heavy burden in terms of both technology and the economy. Moreover, most of the existing bridge DTs work in a centralized mode based on the cloud [10,33–35], as shown in Fig. 1. As can be seen, the digital bridge can synchronize with the physical bridge through the triplet bridge-machine-human interaction, which involves four significant components, i.e., data acquisition and preprocessing, communication, cloud servers, human-machine interface (HMI), to achieve multiple DT services at the back end. However, the issue of resilience in such an architecture [36,37] is usually neglected. Hence, it raises two questions. Firstly, do we need to transmit such massive heterogeneous data for bridge DT, especially in a communication-constraint environment? Secondly, how can we enable bridge DT with fault-tolerant capability, e.g., under a temporary loss of communication? As indicated in the investigation based on practitioners' views in the UK [38], the first question is also related to the major disconnect between academia and industries towards DT applications in bridge O&M, which is the difficulty to keep the bridge DT models updated automatically in routine practice. It can be solved by collecting data continuously from traditional inspections and ongoing structural monitoring, but one of the bottlenecks is focused on the synchronization of massive heterogeneous data. Meanwhile, to our best knowledge, there is no precedent of bridge DT framework considering resilience to endure a temporary loss of communication. This work is going to study the bridge DT communication to solve the above issues from the following perspectives: 1) to enhance DT service efficiency, i.e., to bridge the gap between complexity and time delay of service; 2) to enable bridge with excellent fault tolerance to endure a temporary loss of communication at different levels.

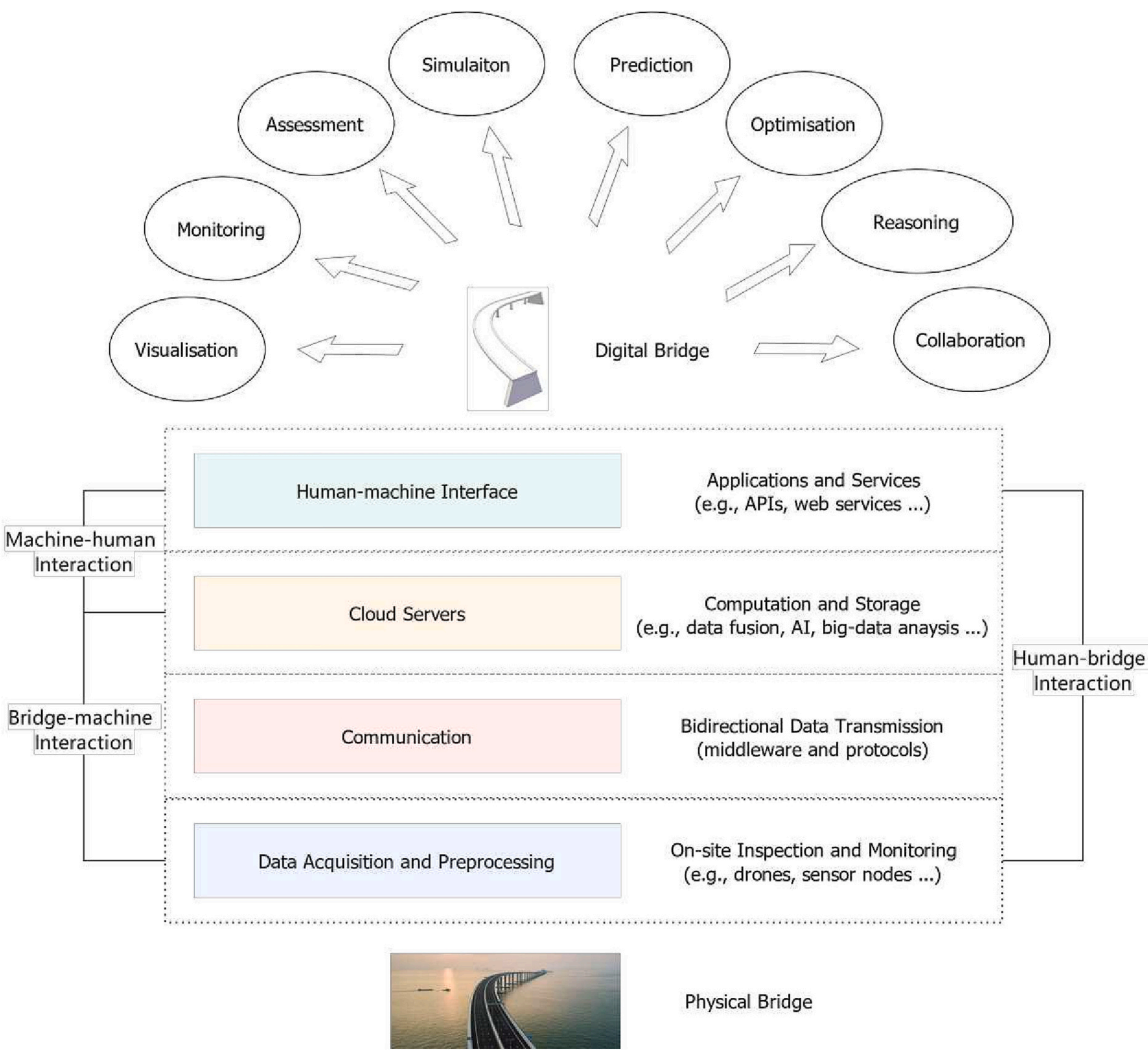

Fig. 1. Current cloud-based Bridge DT Architecture.

2.2. DT related technologies

This part aims to investigate the technologies related to bridge DT, which is useful for the following framework design and prototype development. The content will be organized in the form of a triplet bridge-machine-human interaction, as shown in Fig. 2.

2.2.1. Bridge-machine interaction

Bridge-machine interaction includes various sensing and robotic technologies. Recently, elaborated attached sensors or handheld equipment have been successfully applied on bridges [39] for global structural health monitoring (SHM), including accelerometers [24], strain gauges [35], fiber optical sensors [5], etc., and local non-destructive testing (NDT), such as passive acoustic emission (AE) monitoring [40], active sonar [41] and ultrasonic coda wave interferometry (CWI) detection [42]. Additionally, a high-speed motion camera can be used to monitor bridge vibration and displacements with high accuracy under dynamic loadings based on motion amplification technology [26]. Digital single-lens reflex (DSLR) [43,44] and infrared (IR) cameras [43,44], are also widely used for vision-based bridge inspection to identify surface deficiencies due to their easy integration into traditional inspection workflows. LiDAR (i.e., 3D laser scanning) is another powerful vision-based method as a supplementary to 2D inspection [45] for bridge inspection, which can provide more detailed information such as depth. These sensors and equipment in the regular inspection and real-time monitoring will generate massive heterogeneous data, which is important for bridge condition assessment and decision-making but also puts forward a challenge for DT synchronization.

Traditional bridge inspection with manual access is costly, time-consuming, and even dangerous [4,46,47]. Nowadays, it has got a lot of help from versatile robotic systems, e.g., drones [4,46,47], climbing robots [4,46,47]. Such robotic inspection systems can work with various payloads (i.e., sensors) to access limited areas and better angles, which are difficult or even dangerous for people to reach. Moreover, they are not only carriers of inspection payloads, but they can also improve or assist the inspection with the data from their control system. For example, mobile robots and drones can use the approaches based on Global Navigation Satellite System (GNSS) [48] – i.e., real-time kinematic (RTK) and post-processing kinematic (PPK) positioning, obstacle avoidance system – camera, ultrasonic distance ranger or LiDAR, inertial measurement units (IMUs) to help with defect localization, as shown in Fig. 3. Thus, it will further increase the amount of data for synchronization.

2.2.2. Machine-human interaction

Machine-human interaction includes data communication, storage, common data environment (CDE), and HMI. Transmitting such massive data from bridge-machine interaction requires abundant bandwidth. Therefore, data acquisition in many DTs is based on a wired connection, such as Fieldbus and Ethernet, which can provide fast (up to 10Gbps) and robust data transmission, but it has low scalability and leads to high installation and maintenance costs [49]. Wireless communication is more flexible, which can enable the monitoring of bridges that used to be inaccessible by cables, although it is susceptible to distance and obstacles. The capability of different wireless communication technologies [50–52] is shown in Fig. 4. Short-range wireless communication, such as WIFI, Zigbee and Bluetooth, is only suitable for data collection in situ. Commercial cellular networks operate at a medium range with service costs (charge of data plan). Their bandwidth increases along with frequency bands, i.e., data rate – 5G > 4G > 3G, but their coverage decreases, i.e., distance – 3G > 4G > 5G. Additionally, the low-power wide-area network (LPWAN) technology, which is long-distance wireless communication, includes cellular and non-cellular. Cellular LPWAN, such as NB-IoT and LTE-M, relies on existing commercial cellular networks. In contrast, non-cellular LPWAN, including LoRa, Sigfox, Ingenu, etc., works on free unlicensed industrial, scientific, and medical (ISM) bands. LPWAN is a restricted communication, which works with limited bandwidths (i.e., low data rates), e.g., LoRa (sub-GHz) – up to 50 kbps, NB-IoT – up to 158.5 kbps, GBAN – up to 800 kbps, LTE-M – up to 1Mbps, and is usually bounded by small payload size and duty cycle, such as LoRa and Sigfox, but it also has many advantages, such as long-distance, scalable, and low-cost, which is suitable for bridges in the resource-constraint environment, such as remote areas, and is promising for widespread application.

A CDE is required for bridge DT to store massive heterogeneous data from multiple data sources to support seamless collaboration among different stakeholders across the bridge life cycle, including monitoring data, inspection report, bridge design and construction documents, historical records, rules and standards, inventory, as well as ambient data, such as traffic, weather, air salinity, water speed, and natural disasters. The models and data are conditionally accessed and modified by different users [14]. Meanwhile, each user should have a corresponding priority and resource budget, including time and payment, to avoid potential conflicts with others [35]. It is also worth noting that the required computation and storage resources should not exceed the total capacity of the service provider [35]. BIM with Industry Foundation Classes (IFC) format files is the most popular choice to create such a CDE [53,54] thanks to the consistent and sharable data schema. For example, a dynamic data-driven environment [55] based on IoT-informed BrIM was developed for bridge SHM to support dynamic visualization, seamless updating, long-term monitoring, and data exchange with IFC. Another framework based on BrIM for drone inspection was proposed for data storage and management, which can assign deficiency evaluation to the corresponding element [56].

A virtual platform in computer terminals integrating multiple tools and applications is required to support human-machine interaction [55,57–59]. A few tools have been successfully validated for bridge DT visualization, including Unity [59], Xeokit [60–62], and Visualization Toolkit (VTK) [63]. Docker is recommended for platform deployment as it is less resource-intensive (i.e., multiple containers can share a common kernel) than virtual machines (VMs), and the services can be distributed among a cluster of nodes via Docker Swarm [64]. The Hadoop framework can be used for big-data storage. Meanwhile, NoSQL (such as MongoDB) and NewSQL databases with MapReduce and Spark can

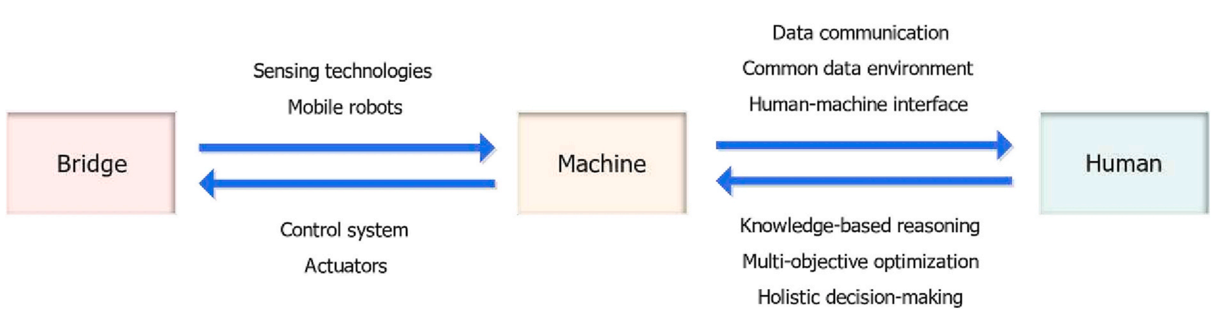

Fig. 2. Triplet bridge-machine-human interaction.

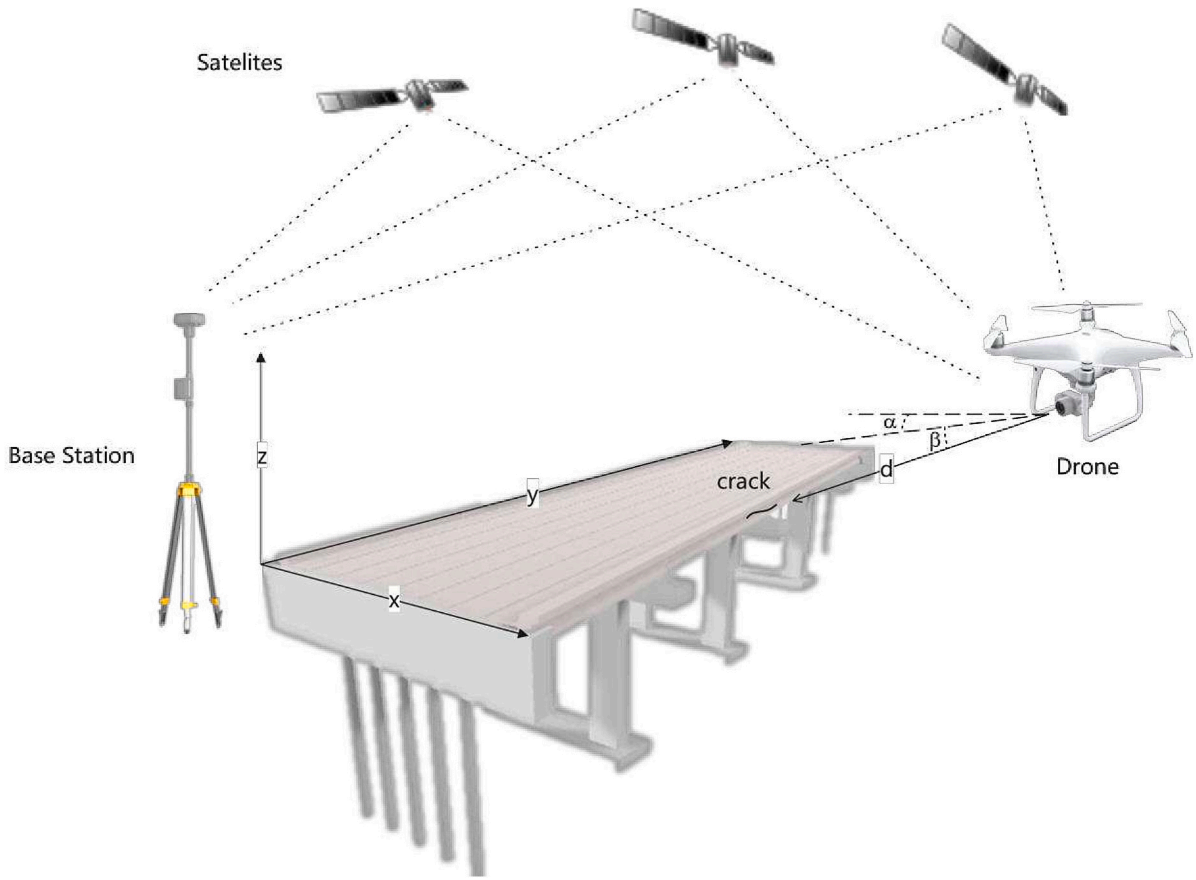

Fig. 3. Defect localization with the drone control system.

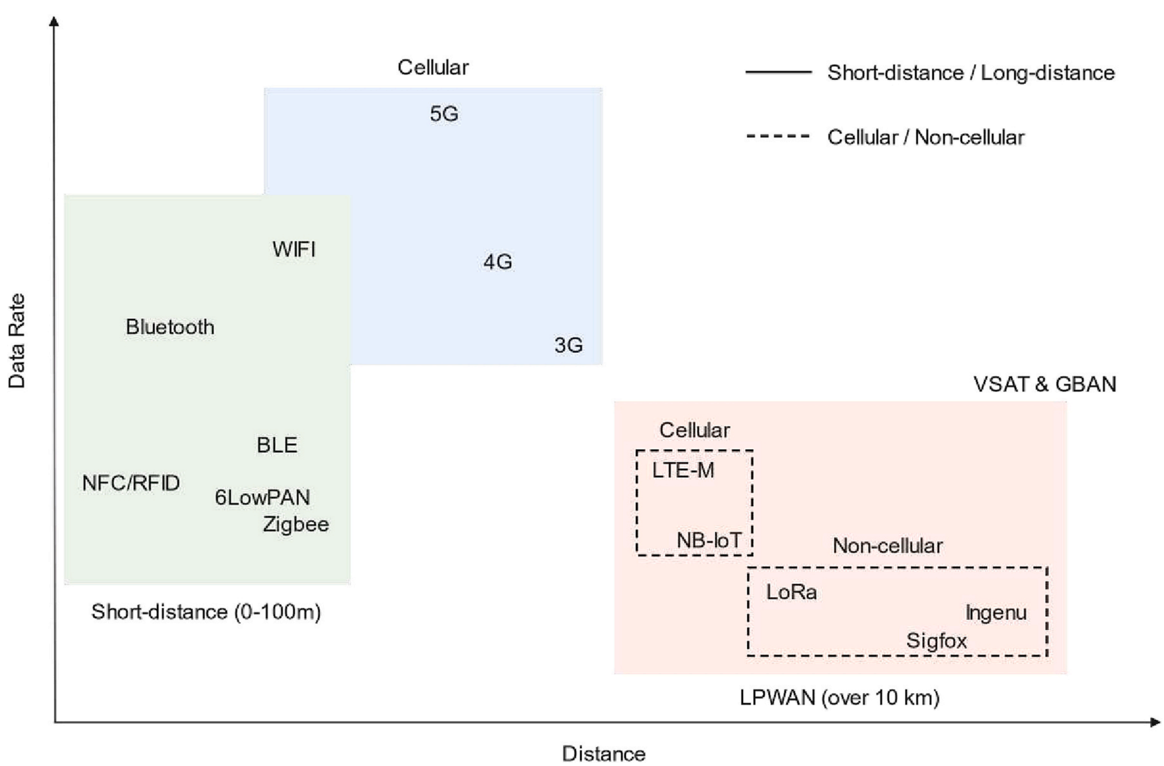

Fig. 4. Capability of typical IoT wireless communication technologies.

accelerate computation for data management [65]. Moreover, the web-based application relying on microservices [27] has been taken as an alternative to a desktop-based application recently due to its lightweight and excellent cross-platform capability. Additionally, extended reality and gaming technologies, such as augmented reality (AR), virtual reality (VR), mixed reality (MR), and Unity engine, are brought in to develop new HMI [66–68]. Emails, text messages, as well as social media (such as Twitter and Facebook), can be also adopted in the platform [69,70].

2.2.3. Human-bridge interaction

Human-bridge interaction aims to make holistic decision-making based on human knowledge or human-designed computational programs. It is reflected in the various DT functions (or services) for bridge O&M, such as condition assessment, maintenance planning, and multi-objective optimization. Traditionally, bridge structural integrity and serviceability are assessed by human engineers based on experience and standards [71], while many data-driven and knowledge-driven approaches are employed in the workflow currently, e.g., data analysis, data mining, machine learning, and knowledge discovery. The data usage in this process can be either indicator-based [72] or the direct use of data in the time or frequency domain [73]. For example, structural damage diagnosis can be achieved through a fully convolutional encoder-decoder architecture using vibration signals from a grid sensor network with high accuracy, to locate damage and classify multiple damage mechanisms [74]. Moreover, bridge deterioration can be predicted based on data mining and knowledge discovery from multi-source data, such as design and construction documents, inspection reports, traffic, weather, disasters, and inventory [19]. Meanwhile, the intervention duration and impact caused by maintenance can also be predicted through a deep neural network (DNN) with embedded entities [8], which can help to make optimal maintenance planning. Additionally, early warnings and protective measures, such as weight restriction, traffic diversion, or even closure, are significant for the safety of physical bridges and users on-site, especially in case of emergency. Such decision-making is also required in the bridge DT, which is usually achieved by knowledge-based reasoning, such as ontology [7] or knowledge graph [7,28,29]. The human-bridge interaction can also enhance maintenance efficiency and save costs both socially and economically.

3. Methodology

As can be seen from 2.2, massive heterogenous data from regular inspection and real-time monitoring of physical bridges has become a challenge for bridge DT synchronization, especially under communication-constraint environments and for large-scale applications. Meanwhile, DT computation and performance are also influenced by the complexity of enormous multi-source data and inevitable system faults (e.g., loss of communication) in terms of time delay and resilience. Although some research [75,76] explored edge computing and federated learning for SHM, which has relatively low communication complexity, they were not developed into a complete and comprehensive bridge DT system. Recently, the concept of Artificial Intelligence of Things (AIoT) has received widespread attention, which is the combination of AI technologies with the Internet of things (IoT) infrastructure to achieve more efficient IoT operations, improve human-machine interactions and enhance data management and analytics [77]. This research aims to study the time delay, complexity, and fault tolerance of bridge DT and reveal how to design a bridge DT to overcome the barriers of data synchronization and communication faults theoretically, as well as develop an AIoT-informed DT communication framework to support bridge O&M with high efficiency, low latency, and excellent resiliency.

3.1. Time delay and complexity

Although there is always a time delay between the physical entity (PE) and the virtual entity (VE) in a DT system, how much of it can be tolerable depends on pre-designed purposes (services) and practical application scenarios. For bridge operation, a short time delay of DT services can enable timely "what-if" analysis and quick emergency response. Specifically, the time delay of DT services, such as early warnings and protective measures, is critical for the safety of physical bridges and the public traveling on the bridges when a disaster is happening or predicted to happen. Meanwhile, a short delay between the physical bridge and the digital-twin bridge can also enhance maintenance efficiency, such as inquiry, inference, and decision-making during the inspection, especially when a complex issue requires big-data analysis or involves multiple stakeholders.

Time delay of DT services consists of communication time T_{comm} and computation time T_{comp}, indicated in Eq. 1. Here, comm and comp are the abbreviations for communication and computation respectively.

$$T_{delay} = T_{comm} + T_{comp} \tag{1}$$

Computation time is directly proportional to computational complexity $O(n)$, which is indicated in Eq. 2 [78]. n is the number of variables. Computational complexity includes time complexity, i.e., the time taken by the algorithm to execute each set of instructions, and space complexity, i.e., the amount of memory consumed by the algorithm [79]. By contrast, the computational time is inversely proportional to the clock frequency of processors when all the processes are sequential, i.e., only one pulse at one time on a single core. For example, suppose an algorithm running on a Y MHz processor takes t seconds to execute, then moves the same algorithm to a Z MHz processor. In that case, the program is reasonably expected to be executed in approximately $(Y/Z) \times t$ seconds [80]. Although it is no longer the case currently due to non-sequential ways (such as multi-core and multi-threading), this relationship is still valid.

$$T_{comp} \propto \frac{Computational\ Complexity}{Clock\ Frequency} \tag{2}$$

Note: I/O and bus time for connecting peripheral devices are negligible.

Communication time is determined by communication complexity, bandwidth, and latency, as indicated in Eq. 3. Here, communication complexity (one-way or multiparty) is the amount of exchanged information (e.g., bits) among PE and VE necessary to perform the computation of certain DT services. The bandwidth is the maximum data transmission speed of specific communication technology. Finally, the latency depends on the distance between communication nodes. Hence, the complexity determines the lower bound of the communication time, especially when communication is restricted with limited bandwidth, such as LPWAN. Therefore, how to reduce communication complexity becomes a critical issue to decrease time delay.

$$T_{comm} = \frac{Communcation\ Complexity}{Bandwidth} + Latency \tag{3}$$

There is a directly proportional relationship between data ambiguity and complexity, i.e., more ambiguity means more complexity. Therefore, the information hierarchy, i.e., the DIKW pyramid, can be leveraged to decrease complexity [81], as shown in Fig. 5. Routine bridge inspection by engineers is a typical DIKW process, which reduces ambiguity by extracting information and knowledge from inspection data based on the engineer's experience. Similarly, structural damage detection involves complexity reduction by obtaining information or knowledge from real-time SHM based on statistical models. Big-data analysis, and machine learning. Moreover, knowledge from human engineers can be transferred to machines by AI (such as supervised or unsupervised learning) and knowledge engineering, thereby achieving the automatic process to reduce complexity in the workflow. For example, agent-based drone inspection can contextualize bridge deficiency and ambient conditions automatically through localization, object detection, and semantic segmentation, instead of human engineers [82].

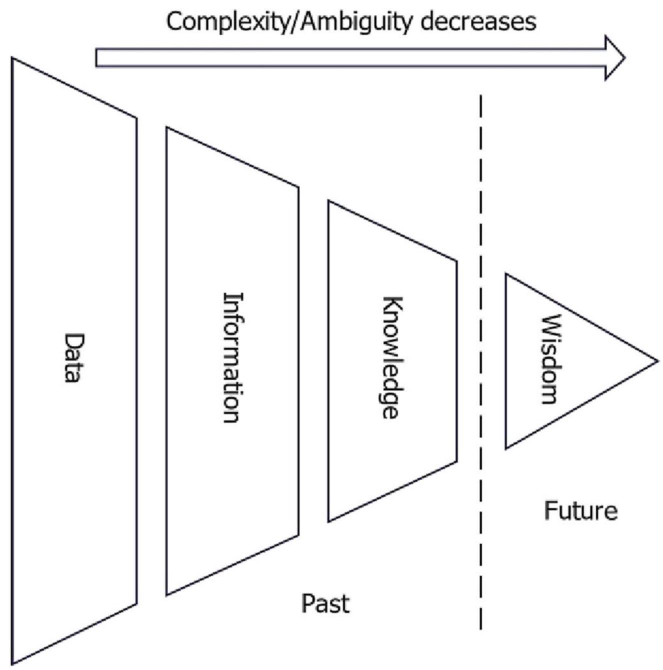

Fig. 5. Complexity/ambiguity decreasing along the information hierarchy.

Edge devices and the DT platform can be taken as two parties with different data sources, e.g., the former includes data collected on the site of physical bridges ($x \in X$). At the same time, the latter involves multi-source data ($y \in Y$) at the cloud server, such as historical records, inventory, weather, and natural disasters. Hence, it brings in an issue that how many bits they need to communicate with each other for computing the function f on $X \times Y$ until one party knows the value of $f(x,y)$ for decision-making. cost(P) is the worst case of bits exchanged (maximum) for a protocol P, which can solve this problem, over all inputs $(x,y) \in X \times Y$. Finally, the communication complexity $D(f)$ is determined with Eq. 4 [83].

$$D(f) = min\{\text{cost}(P) | P(xy) = f(xy) \text{ for all } (xy) \in X \times Y\} \tag{4}$$

Note: x and y are n-bit strings; assume no concerns of computational power.

Theoretically, a communication protocol P can be defined as a rooted binary tree with internal nodes labeled by either E (edge) or C (cloud), as shown in Fig. 6, indicating PE and VE respectively. Each leaf has an output weight w in $\{0,1\}$ (bit exchanging or not). For simplicity, the function $f: X \times Y \to \{0,1\}^n$, encoded as a finite sequence of zeros and ones. f_v is associated with node v (if v is labeled by E, $f_v(x) \to \{0,1\}^m$, and vice versa). The bits at node a are sent by E with the calculated value $f_a(x)$, which is a binary string. The number of bits transmitted to C is $w_a \cdot (log_2(f(x)) + 1)$. Therefore, the number of bits exchanged to compute $f(x, y)$ in the path of $a \to b \to c \to d$ is $log_2(f_a(x)) + log_2(f_b(y)) + 1$.

There is always a simple protocol that sends all $x \in X$ to Y, performing the same as one-way communication, in which $D(f) = O(n)$. This way is the best one can do for the equality function (EQ), which outputs one if $x = y$. However, for other tasks requiring bidirectional communication, i.e., f computation is achieved by both parties, communication complexity can be decreased by a few methods, such as edge computing, so there is $D(f) \leq O(n)$. For example, for a parity function $\oplus_{2n}(x,y)$, the best way is to send $b = \oplus_n(x)$ to y, then calculate $b \oplus (\oplus_n(y))$, in which $D(f)$ is only 1 bit.

In practice, many tasks do not need to always consume the maximum bits exchanged under the worst case. For example, for the structural assessment process (query problem) based on a decision tree (shown in Fig. 7), its cost(P) follows the longest path of the tree. However, many non-severe defect assessments do not need to cost as much complexity as cost(P), so we can properly use hierarchical or interactive data exchange to reduce the practical communication. Furthermore, communication complexity can also be reduced by turning deterministic communication complexity $D(f)$ into randomized communication complexity $Pr[R(x)]$ even in one-way communication, e.g., only update the changing data beyond a threshold θ to the DT for synchronization as long as θ can meet the precision requirement of services.

Moreover, as T_{Delay} is comprised of $T_{communication}$ and $T_{computation}$, the sequence of communication and computing can also have a distinct impact on T_{Delay} of DT services, even if communication and computation complexity is constant, i.e., with a certain protocol and algorithm, as shown in Fig. 8. Such impact can be amplified when either of $T_{communication}$ and $T_{computation}$ plays a dominant role in the process, e.g., communication protocols are quite restricted, such as LPWAN. For such a case, which is the research objective in this work, the total time delay T_{Delay} can be reduced significantly with appropriate edge computing before communication, as shown in Fig. 8.

3.2. Fault tolerance and topology

Fault tolerance is the property that enables a system to continue operating correctly in the event of a failure due to one or more faults of its components. The fault-tolerant capability allows DT to continue its intended services, possibly at a reduced level, rather than failing when some part of the system fails [84]. Currently, most bridge DTs offer data storage and analysis on the cloud (Fig. 1), but the services will fail when cloud servers become unresponsive, such as under a temporary loss of communication. If it occurs when a disaster is happening or is predicted to happen, it might be fatal to the physical bridge and the users on-site, so it is necessary to develop a bridge DT with the required fault-tolerant capability. Edge computing within AIoT can also help to enhance the

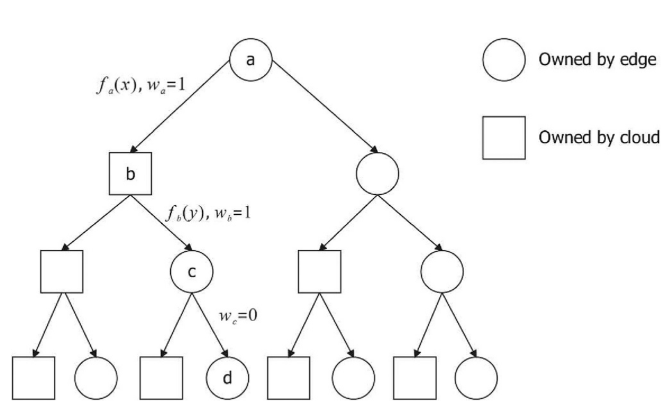

Fig. 6. A rooted binary tree for computing $f(x,y)$ through two-way communication.

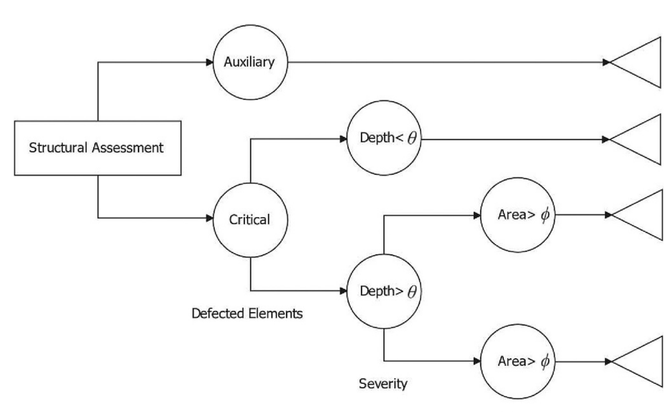

Fig. 7. Bridge structural assessment based on a decision tree.

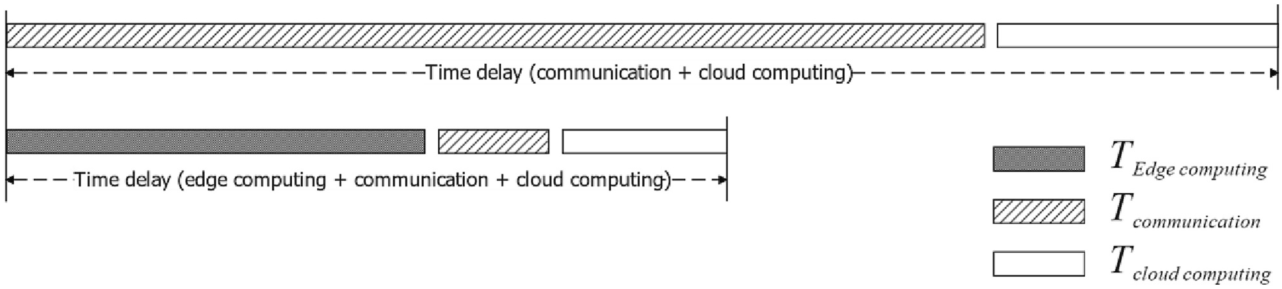

Fig. 8. Edge computing reduces time delay significantly when $T_{communication}$ (dominates)

current bridge DT system resilience significantly. For example, even if communication between the edge and cloud breaks down, the edge devices can still make an AI-based decision to take an appropriate and timely measure via bridge or transport control systems as a response, e.g., load restriction, traffic diversion, or even closure. Such edge computing can be designed on base stations, gateways, embedded systems, or sensor nodes, depending on task complexity. The closer to the physical bridge, the more resilient the design will be.

Besides, the communication topology is also significant to system resilience. A decentralized mesh network (even partial mesh) has better fault tolerance than a centralized star network, as shown in Fig. 9. As presented in 2.2.2, non-cellular LPWAN is suitable for remote bridge management thanks to its decentralized transmission mode and long-distance coverage (over 10 km in rural areas), such as LoRa. Therefore, non-cellular LPWAN with a mesh network can be utilized for local communication on the site of physical bridges and integrated into the cloud-based DT architecture to enable fault tolerance for a temporary loss of communication. Meanwhile, by combining AI-based edge computing, the edge devices and control system of physical bridges can perform as a self-adaptive subsystem when the cloud servers become unresponsive, and their performance can be simulated and predicted at the cloud level. Hence, the whole bridge DT can continue to work at a reduced level without complete failure. Finally, the physical bridge and digital-twin bridge can re-synchronize after the restoration of communication.

3.3. Mathematical idealization

A physical bridge, i.e., PE, can be described as a discrete system in Eq. 5. x_t and x_{t+1} are the states of PE at time t and $t+1$. u_t is the input variable at time t. y_t is the observed variable from IoT sensors. e_t is the error of the measurement.

$$x_{t+1} = f(x_t u_t),\ y_t = h(x_t, u_t, e_t) \tag{5}$$

Here, $\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_y}$. The edge-computing model was trained with the observed variable $y \in \mathbb{R}^{n_y}$, so the inference can be represented as $v_t = g(y_t)$. v_t is the result to support decision-making, i.e., to calculate the input variable u_{t+1} at time $t+1$. Therefore, $u_{t+1} = d(u_t, v_t)$.

If such computing is taken on the cloud, i.e., x_t, y_t and u_t are transmitted to VE, it can leverage powerful computational capability and massive multi-source data ($\varphi \in \mathbb{R}^{n_\varphi}$) in the cloud, which can help to make more precise and holistic decisions. Therefore, there is $V_t = G(y_t, \varphi_t)$.

However, it will also bring in the time delay T_{delay} between edge and cloud, i.e., PE and VE. Given bi-directional time delay in both uplink and downlink, the input variable u derived from cloud computing has a lag of $2d$ behind the result based on edge computing, i.e., $u_{t+2d} = D(u_t, V_t, \theta)$. $\theta \in \mathbb{R}^{n_\theta}$ is the set of model parameters for prediction to offset the time delay. d is the ratio of T_{delay} to the measurement period T_{period}. Therefore, the cloud-based bridge system considering the time delay between PE and VE can be represented as Eq. 6.

$$x_{t+2d+1} = F(x_t u_{t+2d}),\ y_{t+2d} = h(x_{t+2d}, u_{t+2d}, e_{t+2d}) \tag{6}$$

Therefore, edge computing is suggested for quick analysis and response due to its low latency, such as under emergent circumstances. Moreover, edge-based algorithms also pursue low computational complexity in practice, e.g., $u_{t+1} = s(v_t)$ in edge computing is usually based on straightforward "what-if" analysis or fuzzy control rules. In contrast, cloud computing is recommended for holistic decision-making with big-data analysis from multiple sources to achieve a long-term maintenance strategy for the bridge.

For one-way communication, i.e., sending all the three variables x_t, y_t, u_t to the cloud, the communication complexity $D(F) = O(x) + O(y) + O(u)$, where the variables have x bits, y bits, and u bits respectively. Most of the time, the communication time in and among edge devices on a single physical bridge can be negligible.

Furthermore, the total computational resources required by tasks, which are running on the edge device, should not exceed its processing capability [22], indicated in Eqs. 7 and 8.

$$C_{edge}^{processor} > \sum task_i^{processor} \tag{7}$$

$$C_{edge}^{RAM} > \sum task_i^{RAM} \tag{8}$$

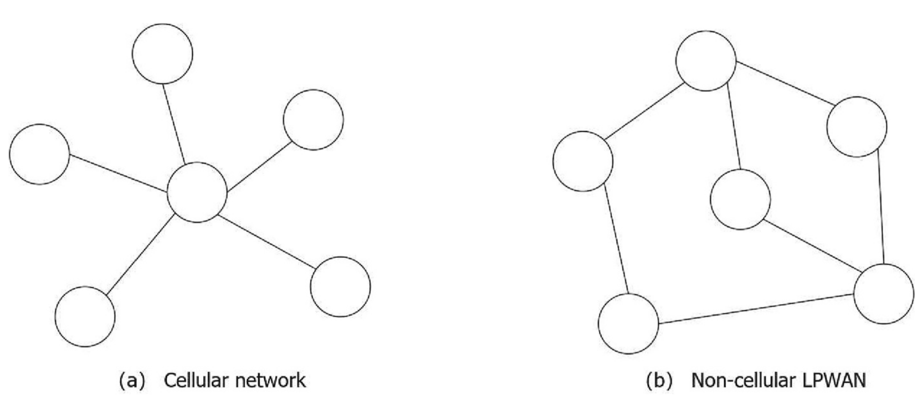

Fig. 9. (a) Cellular network based on star network (b) Non-cellular LPWAN based on the mesh network.

3.4. Petri-net modeling

The bridge DT can be taken as a discrete event dynamic system with distributed architecture. Therefore, it can be described with a Petri net (PN), i.e., a bipartite, weighted, and directed graph (digraph) that has been proven to be an efficient tool for the modeling, analysis, and control of a discrete event system (DES) [85]. A stochastic time PN can help to demonstrate system resilience and time delay, as well as to trace data flow and sources in the system for DT services.

A PN comprises two kinds of nodes, named places, and transitions, with arc connections either from a place to a transition or a transition to a place. In the graphical representation, a place $p \in P$ is drawn as a circle, representing a condition (a particular state of the system). In contrast, a transition $t \in T$ is drawn as a box, representing an event (dynamic activity). Arcs are labeled with the corresponding weights $w \in \mathrm{W}$ ($w = 1$ by default). Places are visited by k tokens, representing data items moving through a PN, which is like communication packages. Marking M is a raw vector that refers to the distribution of tokens throughout the PN at a specific time, indicating the state of the PN. The firing rules indicate when and how tokens are created and destroyed in a new marking M'. More details about Petri nets can be found in [86,87].

Mathematically, a stochastic time PN can be described as below:

$$N = (P, T, A, \mathrm{W}, \Lambda, \Theta)$$

Where P is the finite set of places, $P \neq \phi$; T is the finite set of transitions, $T \neq \phi$; $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs; $W : A \rightarrow \{1, 2, 3 \ldots\}$ is the weight function on the arcs; $\lambda \in \Lambda$ is the firing rates associated with transition, which are related to probabilities of successful communication; $\theta \in \Theta$ is the time elapsed in state transition.

Here, the firing rules are summarized as follows:

1. Transition t_i consume the tokens from each available input arc and generate $\omega_{i,\ output}$ tokens at each output arc. $\omega_{i,\ output}$ is the weight on the output arc.
2. Transition t_i is enabled if the input place p_j has at least $\omega_{j,\ input}$ tokens. $\omega_{j,\ input}$ is the weight on the input arc.
3. An enabled transition t_i fires according to the firing rate λ_i. With probability R, it can be expressed as below:

$$t_i = \begin{cases} 1\ (\mathrm{R} = \lambda_i) \\ 0\ (\mathrm{R} = 1 - \lambda_i) \end{cases}$$

4. After firing, transition t_i removes all the tokens in the place p_i and add $w_{i,\ output} \cdot t_i$ tokens into the next place.

The proposed framework is modeled in a PN, which has seven places (p_1, p_2, $\ldots p_7$) and six transitions (t_1, t_2, $\ldots t_6$), as shown in Fig. 10. The starting place p_1 holds the token of the observed variable y from PE, i.e., physical bridge. t_1 and t_5 are taken on edge devices, such as sensor nodes, embedded systems, and gateways. t_2 and t_4 are communication between PE and VE and fired according to the firing rates λ_2 and λ_4. t_3 is taken on cloud servers. p_4 holds the token of the variable ν collected from multiple resources in the cloud. Finally, the decision-making variables from edge computing and (or) cloud computing converge at p_7 for the control or adjustment of PE to generate a new token of the observed variable y'.

As can be seen from the PN, there is an edge-based loop $t_1 \rightarrow t_5$ and a cloud-based loop $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4$. Here, assume the edge loop working as E and failure as E', and so does the cloud loop, i.e., C and C'. Therefore, the edge loop working probability is $R(E) = \lambda_5$ and the cloud loop working probability is $R(C) = \lambda_2 \cdot \lambda_3 \cdot \lambda_4$.

If both loops (after t_1) are mutually independent, the results can be divided into three categories: a) the result is determined with both edge and cloud loops, wherein the probability is $R(E) \cdot R(C)$; b) determined with either edge or cloud loop, wherein the probability is $R(E') \cdot R(C)$ or $R(E) \cdot R(C')$ respectively; c) system failure, wherein the probability is $R(E') \cdot R(C')$. Because of $0 < \lambda_i < 1$, $R(C') - R(E') \cdot R(C') = \lambda_5 \cdot (1 - \lambda_2 \cdot \lambda_3 \cdot \lambda_4) > 0$. Therefore, the system failure probability decreases by adding the edge loop. Moreover, to achieve a significant enhancement of system

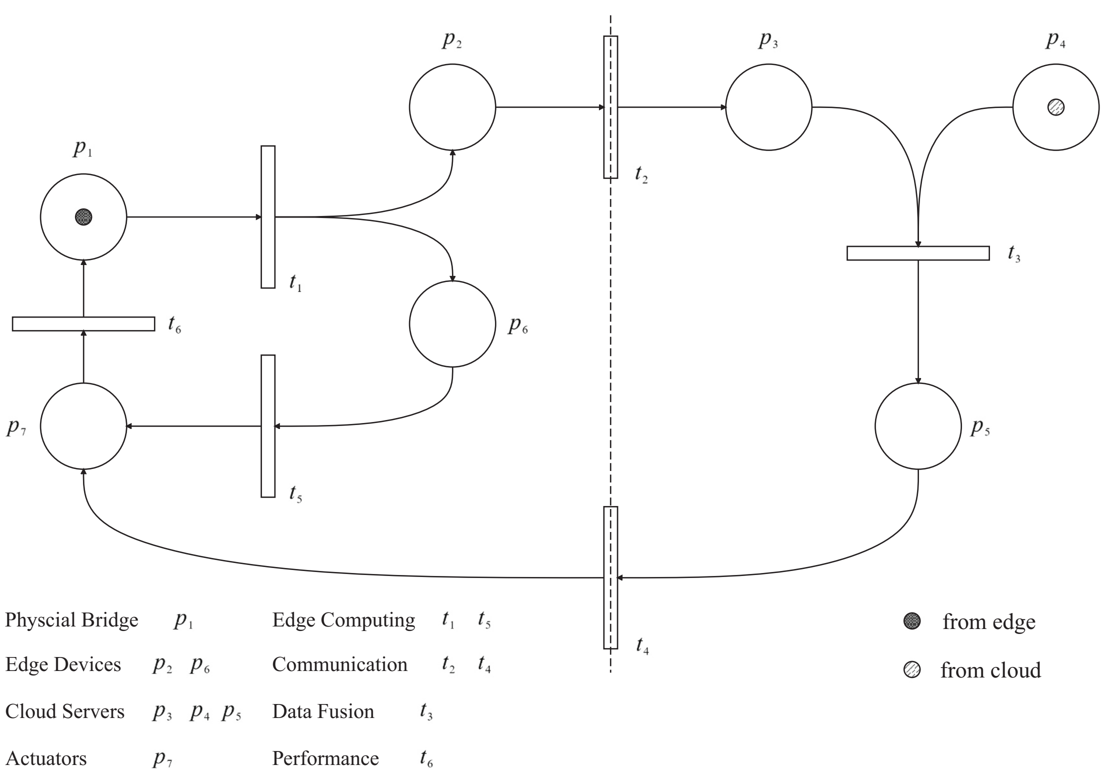

Fig. 10. Petri-net modeling for a bridge DT system initialized from M_0

robustness, λ_5 should be much greater than $\lambda_2 \cdot \lambda_3 \cdot \lambda_4$.

1) If both loops (after t_1) are mutually exclusive with an extra rule that t_5 only fires when t_2 or t_3 or t_4 does not fire, i.e., the edge loop is only enabled when the cloud loop fails. Thus, according to conditional probability, there is $R(C') - R(E'|C') = R(E|C') = (1 - \lambda_2 \cdot \lambda_3 \cdot \lambda_4) \cdot \lambda_5 > 0$. Therefore, the system becomes more robust by adding the edge loop in this case.
2) Moreover, the total time elapsed in a path can be calculated by summing all the elapsed times in the firing schedule [87]. For the edge loop, the time elapsed $\theta_{edge} = \theta_1 + \theta_5$, while for the cloud-based loop, the time elapsed $\theta_{cloud} = \theta_1 + \theta_2 + \theta_3 + \theta_4$. Hence, the difference is $\Delta = \theta_2 + \theta_3 + \theta_4 - \theta_5$. Here, θ_2 and θ_4 are communication time elapsed, which θ_3 and θ_5 are computational time at the cloud and the edge. Hence, a trade-off between computing and communication complexity is required to guarantee the system's performance (as explained in Fig. 8).

4. Framework design and development

A bridge DT is not a single technology but a cross-platform integration of devices, communication middleware, and software packages, and expected to perform a highly automated pipeline with little human intervention, including data acquisition, machine-to-machine (M2M) communication, back-end services, and physical-world response. This works aims to design a specific DT framework to support bridge O&M under the communication-constraint circumstance. To this end, an AIoT-informed bridge DT framework with high efficiency, low latency, and excellent fault tolerance is proposed as shown in Fig. 11, which integrates AI-based edge computing, LPWAN communication, cloud servers for storage and processing multi-source data, MQTT protocols, and a web-based interface. Furthermore, the framework is developed to the level of a prototype based on cross-platform integration. This part presents the framework design and prototype integration step by step, as well as the logic behind each section.

4.1. AI-based edge computing

Firstly, agent-based drones and robots, as well as versatile sensor networks are leveraged for bridge regular inspection and real-time monitoring. AI-based edge computing can enable them with autonomous capability for preliminary analysis and decision-making, such as damage detection, bridge assessment, and early warnings.

An advantage in this way is that edge computing can reduce data complexity significantly by converting them to advanced information or knowledge according to information hierarchy (i.e., DIKW), as shown in Fig. 12. This can enable bridge DT to satisfy the requirement of restricted communication, i.e., LPWAN. The derived information or knowledge can be transmitted to cloud servers to either achieve DT services directly (such as visualization) or join the bipartite interactive computing for function F (see 3.1 and Eq. 6) – due to different data sources owned by edge and cloud. It can reduce the time delay of cloud-based DT services significantly with appropriate sequential design (as Fig. 8). For example, in the drone-enabled bridge inspection, the extracted semantic information based on deep learning and computer vision, such as defect location and severity, can be synchronized to the bridge DT quickly and conveniently through LPWAN, instead of defect images, point cloud, etc. This method can enhance maintenance efficiency through DT services such as historical query, big-data analysis for defect causes, and optimization for inspection and repair.

Another advantage is that preliminary decision-making based on edge computing can provide the quickest response and can still perform the task even if cloud servers become unavailable. For example, the detected bridge damage from a sensor network, which has influenced bridge serviceability, can trigger weight restriction, or even closure through actuators and monitors immediately. Moreover, it can also transmit the derived information to cloud servers or adjacent bridges for collaboration, such as traffic diversion.

Notably, edge computing can be taken on different roles in architecture, such as sensor nodes or gateways, as well as various edge devices equipped with robotic agents or attached on physical bridges, including field-programmable gate array (FPGA), microcontrollers, single-board computers, etc. The power can be supported by batteries, sustainable power supply, and energy harvest [88]. The edge-computing tasks should not exceed the device's computational capability (Eqs. 7 and 8) and power restriction. Meanwhile, the data owned by the edge is limited to the local physical bridge and environment, so edge computing only aims to provide preliminary analysis and decision-making. Moreover, to achieve high resilience, the edge device for decision-making is usually supposed to be placed as close to physical bridges as possible (explained in 3.2), but it can also be designed to hierarchical architecture for different-level tasks, which is going to be discussed in the next section.

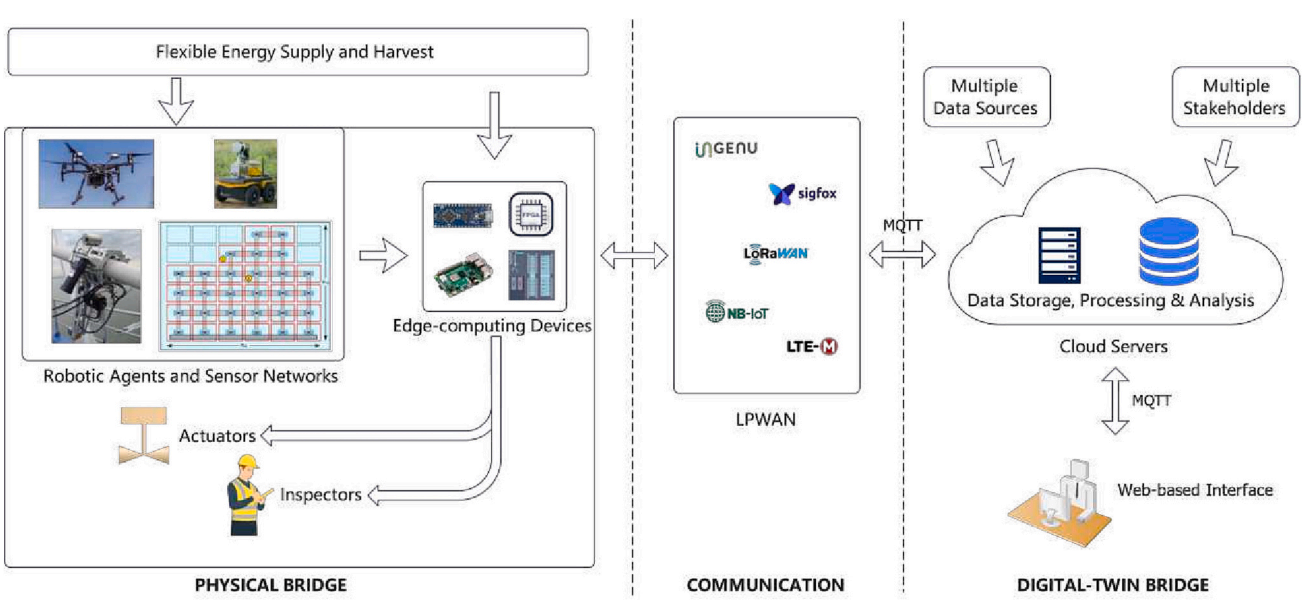

Fig. 11. Proposed AIoT-informed DT framework to support bridge O&M in communication-constraint environment.

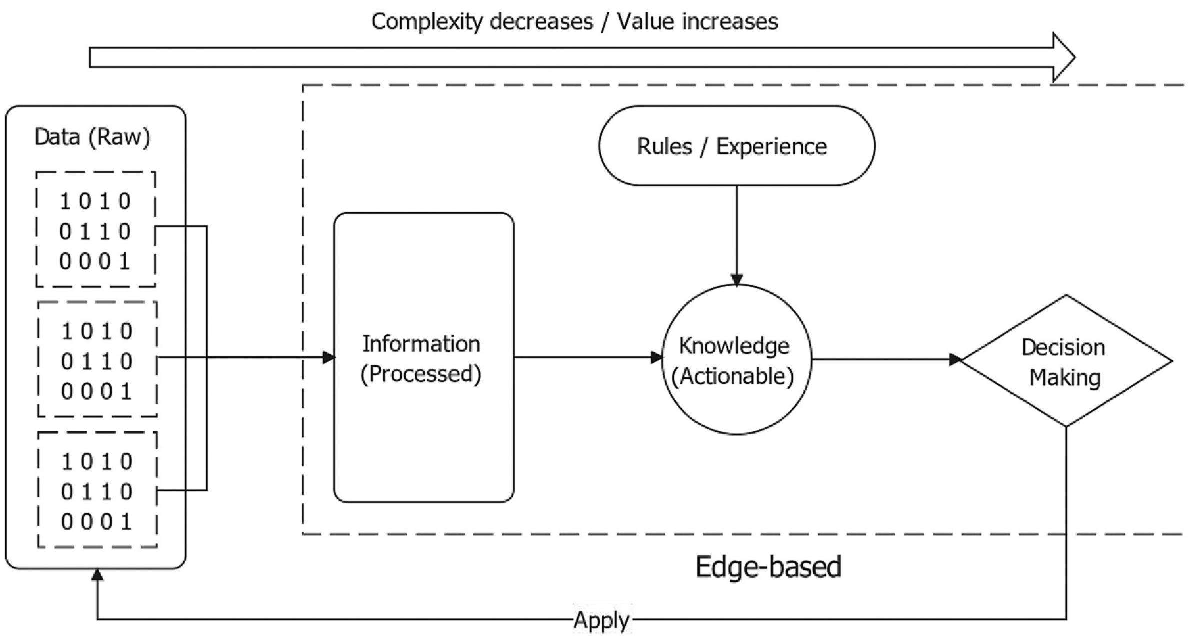

Fig. 12. The edge loop according to DIKW to reduce complexity.

4.2. LPWAN communication

As reviewed in 2.2.2, LPWAN is a set of low-power long-distance communication technologies. As claimed [50–52,89,90], they can sustain a long-term work of up to several years for battery-operated sensor nodes and over 10 km coverage in rural areas, which is beneficial to the connection for bridges in resources-constraint environments. However, LPWAN is restricted by its low data rates, duty cycle, etc., so it is not available for transmission of the massive heterogenous data in the regular inspection and real-time monitoring, e.g., images, and point clouds. Thus, AI-based edge computing is designed to reduce complexity significantly (see 4.1), by converting data to advanced information or knowledge according to DIKW. The derived information can be organized in a code system with a predefined protocol [91]. Moreover, appropriate compression methods can be also applied before transmission to achieve more efficient transmission, such as run-length encoding (RLE), which is a lossless compression approach especially suitable for binary images.

Moreover, non-cellular LPWAN does not require commercial base stations and can take advantage of free ISM bands, which provides an option for flexible and low-cost communication to create bridge DTs. It can be utilized in a hierarchical communication architecture designed for different-level tasks, wherein sensor nodes, gateways, and cloud servers are organized from bottom to top. Gateways can manage multiple sensor nodes and embedded systems (e.g., multiple bridges and robotic agents) in an area through LPWAN simultaneously and can communicate with cloud servers as well as with each other. This enables bridge DTs with fault tolerance by combing edge-based AI for normal functions when cloud servers become unavailable temporarily (resilience). Furthermore, sensor nodes are designed to not only communicate with gateways but also connect to each other via non-cellular LPWAN, which can guarantee preliminary decision-making and quick response on physical bridges even if gateways break down (resilience). In the prototype, LoRa is selected for DT communication to support bridge O&M. The complete hierarchical communication architecture is shown in Fig. 15.

4.3. Cloud and protocols

Cloud servers are designed to provide a CDE for data from multiple sources, including not only edge-based data from regular inspection and real-time monitoring but also cloud-based data such as design and construction documents, historical records, inventory, traffic, weather, and disasters. Both relational and non-relational databases and tools, such as MySQL and MongoDB, are utilized to store and manage such heterogeneous data and information. Meanwhile, cloud computing is another crucial function. The huge and sophisticated models for complex DT services, e.g., structural analysis, deep learning, multi-objective optimization, and holistic decision-making, which require powerful computational capability, are deployed on cloud servers within dockers. Such models and data can be conditionally accessed and modified by different stakeholders. The results can be sent back to end devices on physical bridges through the downlink of LPWAN for performance.

The proposed information or knowledge from cloud servers can be transmitted to HMI through HTTP or MQTT protocols. In the prototype, the TTN (The Things Network) server and the desktops in the lab are integrated as cloud servers to support the web-based platform. The bidirectional communication among cloud servers and the interface is achieved with MQTT protocol thanks to its lightweight and fast transmission capability. Specifically, it can be implemented with “subscribe” and “publish” between brokers and clients using Eclipse Mosquito or Node-RED, as shown in Fig. 13.

4.4. Web-based platform

The bridge DT platform is designed to perform through web services based on RESTful architecture. It has a user-friendly interface, which can enable users to access the available information according to their permissions. In the prototype, Cesium and Xeokit are both employed in the platform for the geographic information system (GIS) and BIM respectively, as shown in Fig. 14. Node.js and npm-anywhere (a static file server) are employed to support the web-based interface. In the Cesium, the bridge location, traffic conditions, ambient situation (such as weather and tides), and project description are displayed, wherein 3D tiles are utilized for bridge visualization. In the Xeokit, IFC models are utilized to enable users with the capability of manipulating each element of the BIM model. Then the bridge condition, such as defect location and severity, bridge serviceability, and structural assessment, are displayed. The web-based click event can be employed to trigger various DT services, such as query, knowledge-based reasoning, and maintenance

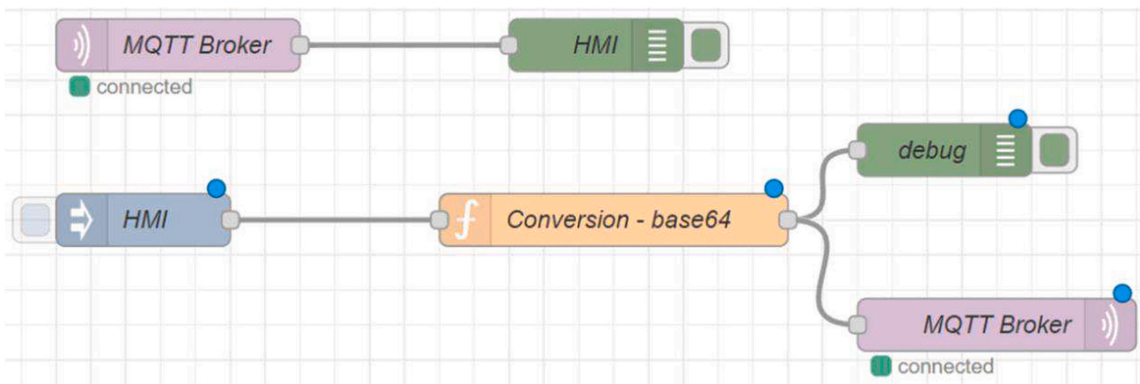

Fig. 13. MQTT implementation between cloud servers and HMI.

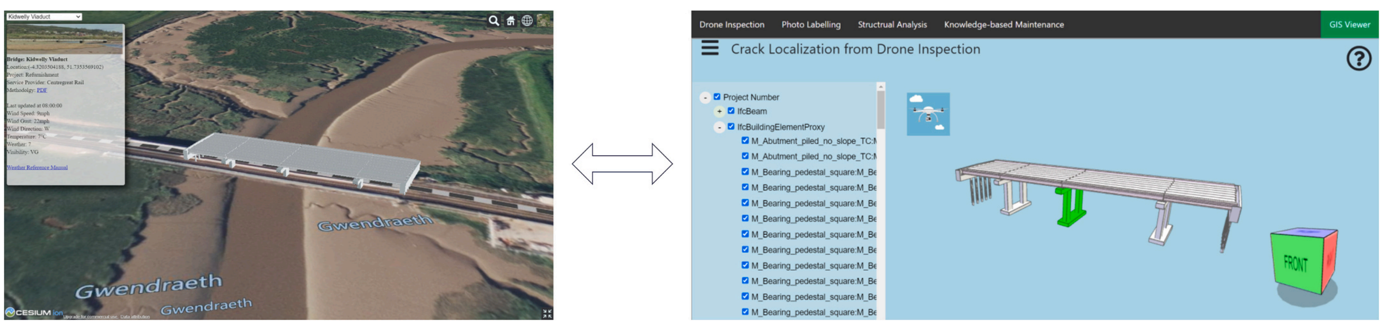

Fig. 14. Platform interface switch between Cesium and Xeokit.

planning. Such a platform can enable users to interact with entities and nodes in VE and PE, e.g., to reconfigure and reboot the edge device in the VE remotely from the platform if the device is registered with a fault.

Finally, following the above sections, the proposed AIoT-informed framework and its cross-platform prototype are shown in Fig. 15. The framework and prototype have high efficiency and low latency, which are achieved through AI-based edge computing. Moreover, edge computing and the resilient hierarchical communication architecture with cellular and non-cellular LPWAN can enable PE with the capability of preliminary data analysis, decision-making, and quick response, even under the temporary loss of communication, which can guarantee the system fault tolerance. Especially, the PE (physical bridges) with edge devices can perform as a self-adaptive subsystem when cloud servers become unavailable. Meanwhile, their performance can be predicted by cloud-based VE using simulation or machine learning. Then PE and VE will synchronize again after communication recovery.

5. Proof of concept – framework validation

In this section, firstly the proposed framework is validated under three different cases during bridge O&M to demonstrate its functionality, i.e., 1) drone-enabled bridge inspection; 2) vibration-based bridge monitoring; 3) dynamic evacuation when cloud servers become unresponsive. Then a comparative analysis between the proposed framework and the previous bridge DTs is conducted.

5.1. Drone-enabled bridge inspection

This experiment aims to synchronize the sufficient defect characteristics to the bridge DT during drone flight in a communication-constraint environment (e.g., LoRa) to update the bending stiffness (BS) reduction coefficient for structural assessment instead of taking large numbers of inspection images back to the office in the traditional workflow or bring them to another place with excellent communication for synchronization. This enables on-site inspection and back-end DT services to collaborate simultaneously, e.g., structural assessment, historical query, in-depth inspection, mechanism analysis, and even instant repair, thereby enhancing maintenance project efficiency.

The AI-based processing is taken on a Raspberry Pi 4 Model B, which can be utilized as a drone on-board computer or a controller in situ. As a prerequisite, the drone needs calibration before inspection using a chessboard at different distances and angles from the lens, to obtain the explicit proportional scale between pixels and actual length (or width). Deep convolutional neural networks (DCNN) can enable drones with automatic defect-detection ability, which has been widely accepted and commonly used in image-based defect detection. Here, a dataset created for bridge crack detection [92] is selected for the experiment, including the 2011 background and 4058 crack images (224 × 224). To accelerate the on-board process, the images are resized to 32 × 32 and trained with a simplified LeNet-5 (fewer parameters) through TensorFlow on the Google Codelab, i.e., train-validation-test split – 60%:20%:20%; optimizer – stochastic gradient descent (SGD); learning rate – 0.001; batch size – 128. The training process and model performance in the test set are shown in Fig. 16 and Table 1. Then the model is converted into a specific version for tinyML through TensorFlow Lite, which is especially suitable for deep-learning model deployment on microcontrollers and embedded systems with improved efficiency. The crack identification for one image is <1 s in the experiment.

Subsequently, detected crack images are segmented into binary images with background and crack through image processing (OTSU thresholding and morphological operations). Then the crack characteristics can be calculated statistically, such as H_{crack}, W_{max} *and* L, as shown in Fig. 17, by combining the distance from the lens to the objective surface (measured with an ultrasonic ranger or a laser scanner, etc.)

Crack orientation can be determined with the camera angle and the flight attitude from IMUs, such as transverse or longitudinal. Furthermore, the defect (e.g., crack) can be localized in the bridge coordinate by combing GNSS positioning (such as RTK or PPK), IMUs, and distance

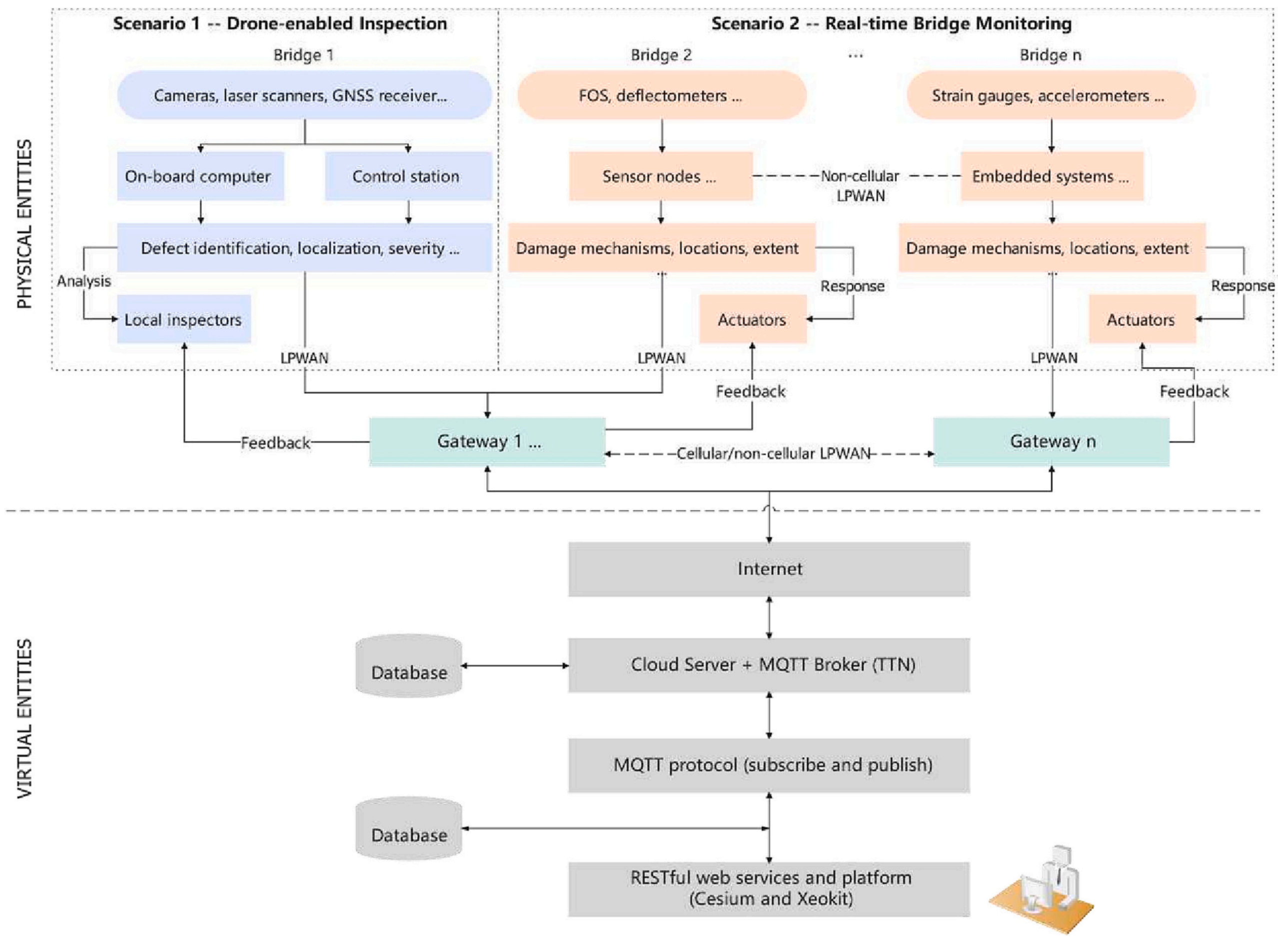

Fig. 15. Proposed framework and developed prototype for bridge DT.

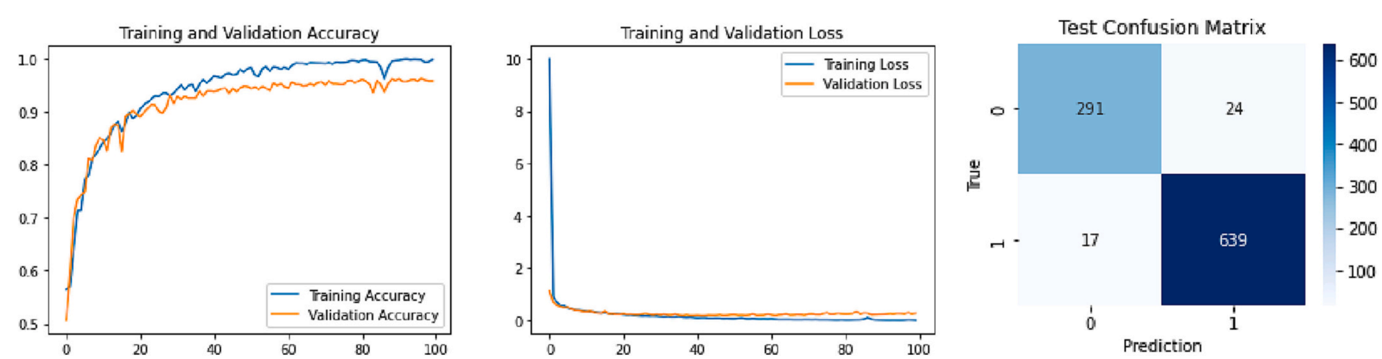

Fig. 16. Training and test for crack detection.

Table 1
Model performance evaluation.

Model	Accuracy	Precision	Recall	F1 score
LeNet-5	0.95	0.95	0.91	0.93

ranger during flight, as shown in Fig. 18 [48]. Given the situation without stable GNSS signals, computer vision (such as bridge element recognition), IMUs, and distance rangers can be leveraged for both drone positioning and defect localization through Kalman Filter.

Given the earth's ground can be taken as a plane within just a few-kilometer distance, D_{drone} can be calculated as Eq. 9. The defect co-ordinates can be calculated as Eqs. 10, 11 and 12. Then the coordinates can be further linked to the precise bridge element, such as the beam, deck, and pier, according to geometric information.

$$D_{drone} = R\sqrt{(\phi_2 - \phi_1)^2 + (\lambda_2 - \lambda_1)^2} \tag{9}$$

Where, L – horizontal distance; R – earth's radius (the parameter that needs to be calibrated); φ_1, φ_2 – base and drone latitude; λ_1, λ_2 – base and drone longitude.

$$X_{crack} = D_{drone} sin\theta - D_{crack} cos\beta cos\alpha \tag{10}$$

$$Y_{crack} = D_{drone} cos\theta - D_{crack} cos\beta sin\alpha \tag{11}$$

$$Z_{crack} = H_{drone} - H_{base} + h_{equip} - D_{crack} sin\beta \tag{12}$$

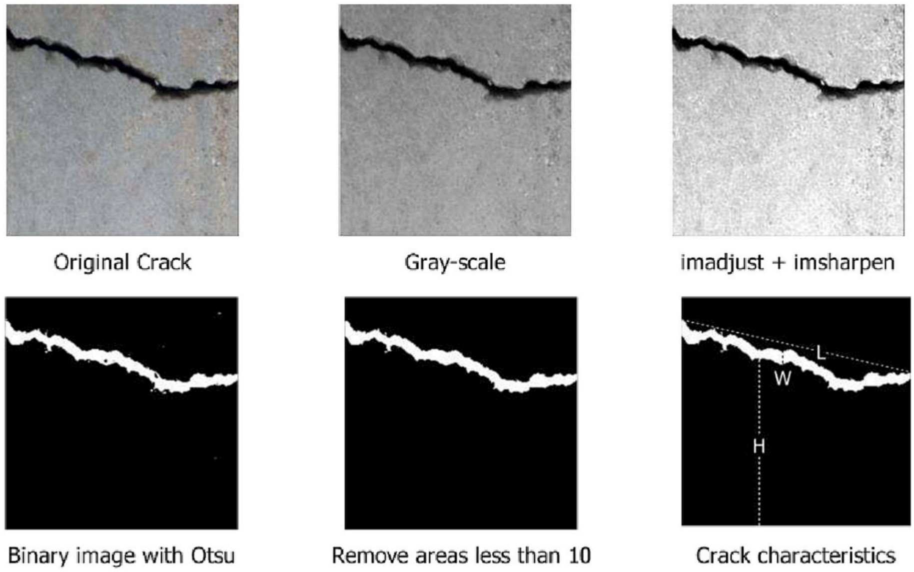

Fig. 17. Crack segmentation through image processing.

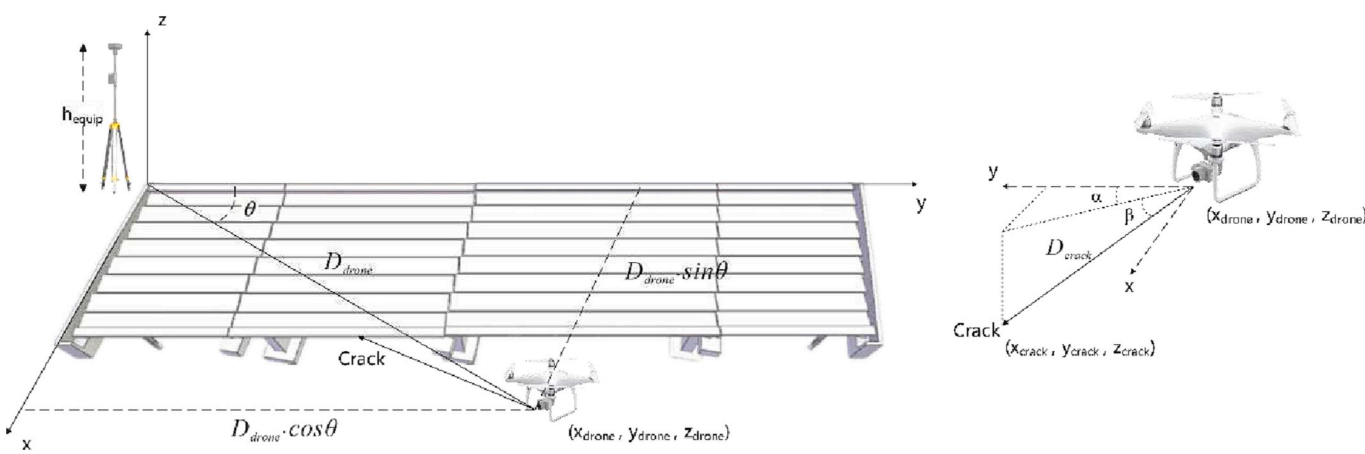

Fig. 18. PPK approach for defect localization in the bridge coordinate system.

The AI-derived critical information for structural assessment, i.e., defect types, characteristics, coordinates, and orientation, can be encoded as Fig. 19 with three significant digits (i.e., a single float) for transmission. It can satisfy the strictest payload requirement of LoRa (at SF12/125 kHz) with the maximum range in EU868. The calculated airtime is 2.138 s, which can achieve synchronization in near real-time.

BS reduction coefficient β is calculated as $\beta = K_i/K_1$, where K_1 is the initial stiffness of the beam in the elastic stage and K_i is the ith loading. The relationship of β to W_{max} and H_{crack} can be derived from the test on the specimen as Eq. 13 [93]. Therefore, β can be calculated using the synchronized crack information, so that the BS reduction can be assessed successfully.

$$\beta = \{f(W_{max})\ g(H_{crack}/h)\}_{min} \tag{13}$$

Where h is the beam height.

Sometimes, defect profiles are unusual, such as the crack shown in Fig. 20. The segmented crack profile can be losslessly compressed via RLE significantly, i.e., 8048 bytes to 654 bytes, and then can be completely recovered in the cloud server. This approach performs better than the previous research for image transmission through LoRa based on lossy compression [94,95], and can still satisfy the DT services, such as visualization and evaluation. However, it is worth noting that the LoRa data rate is still relatively low, even if the example image only requires three communication packets at SF7/125KHz, it still needs the airtime of 1.107 s and duty cycle of 73.8 s for such a transmission. In this case, other LPWAN technologies with higher bandwidth, such as NB-IoT, are more recommended.

This experiment demonstrates that the proposed framework with the

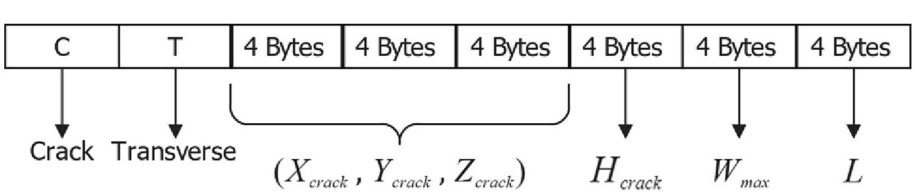

Fig. 19. Encoded defect information to synchronize for structural assessment.

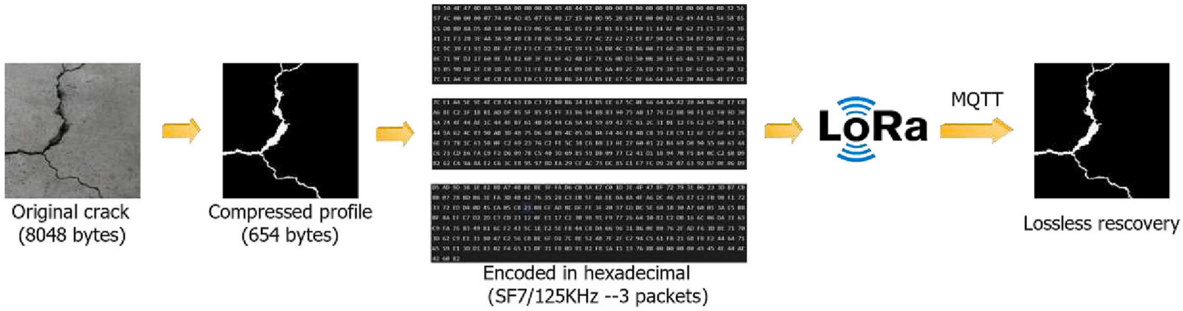

Fig. 20. Synchronization of crack profile through LoRa and RLE.

developed prototype can synchronize the drone-enabled inspection to the cloud server in near real-time for bridge DT services, such as structural assessment and visualization. Meanwhile, the feedback can be transmitted to local inspectors or agent-based drones through the on-board computer or the drone controller. The complete procedure can be shown in Fig. 21.

5.2. Vibration-based monitoring

Previous research [22] developed a cloud-based bridge DT to achieve real-time SHM based on vibration signals using a pre-trained surrogate model based on deep learning, but it relies on excellent communication (i.e., 5G) and its services will fail when cloud servers become unresponsive (lack of resilience). This experiment aims to achieve similar real-time SHM in the prototype and demonstrates the fault tolerance of the proposed framework for temporary loss of communication between edge and cloud.

The public dataset of acceleration signals from the VBM project of the KW51 bridge [74], is employed in the experiment, generated from 6 uniaxial accelerometers during train pass before and after bridge repair (i.e., the damaged and healthy condition respectively). Its sampling frequency is 825.8 Hz and the resolution is 24-bit. Such a huge amount of data is a challenge for synchronization, especially in a communication-constraint environment, i.e., LoRa here. Edge computing is taken on the Raspberry Pi 4 model B. Two different machine-learning approaches are developed for bridge risk identification. One is based on the support vector machine (SVM) with hand-crafted features, including signal features [96] or wavelet-packet energy (WPE) as Eq. 14, and the other is based on 1D-CNN with multi-channel input (shown as Fig. 22), which is similar to the DNN model used in previous bridge DT [22].

$$WPE_{level,i} = \sum |x(n)|^2 \tag{14}$$

Note: wavelet-packet energy (WPE) of discrete-time signal for each

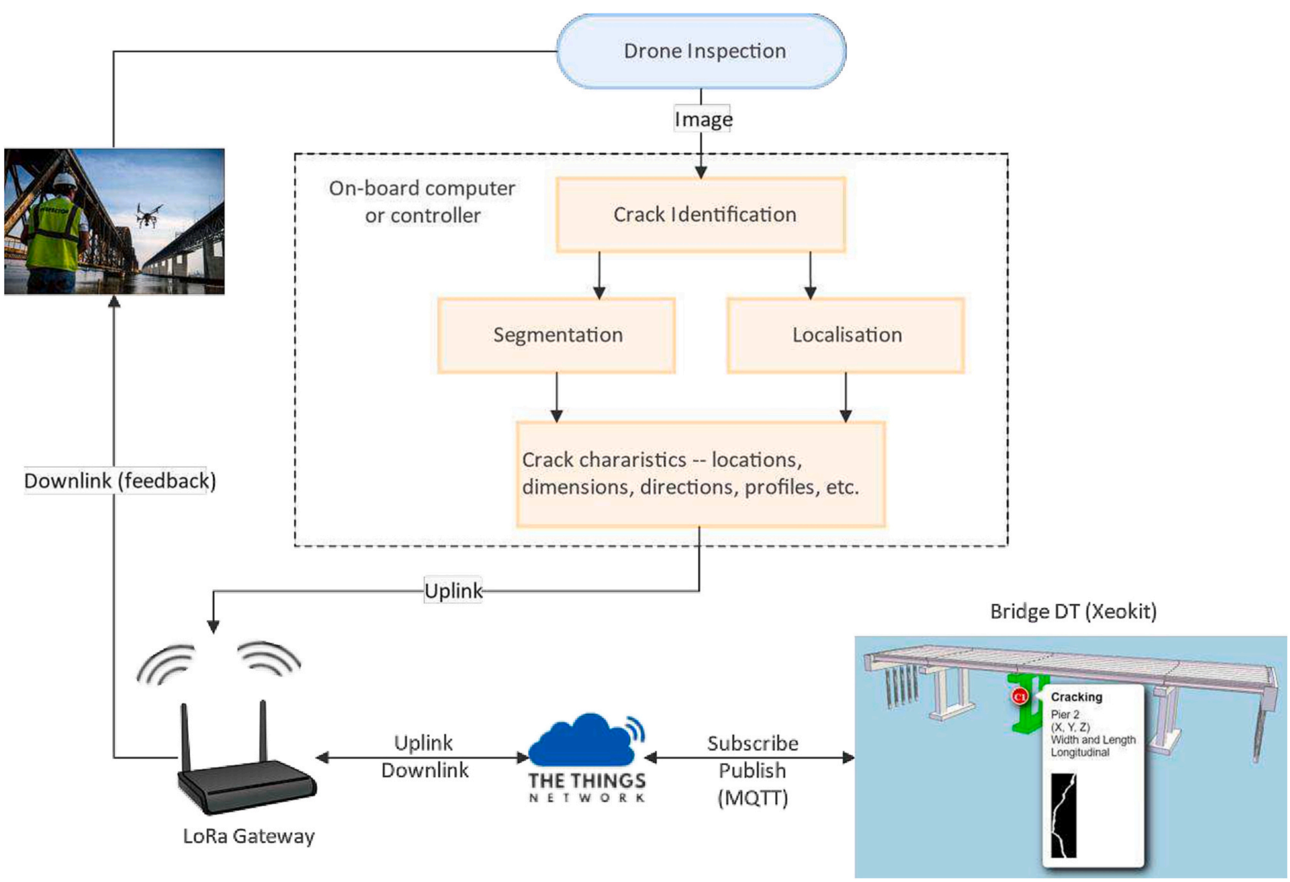

Fig. 21. Developed bridge DT prototype for drone-enabled bridge inspection.

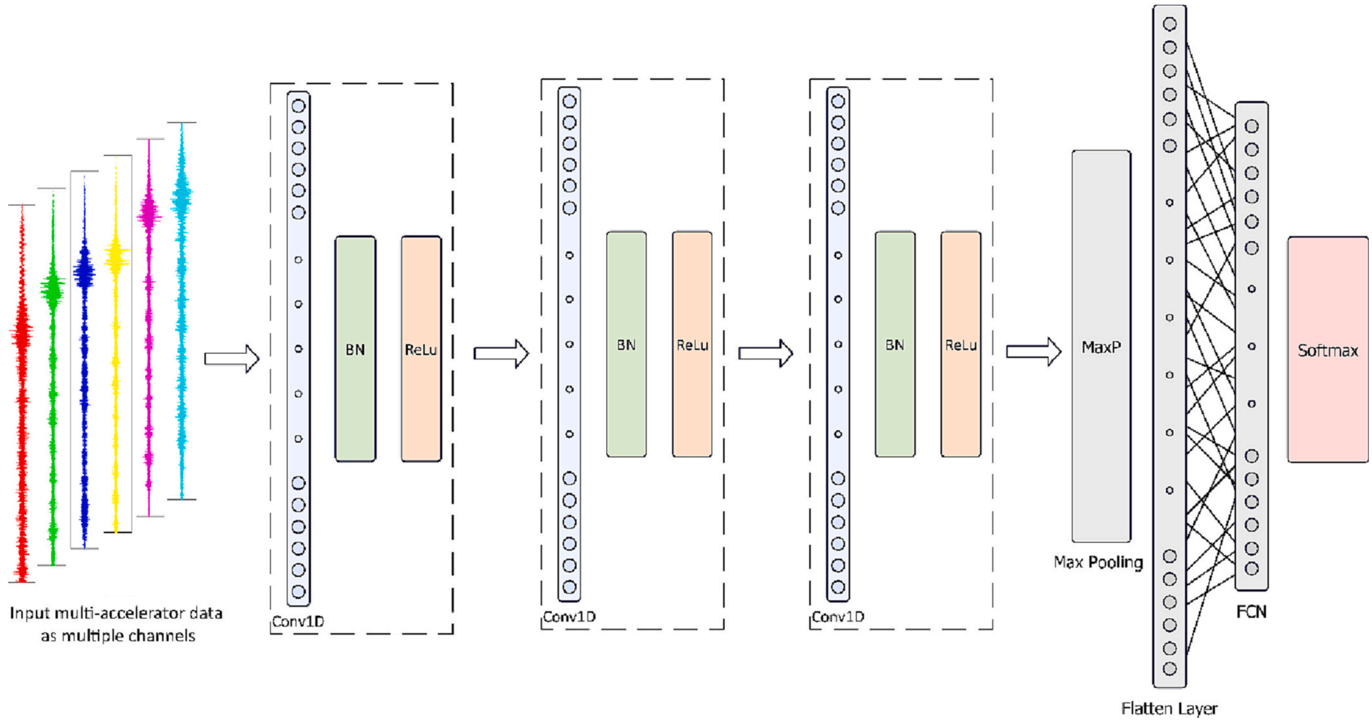

Fig. 22. 1D-CNN architecture with multi-channel input utilized for pattern recognition.

node (level = 3, i = 1–7).

The SVM model with the hand-crafted features is trained on a laptop using GridSearchCV to find the optimal parameter values. Its confusion matrices on the test set are shown in Fig. 23 and Table 2. The 1D-CNN model with raw data (50,176 × 6) is trained through TensorFlow on the Google Codelab, i.e., train-validate-test split – 60%:20%:20%; optimizer – SGD; learning rate – 0.001; batch size – 64. Then it is converted to a tinyML version. Its performance is shown in Table 2. Finally, all the models are deployed on the Raspberry Pi, which can be utilized in the embedded system installed on the physical bridge. Two LED lights (green and red) are connected to the GPIO pins of the Raspberry Pi, whereby the result can be displayed by switching either of them on. With normal communication, the edge can transmit the identified bridge pattern and the timestamp to the cloud using a predefined code system, and then receive cloud-based feedback. Otherwise, when there is a loss of communication between the edge and cloud, the local embedded system can still perform preliminary pattern recognition and trigger corresponding measures on the physical bridge, such as weight restriction, traffic diversion, or even closure. The edge-based inference of SVM with signal features or WPE takes the average time of 1.2104 and

Table 2
Model performance for pattern recognition.

Model	Input	Accuracy	Precision	Recall	F1 score
SVM	SF	0.85	0.83	0.88	0.86
SVM	WPE	0.96	0.98	0.95	0.96
1D-CNN	Acceleration	1	1	1	1

Note: SF – signal features; WPE – wavelet-packet energy (level-3); acceleration data – 50,176 × 6.

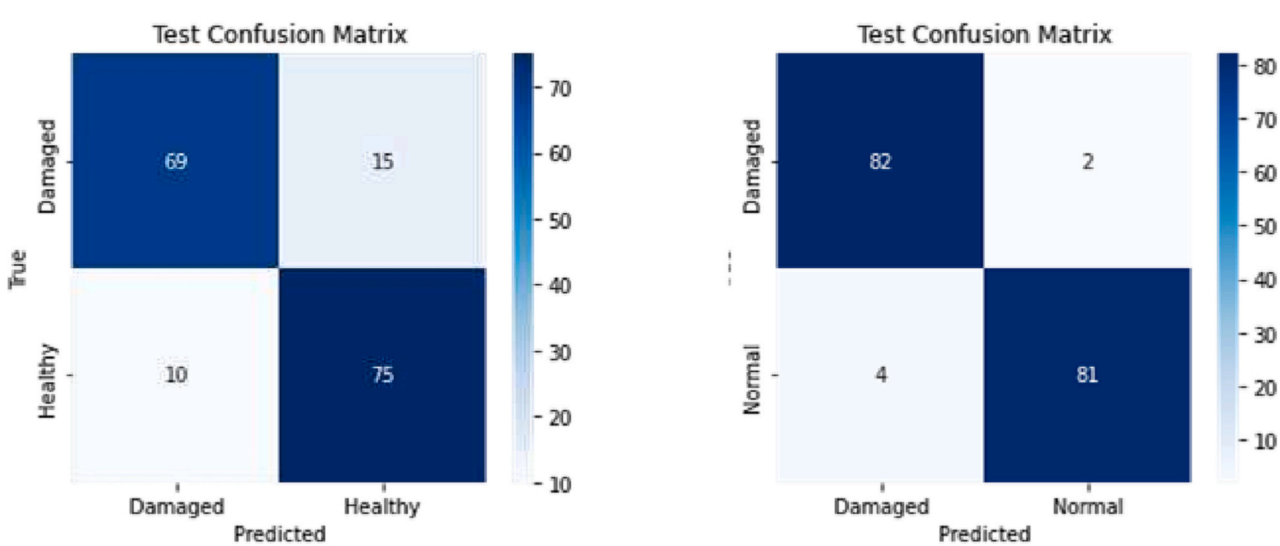

Fig. 23. Test confusion matrices for SVM with signal features and SVM with WPE.

1.5309 s respectively on the test set, while 1D-CNN with raw data takes 0.7825 s. Meanwhile, 1D-CNN has the best performance, shown in Table 2.

It is worth noting that the derived signal features and WPE can be either utilized at the edge or transmitted to the cloud, e.g., signal features require a total of 33 bytes with the calculated airtime of 2.302 s at SF12/125 kHz for LoRa, because their communication complexity has become much lower than the transmission of raw data in the previous DT. This experiment demonstrates that the proposed framework with the developed prototype can achieve similar performance to the previous research [22] for real-time SHM but does not rely on excellent communication anymore, and has better fault tolerance in the operation to guarantee the safety of the physical bridge. The complete procedure is shown in Fig. 24.

5.3. Dynamic evacuation

As an emergency response, dynamic evacuation with route planning is necessary when a disaster is happening or is predicted to happen on the site of physical bridges, such as flash floods or earthquakes. Route planning is a well-known problem, which can be solved with many approaches, such as Floyd [97] and Dijkstra [98] algorithms. However, under extreme weather conditions, the cloud server likely becomes unresponsive, e.g., the internet or gateways break down temporarily, thereby resulting in the failure of DT service for dynamic route planning in the cloud-based system. This is dangerous for public users traveling on the site of bridges. This experiment aims to demonstrate the resilience of the proposed framework to endure a temporary loss of communication at different levels.

An open-source emulator for the LoRa network [99] is employed for the experiment. Suppose there is an area with multiple bridges under the threat of flash flooding, as shown in Fig. 25. People need to evacuate from the left side (flooding area) to the right side (safe area) of the dashed line in the bridge network. The LoRa sensor nodes for water-level monitoring are activated by gateways and can also exchange messages with each other. The stars represent the gateways, while the squares stand for the bridges. The dashed line in communication topology is the LoRa connection, while the complete line and weight in the bridge network are the road and distance between bridges. As a prerequisite, the LoRa module of end-devices is designed to be at least Class B during an emergency, making them reachable at preconfigure times. Gateways have eight channels (sub-bands), allowing sufficient capability for uplink and downlink, thereby minimizing the duty-cycle influence, and can also transmit messages through LoRa between each other as well.

For simplicity, there are only two conditions for bridge serviceability in the simulation, i.e., Y - available, and N - closed. The route planning is only updated when a gateway or sensor node receives the message that a bridge becomes closed, and then the affected weights become infinitely great. The information is encoded as a message of characters indicating the bridge's location and condition. For example, "BN" means bridge B becomes unavailable. Because LoRa gateways are usually built on the Raspberry Pi, they have sufficient computing capability to find the new shortest path via the Floyd algorithm (considering all the nodes) with computational complexity is $O(n^3)$ and space complexity is $O(n)$, where n is the number of nodes. Similarly, the sensor nodes built on Arduinos can also perform computing to find the new shortest path from their own to the safe area via the Dijkstra algorithm with computational complexity $O((n+m)logn)$ and space complexity $n+m$, where n is the number of nodes and m is the number of edges in the graph. Therefore, if cloud servers become unresponsive temporarily, the dynamic route planning for evacuation can still work as the following procedures.

Procedures. Dynamic route planning for evacuation

Input: information of bridge locations and conditions "BN", "CN", ... "HN"
Output: shortest evacuation path for each node $path(n_i)$

while *update messages from unavailable bridges* **do**
  if *cloud servers become unresponsive* **then**
    transmit messages through LoRa between gateways;
    compute $p(n_i)$ on gateways via Floyd algorithm;
    if *gateways become unresponsive* **then**
      relay messages through LoRa via sensor nodes;
      compute $p(n_i)$ at each sensor node via Dijkstra algorithm;
return $path(n_i)$;

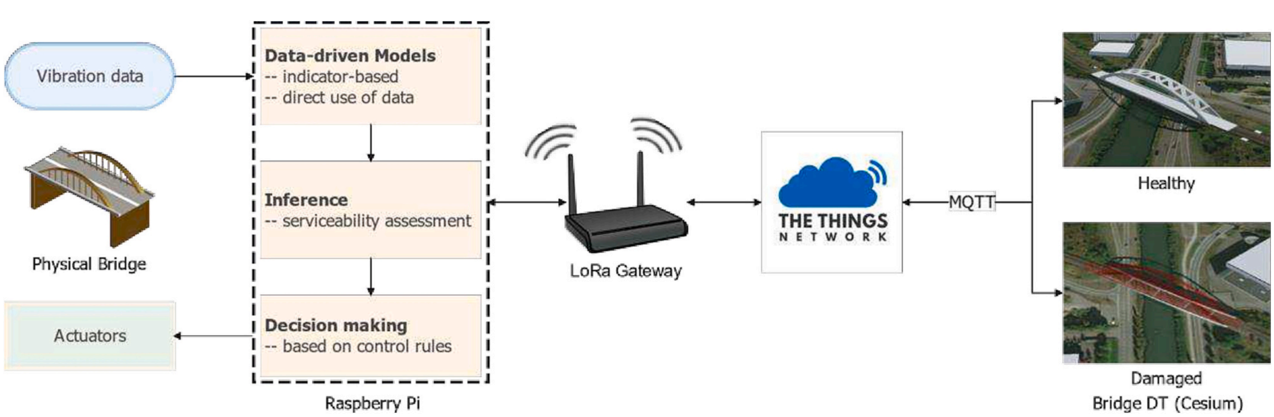

Fig. 24. Proposed framework and developed prototype for bridge VBM.

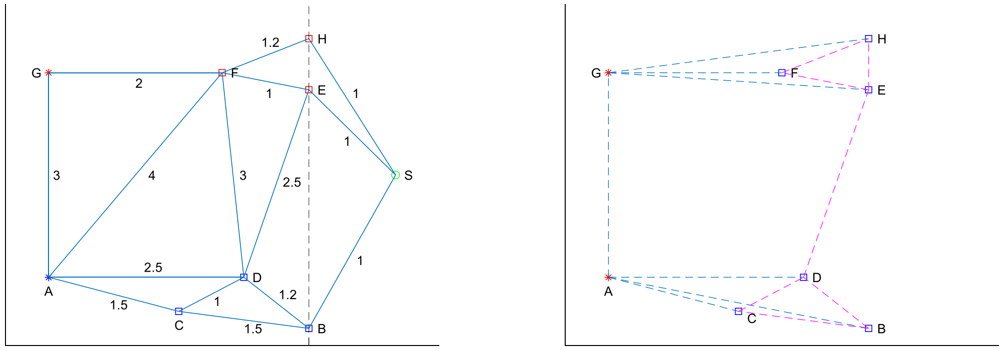

Fig. 25. (1) bridge network; (2) communication topology.

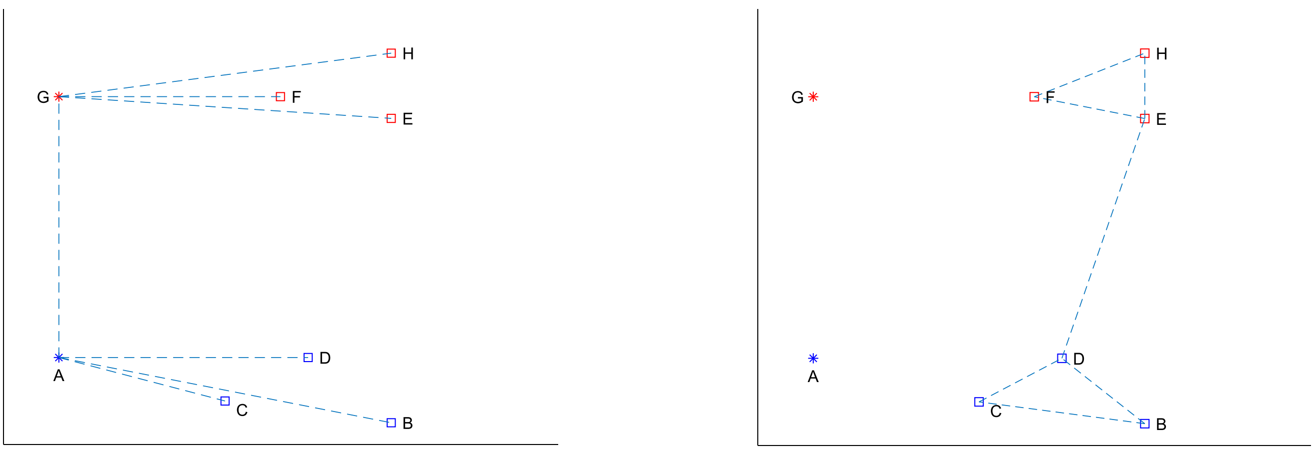

Fig. 26. (1) gateway-based route planning (2) sensor-node-based route planning.

Table 3
Simulation result for dynamic route planning.

Nodes	Initial Route and Distance	BN Route and Distance	Communication Time
A	A C B S and 4	A D E S and 6	2.8017 s / 3.4653 s
B	B S and 1	B D E S and 4.7	2.8017 s / 3.4653 s
C	C B S and 2.5	C D E S and 4.5	2.8017 s / 3.4653 s
D	D B S and 2.2	D E S and 3.5	2.8017 s / 3.4653 s
E	E S and 1	E S and 1	0/ 3.4653 s
F	F E S and 2	F E S and 2	0/ 3.4653 s
G	G F E S and 4	G F E S and 4	0/ 3.4653 s
H	H S and 1	H S and 1	0/ 3.4653 s

The simulation is initialized with all the bridges available. When bridge B becomes closed in the simulation, and cloud servers are out of the connection, the gateways can transmit the messages through LoRa and take the job of dynamic route planning. Moreover, when the gateways become unresponsive as well, the sensor nodes will relay the message through LoRa between each other and find the shortest evacuation route on their own, which becomes a decentralized mode. The difference between gateway-based and sensor-node-based route planning is shown in Fig. 26.

The results are shown in Table 3. Taking "BN" as an example, when cloud servers become unresponsive, it will result in new route planning at nodes A, B, C, and D based on the gateways. The downlink instruction message can be encoded as "BDES CDES DES" (i.e., 13 bytes), which will take up to 2.8017 s of airtime at the mode SF12/125 kHz for both uplink and downlink between nodes and gateways. Moreover, when gateways become unresponsive either, "BN" can be relayed through LoRa to all the nodes (i.e., $B \rightarrow C, D \rightarrow E \rightarrow F, H$). The communication time cost is up to 3.4653 s. This experiment demonstrates the excellent fault-tolerant capability of the proposed framework for DT services to endure a temporary loss of communication, especially under emergent situations. In practice, PE (i.e., multiple bridges) in the proposed framework becomes a resilient and self-adaptive subsystem under such conditions, of which the behavior can be predicted and simulated in the cloud, so PE and VE can be re-synchronized seamlessly when the communication recovers.

Table 4
Comparative analysis between the proposed framework and previous bridge DTs.

Features	Proposed Framework	cDTSHM [31]	Broo et al. [21]	Shim et at. [17]	Jeong et al. [100]
Level	Prototype	Prototype	Pilot project	Concept	Prototype
Data type	Heterogenous	time-series	time-series	Heterogenous	time-series
Data collection	Automatic	Semi-automatic	Automatic	Manual	Semi-automatic
Pre-processing	Edge	Fog layer	Cloud	Cloud	Cloud
Computing	Edge & Cloud	Cloud	Cloud	Local server	Cloud
Communication	LPWAN	5G	Ethernet and 4G	N/A	4G
Resilience	Yes	No	No	N/A	No
HMI	Web	Web	Web	Desktop	Web
Time delay	Near real-time	Near real-time	Near real-time	Periodic	Near real-time

5.4. Comparative analysis

A comparative analysis between the proposed framework and the existing bridge DTs is presented in Table 4. As can be seen, although most of the previous bridge DTs can achieve near real-time DT services based on cloud computing, they relied on excellent communication for data acquisition such as Ethernet, 5G, and 4G, and did not consider the system resilience. Compared with the previous works, the proposed AIoT-informed DT framework and the developed cross-platform prototype can handle massive heterogeneous data efficiently using AI-based edge computing and perform DT services in near real-time, even under the communication-constraint circumstance, i.e., LPWAN. Moreover, it has excellent fault tolerance, which can endure a temporary loss of communication rather than fail completely, and is scalable to support single or multiple bridges in a large area.

6. Conclusion and discussion

With the development of sensing and IoT technologies, massive heterogeneous data from regular inspection and real-time monitoring has become a huge challenge for bridge DT synchronization. However, when DT implementation concerns bridge locations, it may have many issues, such as restricted communication. Meanwhile, most existing bridge DTs are cloud-based and rely on excellent communication without consideration of system resilience to endure a temporary loss of communication. To solve these issues, this work proposed an AIoT-informed DT communication framework to support bridge O&M in a communication-constraint environment with high efficiency, low latency, and excellent fault tolerance.

Firstly, the research indicates that the time delay of DT services consists of computation and communication time costs, which depend on computational and communication complexity respectively, and reveals the distinct impact of their sequence on time consumption for DT services, i.e., edge computing can help to reduce time delay significantly when communication time is dominant in the process. Information hierarchy (i.e., DIKW) is leveraged to indicate how to reduce communication complexity using AI-based edge computing theoretically. Moreover, two-way communication between edge and cloud is recommended to satisfy the restricted communication with minimal complexity for big-data analysis, which involves different data sources owned by edge and cloud, thereby decreasing the time delay. AI-based edge computing can enable the system with resilience to endure a temporary loss of communication, such as preliminary analysis and decision-making, which is especially beneficial to the safety of physical bridges and public users when a disaster is happening or is predicted to happen. Furthermore, a hierarchical communication architecture with excellent fault tolerance can be designed based on LPWAN and the mesh network for different-level tasks.

Then a bridge DT system is idealized mathematically, including state-space representation with time delay and inequalities for hardware processing capability. Meanwhile, the data flow for DT services and the resilience of the proposed framework are demonstrated based on Petri-net modeling with token and conditional probability. Furthermore, the framework is developed to the level of a prototype with cross-platform integration for bridge O&M, including AI-based edge computing, LPWAN communication, cloud servers, MQTT protocols, and a web-based platform with both GIS and BIM.

Finally, the proposed framework and prototype are validated with different cases for bridge O&M, including drone-enabled inspection, VBM, and dynamic evacuation. To enhance the efficiency of AI-based inference at the edge, the deep-learning model is trained on Google CodeLab and then converted to a tinyML version for deployment in the cases. The results demonstrate that 1) the proposed framework can achieve DT synchronization during drone inspection in near real-time under communication-constraint circumstances such as LPWAN; 2) the prototype can achieve similar performance to the previous cloud-based DT [22] in near real-time for vibration-based SHM without relying on excellent communication and has extra resilience; 3) the framework can achieve excellent fault tolerance for DT services through the hierarchical communication architecture to endure a temporary loss of communication at different levels for single or multiple bridges in a large area. These benefits can contribute directly to the efficiency and safety of bridge O&M through DT.

In the next step, the proposed DT framework and prototype will be implemented on a real bridge in the UK for practical application. This framework is also promising for federated learning to protect privacy because different stakeholders prefer to preserve their AI models, which are derived from specific domain knowledge and experience, rather than share them on the cloud. Although this framework has many benefits, which can contribute directly to the efficiency and safety of bridge O&M through DT, it still has some limitations. For example, LPWAN can reduce the power consumption of communication significantly, but AI-based edge computing raises a high requirement for power supply according to the tasks and algorithms, which can be an issue under resource-constraint circumstances. Hence, edge-based AI can only perform preliminary analysis and decision-making currently. Therefore, the sustainable power supply for edge devices and the trade-off between edge and cloud in data storage and computation have great research significance in the future.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Cardiff University – China Scholarship Council (CSC) joint program.

References

[1] Structurally Deficient Bridges | Bridge Infrastructure | ASCE's, Infrastructure Report Card, 2021. https://infrastructurereportcard.org/cat-item/bridges-infrastructure/. accessed Feb. 04, 2023).

[2] Number of substandard road bridges on the rise again. https://www.racfoundation.org/media-centre/number-of-substandard-road-bridges-on-the-rise-again, 2023 accessed Feb. 04, 2023.

[3] V. Barrile, G. Candela, A. Fotia, E. Bernardo, UAV Survey of Bridges and Viaduct: Workflow and Application, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11622, LNCS, 2019, pp. 269–284, https://doi.org/10.1007/978-3-030-24305-0_21.

[4] S.T. Nguyen, H.M. La, A climbing robot for steel bridge inspection, J. Intell. Robotic Syst.: Theory and Appl. 102 (4) (2021) 1–21, https://doi.org/10.1007/s10846-020-01266-1.

[5] H. Alexakis, A. Franza, S. Acikgoz, M.J. Dejong, Monitoring bridge degradation using dynamic strain, acoustic emission and environmental data, in: International Conference on Smart Infrastructure and Construction 2019, ICSIC 2019: Driving Data-Informed Decision-Making 2019, 2019, pp. 523–532, https://doi.org/10.1680/icsic.64669.523.

[6] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, S. Nagarajaiah, Review of bridge structural health monitoring aided by big data and artificial intelligence: from condition assessment to damage detection, J. Struct. Eng. 146 (5) (2020), https://doi.org/10.1061/(asce)st.1943-541x.0002535.

[7] R. Li, T. Mo, J. Yang, S. Jiang, T. Li, Y. Liu, Ontologies-based domain knowledge modeling and heterogeneous sensor data integration for bridge health monitoring systems, IEEE Trans. Industr. Inform. 17 (1) (2021) 321–332, https://doi.org/10.1109/TII.2020.2967561.

[8] Z. Allah Bukhsh, I. Stipanovic, A. Saeed, A.G. Doree, Maintenance intervention predictions using entity-embedding neural networks, Autom. Constr. 116 (March) (2020), 103202, https://doi.org/10.1016/j.autcon.2020.103202.

[9] D. Gürdür, M. Bravo-haro, J. Schooling, Automation in construction Design and implementation of a smart infrastructure digital twin, Autom. Constr. 136 (February) (2022), 104171, https://doi.org/10.1016/j.autcon.2022.104171.

[10] L. Design, et al., A scalable cloud-based cyberinfrastructure platform for bridge monitoring, Struct. Infrastruct. Eng. 15 (1) (2019) 82–102, https://doi.org/10.1080/15732479.2018.1500617.

[11] C. Shim, N. Dang, S. Lon, C. Jeon, Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model, Struct. Infrastruct. Eng. 15 (10) (2019) 1319–1332, https://doi.org/10.1080/15732479.2019.1620789.

[12] M. Li, X. Feng, Y. Han, Brillouin fiber optic sensors and mobile augmented reality-based digital twins for quantitative safety assessment of underground pipelines, Autom. Constr. 144 (March) (2022), 104617, https://doi.org/10.1016/j.autcon.2022.104617.

[13] M. Li, X. Feng, Multisensor data fusion-based structural health monitoring for buried metallic pipelines under complicated stress states, J. Civ. Struct. Heal. Monit. 12 (6) (2022) 1509–1521, https://doi.org/10.1007/s13349-022-00609-w.

[14] C. Ye, et al., A digital twin of bridges for structural health monitoring, in: Structural Health Monitoring 2019: Enabling Intelligent Life-Cycle Health Management for Industry Internet of Things (IIOT) - Proceedings of the 12th International Workshop on Structural Health Monitoring 1, 2019, pp. 1619–1626, https://doi.org/10.12783/shm2019/32287, no. February 2020.

[15] N.S. Dang, C.S. Shim, Bridge assessment for PSC girder bridge using digital twins model, Lecture Notes Civil Eng. 54 (January) (2020) 1241–1246, https://doi.org/10.1007/978-981-15-0802-8_199.

[16] J.S. Kang, K. Chung, E.J. Hong, Multimedia knowledge-based bridge health monitoring using digital twin, Multimed. Tools Appl. 80 (26–27) (2021) 34609–34624, https://doi.org/10.1007/s11042-021-10649-x.

[17] C.S. Shim, H.R. Kang, N.S. Dang, Digital twin models for maintenance of cable-supported bridges, in: International Conference on Smart Infrastructure and Construction 2019, ICSIC 2019: Driving Data-Informed Decision-Making, no. January, 2019, pp. 737–742, https://doi.org/10.1680/icsic.64669.737.

[18] J. Yang, F. Xiang, R. Li, L. Zhang, X. Yang, S. Jiang, X. Liu, Intelligent bridge management via big data knowledge engineering, Autom. Constr. 135 (2022), 104118, https://doi.org/10.1016/j.autcon.2021.104118.

[19] K. Liu, N. El-Gohary, Fusing data extracted from bridge inspection reports for enhanced data-driven bridge deterioration prediction: a hybrid data fusion method, J. Comput. Civ. Eng. 34 (6) (2020) 1–14, https://doi.org/10.1061/(asce)cp.1943-5487.0000921.

[20] D. Dan, Y. Ying, L. Ge, Digital twin system of bridges group based on machine vision fusion monitoring of bridge traffic load, IEEE Trans. Intell. Transp. Syst. 23 (11) (2021) 22190–22205, https://doi.org/10.1109/TITS.2021.3130025.

[21] D. Gürdür Broo, M. Bravo-Haro, J. Schooling, Design and implementation of a smart infrastructure digital twin, Autom. Constr. 136 (Apr. 2022), https://doi.org/10.1016/j.autcon.2022.104171.

[22] H. Dang, M. Tatipamula, H.X. Nguyen, Cloud-based digital twinning for structural health monitoring using deep learning, IEEE Trans. Ind. Inform. 18 (6) (2022) 3820–3830, https://doi.org/10.1109/TII.2021.3115119.

[23] N.S. Dang, H. Kang, S. Lon, C.S. Shim, 3D digital twin models for bridge maintenance, in: Proceedings of 10th International Conference on Short and Medium Span Bridges, no. 73, 2018, pp. 1–9 [Online]. Available: https://www.researchgate.net/publication/331314334_3D_DIGITAL_TWIN_MODELS_FOR_BRIDGE_MAINTENANCE.

[24] K. Maes, G. Lombaert, Monitoring railway bridge KW51 before, during, and after retrofitting, J. Bridg. Eng. 26 (3) (2021) 04721001, https://doi.org/10.1061/(asce)be.1943-5592.0001668.

[25] S.O. Sajedi, X. Liang, Vibration-based semantic damage segmentation for large-scale structural health monitoring, Comput.-Aided Civil Infrast. Eng. 35 (6) (2020) 579–596, https://doi.org/10.1111/mice.12523.

[26] C.Z. Dong, O. Celik, F.N. Catbas, E.J. O'Brien, S. Taylor, Structural displacement monitoring using deep learning-based full field optical flow methods, Struct. Infrastruct. Eng. 16 (1) (2020) 51–71, https://doi.org/10.1080/15732479.2019.1650078.

[27] X. Meng, D.T. Nguyen, Y. Xie, J.S. Owen, P. Psimoulis, S. Ince, P. Bhatia, Design and implementation of a new system for large bridge monitoring—geoshm, Sensors (Switzerland) 18 (3) (2018), https://doi.org/10.3390/s18030775.

[28] G. Ren, R. Ding, H. Li, Building an ontological knowledgebase for bridge maintenance, Adv. Eng. Softw. 130 (November 2018) (2019) 24–40, https://doi.org/10.1016/j.advengsoft.2019.02.001.

[29] J. Yang, F. Xiang, R. Li, L. Zhang, X. Yang, S. Jiang, X. Liu, Intelligent bridge management via big data knowledge engineering, Autom. Constr. 135 (November 2021) (2022) 104118, https://doi.org/10.1016/j.autcon.2021.104118.

[30] R. Li, T. Mo, J. Yang, S. Jiang, T. Li, Y. Liu, Ontologies-based domain knowledge modeling and heterogeneous sensor data integration for bridge health monitoring systems, IEEE Trans. Ind. Inform. 17 (1) (2021) 321–332, https://doi.org/10.1109/TII.2020.2967561.

[31] H. Dang, M. Tatipamula, H.X. Nguyen, Cloud-based digital twinning for structural health monitoring using deep learning, IEEE Trans. Ind. Inform. 18 (6) (Jun. 2022) 3820–3830, https://doi.org/10.1109/TII.2021.3115119.

[32] Hamburg Port Authority – smartBRIDGE Hamburg – World Port Sustainability Program. https://sustainableworldports.org/project/hamburg-port-authority-smartbridge/, 2023 accessed Feb. 21, 2023.

[33] P. Qian, D. Zhang, X. Tian, Y. Si, L. Li, A novel wind turbine condition monitoring method based on cloud computing, Renew. Energy 135 (2019) 390–398, https://doi.org/10.1016/j.renene.2018.12.045.

[34] H. Tran-Ngoc, S. Khatir, G. De Roeck, T. Bui-Tien, L. Nguyen-Ngoc, M. Abdel Wahab, Model updating for Nam O bridge using particle swarm optimization algorithm and genetic algorithm, Sensors 18 (2018) 12, https://doi.org/10.3390/s18124131.

[35] H. Dang, M. Tatipamula, H.X. Nguyen, Cloud-based digital twinning for structural health monitoring using deep learning, IEEE Trans. Ind. Inform. 18 (6) (2022) 3820–3830, https://doi.org/10.1109/TII.2021.3115119.

[36] J. Morris, D. Kroening, P. Koopman, Fault tolerance tradeoffs in moving from decentralized to centralized embedded systems, Int. Conf. Dependable Syst. Net. (2004) 377–386, https://doi.org/10.1109/DSN.2004.1311907.

[37] A. Helmrich, S. Markolf, R. Li, T. Carvalhaes, Y. Kim, E. Bondank, M. Chester, Complexity centralization and decentralization for resilient infrastructure and complexity, Environ. Res.: Infrastruct. Sustainabil. 1 (2021) 6 [Online]. Available: https://iopscience.iop.org/article/10.1088/2634-4505/ac0a4f/meta.

[38] C. Ye, S.C. Kuok, L.J. Butler, C.R. Middleton, Implementing bridge model updating for operation and maintenance purposes: examination based on UK practitioners' views, Struct. Infrastruct. Eng. 0 (0) (2021) 1–20, https://doi.org/10.1080/15732479.2021.1914115.

[39] P. Cawley, Structural health monitoring: closing the gap between research and industrial deployment, Struct. Health Monit. 17 (5) (Sep. 2018) 1225–1244, https://doi.org/10.1177/1475921717750047.

[40] A. Nair, C.S. Cai, Acoustic emission monitoring of bridges: review and case studies, Eng. Struct. 32 (6) (2010) 1704–1714, https://doi.org/10.1016/j.engstruct.2010.02.020.

[41] N. Yousefpour, S. Downie, S. Walker, N. Perkins, H. Dikanski, Machine learning solutions for bridge scour forecast based on monitoring data, Transport. Res. Record: J. Transport. Res. Board 2675 (10) (2021) 745–763, https://doi.org/10.1177/03611981211012693.

[42] S. Lee, N. Kalos, D.H. Shin, Non-destructive testing methods in the U.S. for bridge inspection and maintenance, KSCE J. Civ. Eng. 18 (5) (2014) 1322–1331, https://doi.org/10.1007/s12205-014-0633-9.

[43] T. Omar, M.L. Nehdi, Thermal detection of subsurface delaminations in reinforced concrete bridge decks using unmanned aerial vehicle, American Concrete Institute, ACI Special Publication 2017-March (SP 331) (2017) 1–14, https://doi.org/10.14359/51715590.

[44] C.M. Yeum, S.J. Dyke, Vision-based automated crack detection for bridge inspection, Comput.-Aided Civil Infrast. Eng. 30 (10) (2015) 759–770, https://doi.org/10.1111/mice.12141.

[45] C. Ye, S. Acikgoz, S. Pendrigh, E. Riley, M.J. DeJong, Mapping deformations and inferring movements of masonry arch bridges using point cloud data, Eng. Struct. 173 (July) (2018) 530–545, https://doi.org/10.1016/j.engstruct.2018.06.094.

[46] V. Barrile, G. Candela, A. Fotia, E. Bernardo, UAV Survey of Bridges and Viaduct: Workflow and Application, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11622, LNCS, 2019, pp. 269–284, https://doi.org/10.1007/978-3-030-24305-0_21.

[47] B. Sutter, et al., A semi-autonomous mobile robot for bridge inspection, Autom. Constr. 91 (May 2017) (2018) 111–119, https://doi.org/10.1016/j.autcon.2018.02.013.

[48] Y. Gao, H. Li, G. Xiong, An Efficient and Resilient Digital-twin Communication Framework for Smart Bridge Structural Survey and Maintenance, in: Proceedings of 29th International Workshop on Intelligent Computing in Engineering (EG-ICE), Aarhus, Denmark, 2022, pp. 165–175, https://doi.org/10.7146/aul.455.c207.

[49] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, A. Taroni, Wired and wireless sensor networks for industrial applications, Microelectron. J. 40 (9) (2009) 1322–1336, https://doi.org/10.1016/j.mejo.2008.08.012.

[50] L. Alonso, J. Barbarán, J. Chen, M. Díaz, L. Llopis, B. Rubio, Middleware and communication technologies for structural health monitoring of critical infrastructures: A survey, Comput. Standards Interf. 56 (March 2017) (2018) 83–100, https://doi.org/10.1016/j.csi.2017.09.007.

[51] B. Foubert, N. Mitton, Long-range wireless radio technologies: a survey, Future Internet 12 (1) (2020), https://doi.org/10.3390/fi12010013.

[52] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, A comparative study of LPWAN technologies for large-scale IoT deployment, ICT Express 5 (1) (2019) 1–7, https://doi.org/10.1016/j.icte.2017.12.005.

[53] S.M. Oh, Y.J. Kim, C.S. Park, I.H. Kim, Process-driven BIM-based optimal design using integration of EnergyPlus, genetic algorithm, and Pareto optimality, in: Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, Australia, 2011, pp. 894–901 [Online]. Available: http://www.ibpsa.org/proceedings/BS2011/P_1354.pdf.

[54] K. Afsari, L. Florez, E. Maneke, M. Afkhamiaghda, An experimental investigation of the integration of smart building components with building information model (BIM), in: Proceedings of the 36th International Symposium on Automation and Robotics in Construction, ISARC 2019, no. Isarc, 2019, pp. 578–585, https://doi.org/10.22260/isarc2019/0077.

[55] J.M.D. Delgado, L.J. Butler, N. Gibbons, I. Brilakis, M.Z.E.B. Elshafie, C. Middleton, Management of structural monitoring data of bridges using BIM, Proceed. Instit. Civil Eng.: Bridge Eng. 170 (3) (2017) 204–218, https://doi.org/10.1680/jbren.16.00013.

[56] Y. Xu, Y. Turkan, BrIM and UAS for bridge inspections and management, Eng. Constr. Archit. Manag. 27 (3) (2020) 785–807, https://doi.org/10.1108/ECAM-12-2018-0556.

[57] P. Singh, A. Sadhu, System identification-enhanced visualization tool for infrastructure monitoring and maintenance, Front. Built. Environ. 6 (May) (2020), https://doi.org/10.3389/fbuil.2020.00076.

[58] C. Boddupalli, A. Sadhu, E. Rezazadeh Azar, S. Pattyson, Improved visualization of infrastructure monitoring data using building information modeling, Struct. Infrastruct. Eng. 15 (9) (2019) 1247–1263, https://doi.org/10.1080/15732479.2019.1602150.

[59] J.M. Davila Delgado, L.J. Butler, I. Brilakis, M.Z.E.B. Elshafie, C.R. Middleton, Structural performance monitoring using a dynamic data-driven BIM environment, J. Comput. Civ. Eng. 32 (3) (2018) 04018009, https://doi.org/10.1061/(asce)cp.1943-5487.0000749.

[60] F. Diara, F. Rinaudo, Ark-bim: open-source cloud-based hbim platform for archaeology, Appl. Sci. (Switzerland) 11 (18) (Sep. 2021), https://doi.org/10.3390/app11188770.

[61] L. Chamari, E. Petrova, P. Pauwels, A web-based approach to BMS, BIM and IoT integration: a case study, in: CLIMA 2022 The 14th REHVA HVAC World Congress, May 2022, https://doi.org/10.34641/clima.2022.228.

[62] Y. Zhang, J. Beetz, Building-CPS: Cyber-Physical System for Building Environment Monitoring, in: Proceedings of the Conference CIB W78, 2021, pp. 11–15 [Online]. Available: https://www.researchgate.net/publication/355456670.

[63] Y.M. Hsieh, Y.S. Lu, Visualization system for field monitoring data and its effectiveness, Autom. Constr. 26 (2012) 54–68, https://doi.org/10.1016/j.autcon.2012.03.004.

[64] A. Khudhair, H. Li, T. Bower, G. Ren, A theoretical holistic decision-making framework supporting collaborative design based on common data analysis (CDA) method, J. Build. Eng. 46 (November) (2022), 103686, https://doi.org/10.1016/j.jobe.2021.103686.

[65] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, S. Nagarajaiah, Review of bridge structural health monitoring aided by big data and artificial intelligence: from condition assessment to damage detection, J. Struct. Eng. 146 (5) (May 2020) 04020073, https://doi.org/10.1061/(asce)st.1943-541x.0002535.

[66] C.H.J. Duy Cuong Nguyen, Chang Su Shim, The Quan Nguyen, Ruoyu Jin, 'Developing a mixed-reality based application for bridge inspection and maintenance, in: Enabling The Development And Implementation of Digital Twins - Proceedings of the 20th International Conference on Construction Applications of Virtual Reality - (30th Sep - 2nd Oct 2020), Teeside University, 2020, pp. 31–43 [Online]. Available, https://openresearch.lsbu.ac.uk/item/8qyzx.

[67] M. Omer, L. Margetts, M. Hadi Mosleh, S. Hewitt, M. Parwaiz, Use of gaming technology to bring bridge inspection to the office, Struct. Infrastruct. Eng. 15 (10) (2019) 1292–1307, https://doi.org/10.1080/15732479.2019.1615962.

[68] Y. Li, M.M. Karim, R. Qin, A virtual-reality-based training and assessment system for bridge inspectors with an assistant drone, IEEE Trans. Hum. Mach. Syst. (2022) 1–11, https://doi.org/10.1109/thms.2022.3155373.

[69] S.N. Sakib, T. Ane, N. Matin, M.S. Kaiser, An intelligent flood monitoring system for Bangladesh using wireless sensor network, in: 2016 5th international conference on informatics, Electronics and Vision 2016, ICIEV, 2016, pp. 979–984, https://doi.org/10.1109/ICIEV.2016.7760145.

[70] E. Leon, C. Alberoni, M. Wister, J. Hernández-Nolasco, Flood Early Warning System by Twitter Using LoRa, 2018, p. 1213, https://doi.org/10.3390/proceedings2191213.

[71] I. Srikanth, M. Arockiasamy, Deterioration models for prediction of remaining useful life of timber and concrete bridges: a review, J. Traffic and Transport. Eng. (English Edition) 7 (2) (2020) 152–173, https://doi.org/10.1016/j.jtte.2019.09.005.

[72] Y. Goi, C.W. Kim, Damage detection of a truss bridge utilizing a damage indicator from multivariate autoregressive model, J. Civ. Struct. Heal. Monit. 7 (2) (2017) 153–162, https://doi.org/10.1007/s13349-017-0222-y.

[73] X.X. Cheng, J.H. Fan, Z.H. Xiao, Finite element model updating for the Tsing ma bridge tower based on surrogate models, J. Low Frequency Noise Vibration and Active Cont. 41 (2) (2022) 500–518, https://doi.org/10.1177/14613484211058999.

[74] S.O. Sajedi, X. Liang, Vibration-based semantic damage segmentation for large-scale structural health monitoring, Comput.-Aided Civil Infrast. Eng. 35 (6) (2020) 579–596, https://doi.org/10.1111/mice.12523.

[75] R.K. Verma, K.K. Pattanaik, P.B.R. Dissanayake, A.J. Dammika, H.A.D. Buddika, M.R. Kaloop, Damage detection in bridge structures: An edge computing approach, arXiv preprint (2023), https://doi.org/10.48550/arXiv.2008.06724.

[76] A. Anaissi, B. Suleiman, W. Alyassine, Personalised federated learning framework for damage detection in structural health monitoring, J. Civ. Struct. Heal. Monit. (2022), https://doi.org/10.1007/s13349-022-00615-y.

[77] Artificial Intelligence of Things - Wikipedia. https://en.wikipedia.org/wiki/Artificial_intelligence_of_things, 2023 accessed Jan. 28, 2023.

[78] D. Harris-birtill, R. Harris-birtill, Understanding computation time: a critical discussion of time as a computational performance metric, Time in Variance 2 (2021) 220–248, https://doi.org/10.1163/9789004470170_014.

[79] W. Dean, Computational Complexity Theory [Online]. Available: https://plato.stanford.edu/entries/computational-complexity/, 2015.

[80] S. Baruah, A. Burns, Sustainable scheduling analysis, Proceed. - Real-Time Syst. Symp. (2006) 159–168, https://doi.org/10.1109/RTSS.2006.47.

[81] S. Baškarada, A. Koronios, Data, information, knowledge, wisdom (DIKW): a semiotic theoretical and empirical exploration of the hierarchy and its quality dimension, Australas. J. Inf. Syst. 18 (1) (2013) 5–24, https://doi.org/10.3127/ajis.v18i1.748.

[82] E. Bianchi, A.L. Abbott, P. Tokekar, M. Hebdon, COCO-bridge: structural detail data set for bridge inspections, J. Comput. Civ. Eng. 35 (3) (2021), https://doi.org/10.1061/(asce)cp.1943-5487.0000949.

[83] A.A. Razborov, Communication complexity, An Invit. Math. (2011) 97–117, https://doi.org/10.1007/978-3-642-19533-4_8.

[84] Fault tolerance - Wikipedia. https://en.wikipedia.org/wiki/Fault_tolerance, 2023 accessed Aug. 27, 2022.

[85] M. Chiachío, M. Megía, J. Chiachío, J. Fernandez, M.L. Jalón, Structural digital twin framework: formulation and technology integration, Autom. Constr. 140 (May) (2022), 104333, https://doi.org/10.1016/j.autcon.2022.104333.

[86] T. Murata, Petri nets: properties, analysis and applications, Proc. IEEE 77 (4) (1989) 541–580, https://doi.org/10.1109/5.24143.

[87] J.R. Silva, P.M.G. Foyo, Timed petri nets, Comput. Net. ISDN Syst. 10 (5) (1985) 312–313, https://doi.org/10.1016/0169-7552(85)90073-X.

[88] W. Li, et al., On enabling sustainable edge computing with renewable energy resources, IEEE Commun. Mag. 56 (5) (2018) 94–101, https://doi.org/10.1109/MCOM.2018.1700888.

[89] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, Overview of cellular LPWAN technologies for IoT deployment: Sigfox, LoRaWAN, and NB-IoT, in: 2018 IEEE international conference on pervasive computing and communications workshops, PerCom Workshops 2018, 2018, pp. 197–202, https://doi.org/10.1109/PERCOMW.2018.8480255.

[90] V.K. Sarker, J.P. Queralta, T.N. Gia, H. Tenhunen, T. Westerlund, A survey on LoRa for IoT: Integrating edge computing, in: 2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019, no. June, 2019, pp. 295–300, https://doi.org/10.1109/FMEC.2019.8795313.

[91] C.S. Shim, N.S. Dang, S. Lon, C.H. Jeon, Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model, Struct. Infrastruct. Eng. 15 (10) (Oct. 2019) 1319–1332, https://doi.org/10.1080/15732479.2019.1620789.

[92] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, X. Chen, Automatic bridge crack detection using a convolutional neural network, Appl. Sci. (Switzerland) 9 (14) (2019), https://doi.org/10.3390/app9142867.

[93] D. Tan, Assessment of existing Bridge's beam bending stiffness using crack characteristics, Engineering 12 (02) (2020) 82–89, https://doi.org/10.4236/eng.2020.122008.

[94] T. Chen, D. Eager, D. Makaroff, Efficient Image Transmission Using LoRa Technology In Agricultural Monitoring IoT Systems, in: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 2019, pp. 937–944, https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00166.

[95] C. Pham, Robust CSMA for long-range LoRa transmissions with image sensing devices, IFIP Wireless Days 2018-April (2018) 116–122, https://doi.org/10.1109/WD.2018.8361706.

[96] Signal Features - MATLAB & Simulink - MathWorks United Kingdom. https://uk.mathworks.com/help/predmaint/ug/signal-features.html, 2023 accessed Feb. 01, 2023.

[97] J. Wang, Y. Sun, Z. Liu, P. Yang, T. Lin, Route planning based on floyd algorithm for intelligence transportation system, in: IEEE ICIT 2007–2007 IEEE International Conference on Integration Technology, 2007, pp. 544–546, https://doi.org/10.1109/ICITECHNOLOGY.2007.4290376.

[98] D.K. Fan, P. Shi, Improvement of Dijkstraa's algorithm and its application in route planning, in: Proceedings - 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, vol. 4, no. Fskd, 2010, pp. 1901–1904, https://doi.org/10.1109/FSKD.2010.5569452.

[99] B. al Homssi, K. Dakic, S. Maselli, H. Wolf, S. Kandeepan, A. Al-Hourani, IoT network Design using open-source LoRa coverage emulator, IEEE Access 9 (2021) 53636–53646, https://doi.org/10.1109/ACCESS.2021.3070976.

[100] S. Jeong, R. Hou, J.P. Lynch, H. Sohn, K.H. Law, A scalable cloud-based cyberinfrastructure platform for bridge monitoring, Struct. Infrastruct. Eng. 15 (1) (Jan. 2019) 82–102, https://doi.org/10.1080/15732479.2018.1500617.