Heliyon 8 (2022) e09634

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Smart deployment of IoT-TelosB service care StreamRobot using software-defined reliability optimisation design

Kennedy Chinedu Okafor [a,b,*], Omowunmi Mary Longe [b]

[a] Mechatronics Engineering, Federal University of Technology-Owerri, Nigeria
[b] Electrical and Electronic Engineering Science, University of Johannesburg, South Africa

ARTICLE INFO

Keywords:
Cloud networks
Internet of things
StreamRobots
Neural network
Fog analytics
Software-defined network

ABSTRACT

Intelligent service care robots have increasingly been developed in mission-critical sectors such as healthcare systems, transportation, manufacturing, and environmental applications. The major drawbacks include the open-source Internet of Things (IoT) platform vulnerabilities, node failures, computational latency, and small memory capacity in IoT sensing nodes. This article provides reliable predictive analytics with the optimisation of data transmission characteristics in StreamRobot. Software-defined reliable optimisation design is applied in the system architecture. For the IoT implementation, the edge system model formulation is presented with a focus on edge cluster log-normality distribution, reliability, and equilibrium stability considerations. A real-world scenario for accurate data streams generation from in-built TelosB sensing nodes is converged at a sink-analytic dashboard. Two-phase configurations, namely off-taker and on-demand, link-state protocols are mapped for deterministic data stream offloading. An orphan reconnection trigger mechanism is used for reliable node-to-sink resilient data transmissions. Data collection is achieved, using component-based programming in the experimental testbed. Measurement parameters are derived with TelosB IoT nodes. Reliability validations on remote monitoring and prediction processes are studied considering neural constrained software-defined networking (SDN) intelligence. An OpenFlow-SDN construct is deployed to offload traffic from the edge to the fog layer. At the core, fog detection-to-cloud predictive machine learning (FD-CPML) is used to predict real-time data streams. Prediction accuracy is validated with decision tree, logistic regression, and the proposed FD-CPML. The data streams latency gave 40.00%, 33.33%, and 26.67%, respectively. Similarly, linear predictive scalability behaviour on the network plane gave 30.12%, 33.73%, and 36.15% respectively. The results show satisfactory responses in terms of reliable communication and intelligent monitoring of node failures.

1. Introduction

The term ‘intelligent service-care robots’ (ISCRs) or ‘StreamRobots’ refers to a purposeful machine with a built-in intuitive capacity to gather information streams from both external and internal environments, while using reliability optimisations (e.g., domain knowledge) to carefully deliver service tasks. In context, TelosB nodes refer to low-powered open-source modules that support experiments with universal serial bus programming, IEEE 802.15, an integrated radio antenna, as well as a low-power scaled processor running on TinyOS 1.1. This is useful in robotic computing (RC) which is one of the most well-known methods of bringing a synergistic interaction of robots with challenging work environments. Despite its usefulness in task provisioning, little effort has been committed to its physical parameter optimisation for reliable data transmission in StreamRobots.

Within manufacturing/Industry 4.0, transportation, energy systems, and health care delivery, among others, service robots may use well-calibrated parameters based on RC to solve complex problems [1]. For instance, ISCR was massively applied to combat the COVID-19 pandemic, e.g., by taking care of elderly and sick people [2]. These ISCRs are directly engaging human users to solve complex problems. This is because interactions that foster human-friendly relationships with ISCR are encouraged for complex services [3]. In this case, StreamRobots incorporate social-cognitive features of humans and can be deployed in complex environments such as hospitals and open places [4]. Specifically, there are major concerns in IoT sensor-powered robots for pre-

* Corresponding author.
E-mail addresses: kennedy.okafor@futo.edu.ng (K.C. Okafor), omowunmil@uj.ac.za (O.M. Longe).

https://doi.org/10.1016/j.heliyon.2022.e09634

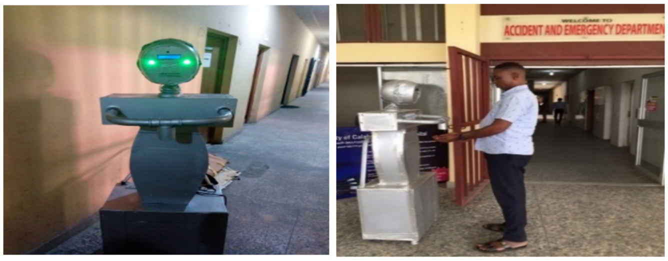


Fig. 1. a-b. StreamRobot deployment at the University of Calabar Teaching Hospital, Nigeria (Source: Author's design prototype version 1.0). a) StreamRobot at the Accident and Emergency Unit. b) End-user interaction with the StreamRobot.

dictive vital signal scanning (PVSS), especially for infected people [5]: These are reliable signal transmission (RST) and resource conservation.

Considering the former: RST in the robots requires smart optimisation. The reason is that ISRs drive the democratisation of complex processes that advance robots to provide human support and assistance. Second, resource conservation is a key issue due to prolonged outdoor exposure, even in StreamRobots. As such, a reliable IoT service robot for the COVID-19 PVSSS must have the predictive capability to detect vulnerabilities and absorb parametric deviations with high accuracy. Such robots should be able to assist healthcare experts in offering treatments to patients in isolation. In this case, a probability distribution that offers a continuous randomised variable could be used for efficient parameter determination in a service robot.

In both cases, the adaptation of artificial intelligence (AI) into robotics will handle complex issues in mission-critical applications. For example, the use of ISCRs in many sectors lowers production costs (i.e., operation expenditure and OPEX) while increasing output productivity. Monotonous and repetitive tasks make ISCR ideal since AI seamlessly replaces human agents, thereby yielding improved outcomes. According to the International Federation of Robotics (IFR), production automation is increasing globally. There are 74 robotic units per 10,000 employees and this is the recent average of global robot density in the manufacturing industries. Given the daily improvements in application and capabilities, critical sectors such as healthcare are constantly leveraging robots for various computational tasks. This is found in complex surgeries, clinical training, medicine dispensing, and personal care, among others.

Therefore, using ISCRs fused with computational SDN procedures, makes it possible to achieve functions such as identifying infected patients, disinfecting, cleaning up, and even delivering drugs to patients under treatment [6]. In most cases, the use of transactional analytics helps in getting quick results, especially the vitals. As such, explicit sensing operations and physical signal characterisation of sensor nodes can assist in deriving coarse-grained solutions in dynamic TelosB based applications [7]. This leads to reliable datasets for analytics within indoor and outdoor deployments. For lightweight computation, software-defined methods can be used to process vital parameters from IoTs.

In a current work, shown in Fig. 1a-b, the StreamRobot was designed and deployed at the Accident and Emergency Unit of the University of Calabar teaching hospital in Nigeria. The system had an AI health vita-signal monitoring system and combines computer vision (CV) with infrared scanning to determine patients' forehead temperature and other vitals via wireless signalling. It was deployed to combat the COVID-19 pandemic and provide support for frontline health workers. The optimisation of the data transmission characteristics in the StreamRobot, using the software-defined technique, is the major focus of this article.

As part of the design requirements, data gathering is achieved with IoT integrations. However, to use these nodes, its energy metrics for transmission, forwarding, and reception of data streams through wireless links need reliability optimisation. In this case, scheduling algorithms such as semi-dynamic and dynamic schemes can be implemented in the TinyOS environment, housing TelosB nodes. Such an example can be found in environmental cyber-physical systems [8]. Once deployed, energy management operations must be satisfied to ensure continuous data-streaming into the cloud sink. This means that with reliability infusion in IoT-based autonomous robots, mission-critical services can achieve significant control within its deployment context.

For StreamRobots, node reliable connectivity is very important, since any delay or node failure can dislodge the system from receiving processed data. Additionally, disconnections can lead to the malfunctioning of the sensing communication network. This is unacceptable in such a mission-critical environment. Most works have attempted to fix this concern using node redundant schemes. In this case, new nodes are introduced where existing StreamRobot-nodes have failed. This appears to be counter-productive, especially in the COVID-19 era where service robots need minimal human intervention. As such, this may not be viable.

Another sensor optimisation scheme focussed on a wireless sensor network (WSN) relocation for connection restoration [9]. This is because WSN is still commonly used in hazardous terrains such as battlefields, power plants, dense forests, and similar areas. The restoration of the network in these areas is very difficult. Therefore, network restoration can be possible when the traffic is localised.

Ultimately, connectivity and coverage in IoT service network (ISNs) are critical when considering the overall network lifetime within outdoor and indoor environments. As observed in IoT sensor nodes, battery power, CPU execution, and connection resources are limited. Therefore, a huge number may be needed to effectively meet demands in selected domains and improve data gathering with reliability. During the deployment of ISNs, a neighbourhood connection is established so that these nodes can execute tasks (i.e., transmit and forward data streams) to the uplink sinks.

However, the reliability of sensor nodes is a major factor in the service robot. Besides, link outage, which affects data stream propagation, is always unacceptable.

Therefore, this article presents a reliability model for StreamRobots (for IoT-TelosB sensor nodes) that uses link-state-on-demand characterisation infusion (LS-ODCI) to perform data stream uploads in two phases, namely off-taker (OT) and on-demand link-state (ODLS). The former generates data traffic while the latter has the cluster head aggregator whose role is to move data stream into the analytics server via software-defined controller interfaces. The entire baseline IoT node that generates data traffic, forms the OT and has a neighbourhood list (NL) or lookup table (LUT) for node identifiers (IDs). The ODLS has the cluster head aggregator whose role is to move data stream into the analytics server. The ODLS node is a very sensitive and dynamic node with expandable compute, storage, and power resources. The OT and ODLS are fully heterogeneous (i.e., some have more capabilities concerning computing resources and power drain) for data stream transmission or reception. At all times, the ODLS monitors the broadcasts from OTs and sets an energy equilibrium using congestion or beacon feedback notification. Failure nodes are detected by the ODLS as the aggregator's head and corrected by the orphan reconnection trigger mechanism (ORTM). Such failures may be due to localisation impairments at the node level. Through its recycle monitoring, it offers a lower overhead and extended lifespan via optimal battery/energy utilisation. Finally, the various reliability algorithms are introduced for StreamRobot functionality. The considered metrics include a link quality indicator (LQI), a received signal strength indicator (RSSI), battery life, and throughput.

The main contribution of this article is highlighted as follows:

i. To propose a StreamRobot edge plane system architecture for data acquisition that integrates edge analytics with the IoT TelosB nodes. Further, the work aims to reduce the memory-overhead due to the storage constraints in IoT devices.

ii. To propose log-normality probability distribution for StreamRobots with interface IoT WSN.
iii. To create a resource-conservative algorithm that is coverage-aware with a connectivity convergence scheme. This makes use of a dynamic transmission for baseline service nodes, while removing failures in complex deployments.
iv. To resolve boundary problems using the reliability orphan-node relive algorithm.
v. To show an optimisation reliability protocol termed SDN-intelligence, for StreamRobot IoT data transmission. The scheme executes data and control layers to save the orphan nodes while ensuring that the forwarding layer uses its engine to process the received broadcasts. In this case, every received signal is pushed to the sink-SDN node for topological database build-up for reliable route selection.
vi. To demonstrate the proof-of-concept with the OpenFlow SDN-IoT hardware prototype and machine learning validations.

The novelty of the proposed work is found in the StreamRobot reliability optimisation for multi-connectivity among its sensor nodes. The schemes are based on the advantages of AI and SDN. The log normality theory promotes system stability, leading to standardisation and operationalisation. The major attribute of the proposed approach is the layered computational structure based on the ideals of principles of IoT-SDN, edge computing for system orchestration, and streams computing. These schemes provide support for decentralised network provisioning. Agility and flexibility gave rise to improved parametric signalling and versatility in the OpenFlow SDN-IoT hardware prototype.

To the best knowledge of the authors, this research represents the first practical effort to address the issue of simultaneous IoT node characterisation, using both analytical and applied testbeds with TelosB-IoT nodes. Explicit analytical relations describing the TelosB sensor node-system reliability constraints are derived. The main objective of this article is to optimise data transmission characteristics.

Section 2 focusses on the existing literature about this study. Section 3 presents the system model for edge log normality distribution. Section 4 describes the field TelosB deployment specifications. Section 5 presents the OpenFlow SDN integration architecture. Section 6 presents the deployment system testbed. In Section 7, a StreamRobot fog detection cloud predictive machine learning model (FD-CPMLM) is discussed. Section 8 presents the hardware prototype. Finally, Section 9 concludes the article.

2. Literature review

2.1. Service robots

There are current efforts on sensor applications in service robots. For instance, the IoT blockchain integration on TelosB MSP430 platform CC2420 IEEE 802.15.4 radio interface has been developed for reliable data transmission [7]. In terms of signalling, research efforts have been carried out to determine LQI, a channel prediction model, and RSSI in an outdoor environment [10]. The authors [11] applied TelosB sensor nodes in web services for battlefield management systems. The article [12] discussed single-chip nodes for autonomous node programming over USB model design. In [13], medical sensor networks (MSN) were developed for e-healthcare systems, considering trust management in TelosB nodes where collection tree protocols were applied. In [14], an experimental study on the ZigBee frequency agile (FA) scheme was implemented on the TelosB testbed via test case experiments. In [15], an accuracy-aware diffusion process mapping scheme was developed with smart aquatic mobile sensors – TelosB nodes. In their work, scheduling movements were introduced while validating their profile accuracy with TelosB nodes.

2.2. System optimisation algorithms

In developing intelligent service robots, various contributions to the optimisation scheme are studied. The authors [16] have discussed a hybrid technique capturing particle swarm optimisation (PSO) and bacterial foraging optimisation (BFO) for a multi-machine power loading design system stability. The article [17] proposes BAT search algorithm (BSA) for the optimal design of power system stabilisers (PSSs) in a multimachine environment. The optimisation problem is solved, using BSA under parameter tuning. The authors [18] introduce the Cuckoo search (CS) algorithm for an optimal power system stabiliser (PSS) design in a multimachine power system. The PSS parameter tuning problem is formulated as an optimisation problem that is solved by a CS algorithm. The article [19] presents ant colony optimisation for the identification of the number as well as locations of IoT relays within QoS thresholds.

In [20], the authors propose the Bettle antennae algorithm (BAA) in wireless sensor networks for jamming attack points. The authors [21] focus on a new approach, deployed for human motion-denoising via a joint optimisation of kinematic and anthropometric constraints. This is considered as noisy-wave skeleton data meant for depth-sensor-based motion capture (D-Mocap). The captured data are usually error-prone, outliers, and distorted. The article [22] proposes an energy-aware and trust-based routing protocol for WSNs, leveraging adaptive genetic algorithms for resisting routing attacks and minimising energy consumption triggered by data transmission. The authors [23] focus on a topology optimisation method (TOM) needed for precise multi-axis inertia sensing needed in self-positioning autonomous robot control. The work [24] explores a soft sensing optimisation scheme for detecting obstacles and discriminating the scalable obstacles using an untethered miniature, soft, C-legged robot, M-SQuad, the first modular C-legged quadruped consisting of three modules. The authors [25] have developed an optimisation model, addressed with an elitist preservation genetic algorithm (EGA). There have been some optimisation techniques and research efforts in real-time monitoring of signals within and outside healthcare systems. Most are found in service-robot monitoring models discussed in robotic computing repositories, for instance, the service-robot types (SRT), industrial robots, human-aware robots, AI-assisted smart robots, high-speed industrial robots, and robot frameworks [26]. This work compares with selected reliability optimisation schemes discussed in Section 6.

2.3. Research gaps

i. Existing literature suggests a lack of discussion and findings at the IoT sensor node log normality distribution, especially in TelosB service robotic applications.
ii. None of the research focusses on continuous traffic monitoring, such as real-time scanning of patients for COVID-19 vulnerabilities.
iii. Most efforts fail to identify optimal operational metrics in sensor activities.
iv. There are very few works on real-time information gathering, using IoT nodes for indoor data acquisition with zero error deviation, especially in mobile robots deployed in cities during COVID-19.
v. There are non-existent IoT open-source TelosB nodes used in StreamRobots for the estimation of physical node characteristics from the existing studies.
vi. Orphan node problems are yet to be fully addressed in complex networks. However, for replication of a typical TelosB system, the details of IEEE 802.15.4 characteristics (compatible IoT open-source TelosB) are needed for classical experimental studies [27, 28, 29]. This can be used to estimate temperature, relative humidity, and light parameters in the ISCR. The next section will discuss the system architecture and mathematical background for the implementation testbed.

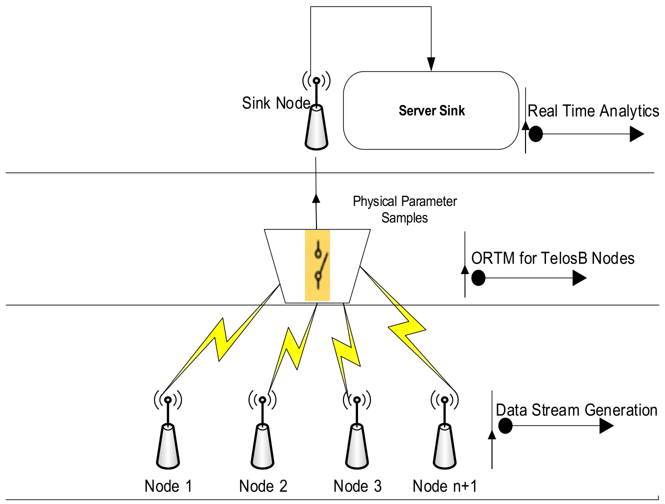


Fig. 2. StreamRobot edge sub-system layout.

3. Edge system model formulation

3.1. Edge cluster log-normality distribution model

In this section, log normality distribution (LND) is described for StreamRobot TelosB sensor node clusters with buffers storing the generated data streams, as shown in Fig. 2.

LND is the continuous probability distribution of a random IoT node variable whose logarithm is normally distributed. This is adapted into the sensor nodes for stream end-to-end data transmission. In Fig. 2, the IoT sensor node data-streams reliability and the payload is constrained by the mean and standard deviation parameter to the sink server where analytics takes place. For random stream variables, it has implications on the traffic parameters of StreamRobots, such as battery life, LQI, RSSI, and transmission throughputs in ISCRs. Any deviation from the expected patterns (or both the parent and orphan nodes) may lead to predictive errors. For the edge clusters, transmission reliability is very important. The baseline sensor nodes (OTs) provide active data stream propagation into the ODLS (head aggregator). An efficient optimisation algorithm can reliably transmit data streams into the server sink through the SDN clusters.

3.2. StreamRobot reliability model

Consider a lognormal distribution for edge nodes $X_1, X_2, X_3 \dots \dots \dots X_{\mathrm{n+1}}$ in StreamRobots whose traffic strength of data streams is given as $S_{t,}$ and its payload is given as L_p. Both are constrained to a sink logger (cf. Fig. 2).

For a scenario of continuous data streaming from edge devices, there is a need to derive a reliability function for such distribution. Therefore, Eq. (1) is used as the baseline reliability model, which is theoretically defined as the probability of success at time t, which is denoted $R(t)$ [30].

$$R(t) = Pr\{T > t\} = \int_{t=0}^{\infty} f(x)dx \tag{1}$$

where $f(x)$ is the failure probability density function and t is the length of the period (which is assumed to start from time zero).

In the case of Fig. 2, for a new random variable $X_{T,}$ $R(t)$ is given as the ratio of traffic strength $S_{t,}$ to data stream payload L_{p} [30]. Let us now define the intrinsic reliability of a new random variable X_{T} (i.e., edge node) as the ratio of traffic strength $S_{t,}$ to data stream payload $L_{p,}$ hence, using (1), which gives Eq. (2). Complete characterization of the system reliability

$$R(t) = Pr\{T > t\} = \int_{t=0}^{\infty} f(x)dx = \frac{S_{t,}}{L_{p,}} \tag{2}$$

This then gives (3):

$$Ln\, R(t) = ln\, S_{t,} - \ln L_{p,} \tag{3}$$

However, $Ln\, R(t)$ can be observed to be normally distributed, since both $ln\, S_{t,}$ and $lnL_{p,}$ are also distributed normally. Now, the edge reliability of the TelosB nodes can be expressed as Eq. (4):

$$R(t) = Pr\left(\frac{S_{t,}}{L_{p,}} > 1\right) = Pr(X_{T,} > 1) = \int_{t=0}^{\infty} f(x)dx \tag{4}$$

Since $Ln\, R(t)$ is normally distributed, $X_{T,}$ follows a lognormal distribution. Hence, it is feasible to define a standard normal variate Z using Eq. (5) as

$$Z(t) = \frac{lnx - \mu lnx}{\sigma_{in}x} \tag{5}$$

where μlnx and $\sigma_{in}x$ are the mean and standard deviation of $Ln\, X_{T, \dots\dots} (X_{n+1,})$ respectively.

If Eq. (5) is rewritten in terms of Z, the new limits of integration can be defined such that when $x = 1$, $Z(t) = Z_1 = \frac{ln\mathrm{x} - \mu lnx}{\sigma_{in}x} = \frac{0 - (\mu lns - \mu lnL)}{\sqrt{\sigma^2_{in_S}} + \sigma^2_{in_L}}$, and when $x = \infty$, $Z = Z_2 = \infty$. The random variable $X_{T,}$ is the sensor node with normal distribution, while x is the instance of the failure probability density function of $X_{T,}$. Therefore, Eq. (4) can be rewritten in terms of Z using Eq. (6):

$$R = \int_{Z_1}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-1/2x^2} dz \tag{6}$$

where

$$Z_1 = \frac{-(\boldsymbol{\mu lns} - \boldsymbol{\mu lnL})}{\sqrt{\boldsymbol{\sigma^2_{in_S}} + \boldsymbol{\sigma^2_{in_L}}}} \quad \text{for } T > t \tag{7}$$

Eq. (7) gives the bounded Z from Eq. (5). Now, if $X_{T,}$ is log-normally distributed, its probability density function is given by Eq. (8).

$$f(x) = \frac{1}{\sqrt{2\pi\sigma x}} exp\left\{-\frac{1}{2}\left(\frac{Lnx - \mu}{\sigma}\right)^2\right\} \quad \text{for } X_T > 0 \tag{8}$$

where $\mu = \in (Lnx) = \mu_{in}x$, $\sigma = \sigma_{in}x$ and the variable Lnx are normally distributed. By defining $Y = Lnx$, it is feasible to obtain the expected value (edge data stream) and the standard deviation of x as Eq. (9):

$$E(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} exp\left\{-\frac{1}{2}\left(\frac{Lnx - \mu}{\sigma}\right)^2\right\} dx \tag{9}$$

Since $x = e^y$ and $dy = \frac{1}{x}dx$ or $dx = xdy$, then (9) can be written as Eq. (10):

$$E(x) = E(e^y) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^y exp\left\{-\frac{1}{2}\left(\frac{Lnx - \mu}{\sigma}\right)^2\right\} dy \tag{10}$$

Then (10) can be further simplified concerning (2) and given by Eq. (11):

$$E(x) = exp\left(\mu + \frac{\sigma^2}{2}\right) \tag{11}$$

The computation of σ_x requires the evaluation of $E(x^2)$ and is represented in Eq. (12):

$$E\left(x^2\right) = E\left(e^{-2y}\right) = \int_{-\infty}^{\infty} \frac{e^{2y}}{\sqrt{2\pi\sigma_x}} exp\left\{-\frac{1}{2}\left(\frac{Lnx - \mu}{\sigma}\right)^2\right\} dx \tag{12}$$

For the edge plane, (12) can be further simplified to (4), using Eq. (13):

$$E\left(x^2\right) = \exp\left\{2\left(\mu + \sigma^2\right)\right\} \tag{13}$$

Therefore, the estimated variance of the edge node clusters, X_{n+1} is given by Eq. (14):

$$\sigma_x^2 = E(\left\{\left(x - E(x)\right\}^2\right) = E(x)^2 - \left\{E(x)\right\}^2$$

$$\sigma_x^2 = exp\left\{2\left(\mu + \sigma^2\right)\right\} - exp\left\{2\left(\mu + \frac{\sigma^2}{2}\right)\right\}$$

$$\sigma_x^2 = exp\left(\mu + \sigma^2\right)(\exp\left(\sigma^2\right) - 1 \tag{14}$$

This means that Eq. (11) and Eq. (14) can be resolved to find μ and σ^2 as presented in Eq. (15):

$$\mu = LnE(x) - \frac{\sigma^2}{2} = Ln\mu_x - \frac{\sigma^2}{2} \quad \text{and}$$

$$\sigma^2 = Ln\left[\frac{\sigma_x^2}{(E(x))^2} + 1\right] \tag{15}$$

From a linear programming perspective, the objective function in StreamRobots is to optimise data transmission reliability and avoid losses based on well-selected constraints and its relation map between the decision variables. In the StreamRobot, the constraints refer to capacity, availability, resources, technology, etc., and reflect the limitations of the business environment.

From Eq. (15), the reliability of the traffic strength $S_{t,}$ and its payload $L_{p,}$ are constrained by the mean and standard deviation to a sink-logger. This optimisation derivative has implications on traffic delivery and can affect the performance of StreamRobots in their deployed environments. For instance, with a limit of very low value, this will imply that the values transmitted will be very close to the mean or expected value, eliminating the need for parity checks. With a lower payload reduction, a transmission efficiency will be guaranteed. To achieve LND for random OTs distributed at scale, an exponential function is needed for only positive real nodes in Eq. (15).

3.3. StreamRobot equilibria and stability

Let us assume a contextual case of an autonomous linear discrete model of the form Eq. (16). This leads to joint kinematics optimization described [21].

$$U_n = aU_{n-1} + b(a \neq 1) \tag{16}$$

where a and b are constant coefficients in the StreamRobot. If U^* is the equilibrium solution of the model, then

$$U_n = U_{n-1} = U^*$$

$aU^* + b = U^*$, therefore Eq. (17) holds.

$$U^* = \frac{b}{1-a} \tag{17}$$

The equilibrium point U^* is said to be stable if all the solutions of $U_n = aU_{n-1} + b$ approach Eq. (17) as n becomes large ($\rightarrow \infty$). The equilibrium point U^* is unstable if all solutions (if it exists) diverge from U^* to $\pm\infty$. The stability of the equilibrium solution U^* of the equation $U_n = aU_{n-1} + b$ depends on a. For the reliability test conditions in Eq. (17), it is stable if $|a| < 1$ and unstable if $|a| > 1$.

As depicted in Fig. 2, the TelosB CC2420 sensor node with ORTM is used to collect real-world data, as discussed in Section 5. By increasing the number of edge nodes in the topology, packets are easily forwarded to the next hop by using a tree routing mechanism. In this case, the ORTM computes the packet destination address, using its distributed Internet protocol (IP) address assignment mechanism. However, in a very complex node deployment, network parameter constraints may trigger an IoT orphan node problem (IONP). Orphan nodes are devices that are unable to obtain network addresses and are thus disconnected from the network. This is the only major limitation of complex node deployment. The algorithm is used to describe how the TelosB node uses link-state-on-demand characterisation infusion (LS-ODCI) application Programming Interface (API) to process the wireless parameters. This is used to transfer data from the edge to the sink where higher analytic processing takes into account the small memory space, energy life, node failure, and quality of service requirements.

The input and output variables were carefully scheduled to optimise traffic transmission. This is fully coordinated through machine-node to machine-node communication. Unlike in most works, the parametric variables are processed within the QoS thresholds without draining the battery quickly.

Algorithm 1 Log normality TelosB integration API with LS-ODCI.

```
Inputs:      R(t), OT( );, LQI ( );, RSSI ( );, Batterylife ( );, Throughput ( );
Output:      Sink_ODLS ( );
Begin ( )    LS-ODCI ( );
Parameters:    SensorRouterC_parameters_weight←Set;
               // Node parameters Ready
               configuration. SensorRouter AppC { }
int i←0;
While i < SensorRouterC_monitorCallSchedule do
        Metrics_historyItem←HistoryList.get (HistoryList.Size) ( );
        CallEnergymodel ( );
        NodeTxDataMoving←SenorRouterC (Container history);
        totalSensorRouterC_weight ←total NodeTxDataMoving_weight;
        i++;
end while
 Receive.receive(message_t* msg, void *payload, uint8_t len) {
 NodeTxnData* in = (NodeTxnData*) payload;
             NodeTxnData* out; //Ophan reconnection trigger mechanism
             in->sensorNodeparameter = call CC2420Packet.
             getparameter (msg);
Return
```

Considering the mathematical perspective offered by the objective function in Eq. (18) to maximise transmission reliability, as X_i is the designated set of activities in the StreamRobot. i precisely indicates the activity set in active operation from the 1^{st} to the n^{th} state. This also gives the activity list of actions in the state transition. Added to that, C_i is the compounded streams aggregation payload parity control that activity i generates (i.e., the 1^{st} to the n^{th}). This optimization model [30] is enhanced in Eq. (18)

$$Max\,f(x) = \sum_{i=1}^{n} C_i X_i(\mu ln x, \sigma_{in} x) \tag{18}$$

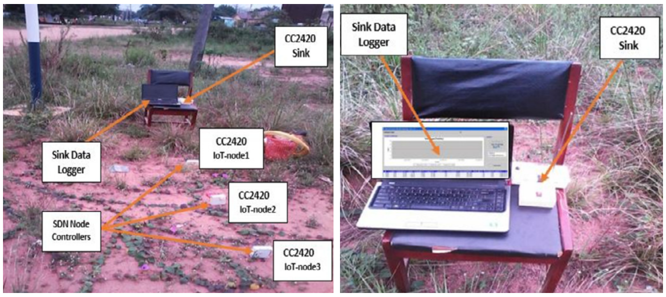
To maximise data transmission reliability in StreamRobots:

- c_i is the coefficient that matches the ith variable.
- X_i is the ISCR ith decision variable.
- \Subject to reliability test conditions in Eq. (17).

4. Field sensor node deployment

4.1. TelosB testbed characterisation

This section describes the testbed design parameters of the TelosB device used in the StreamRobot deployment. An experimental task is used to characterise the device setup while gathering key para-meters needed for the service sensor edge analytics system. Using previous details in [28] and [29], a discussion on the sensor node design and deployment is highlighted in Section 5. To validate the ISN reliability model, four TelosB nodes were used to facilitate the test measurements. This was because of cost and convenience at the time of this study. However, the setup was sufficient for the data acquisition needed for this study, but it can be scaled up, depending on the deployment environment. A demonstration of how the sensors could rapidly acquire physical parameters and then transmit the data streams into the sink, is



a) CC2420 Edge node clusters b) Sink analytics

Fig. 3. A log-normality FUTO outdoor testbed for edge analytics with sink nodes.

shown. This was done regardless of the data size and the transitional kinetic energy needed to forward the data to the server. The experimental setup process/testbed design parameters and deployments are discussed in section below.

4.2. Optimal addressing scheme

In this section, the optimal addressing scheme for reliable parametric transmission is discussed. In this case, the IEEE 802.15.4 standard is used. This has a TI-MSP430F1611 microcontroller and Texas instruments CC2420 RF chip (i.e., IEEE 802.15.4 2.4 GHz wireless module). It uses a TinyOS 2.x that is compatible with a USB interface. The 48Kb memory is then used with a 12-bit ADC, having eight channels. In this work, the USB interface is attached to the sink server for telemetry captures.

For the StreamRobot testbed architecture in Fig. 3a, four nodes (CC2420 TelosB nodes) are considered for ease of configuration and simplicity per domain site. Fig. 3b depicts the sink analytics for data aggregation from the ISCRs/StreamRobot. The design of the addressing scheme for OTs and ODLS (including the server sink) leveraged a scheme known as classless inter-domain routing (CIDR) [31]. Similarly, stateful Multi-Index Hybrid Trie (MIHT) can be used for a network scale of any size [32]. Hence, CIDR and MIHT achieve scalability, route aggregation (summarisation), dynamic updating, lower administrative distances, split horizon, and route poison control. These are very important in the system design.

The CIDR IP address assignment policy introduced may be conservative for a very large-scale deployment. This could result in a poor utilisation of the available pool in a subnet for such complex deployments. IoT nodes that could fail to receive real-time network addresses will be disconnected from the service network, making them orphan nodes. This boundary problem is resolved, using the reliability orphan-node relive algorithm (RONRA). This is introduced to boost the reliability of data transmission and effectively reduce such problems as shown in Fig. 3a and 3b. Fig. 3a shows the physical testbed with linearised IoT node clusters, communicating with a sink logger in Fig. 3b (i.e., sink analytics). A reliability optimisation is built into Algorithm 2, considering Fig. 3a-b. It shows a reliability node recovery scheme called the orphan reconnection trigger mechanism (ORTM). This is based on an improved probability distribution function which includes multi-node instances (MNIs) in-service field deployments. The ORTM selects nodes with active IP addresses to connect similar node instances or sink nodes. This is done to scale up connectivity in the node-to-sink network. It effectively leads to reduced cases of IoT orphan nodes, especially in complex deployments.

Algorithm 2 shows a more complex analysis. It shows both orphan and parent nodes established for seamless communication. From the orphan path, parent nodes are checked in the broadcast range. Once satisfied, it requests for the get-IP address of all nodes. When all the IP addresses are retrieved, it then compares the mean and standard deviation (SD) for the best node ID. After confirmation, a joint request is sent to the parent node with the largest SD. The parent nodes, in all instances, will check for their neighbours, send IP get-requests, and execute the algorithm. As such, an association established with the orphan node and parent nodes is made. This takes care of the case of an increased number of nodes on the robot.

Algorithm 2 Log normality TelosB integration with ORTM.

```
Inputs:        R(t), total number of IoT nodes on site, node-
               parent; ($N_p$ parent), Node.Array,
Output:        Sink_ODLS // parent node terminal for rounds
Begin ( )      $N_p$ parent ← remaining capability
               // outstanding capability of the parent
Parameters:        Define_Array.SensorRouterC_pa-
                   rameters ← Set; // Node array
                   Ready
                   Configuration.SensorRouterAppC { }
int I ← 0;
While i < SensorRouterC_monitorCallSchedule do
        Cal_inf (Num,$N_p$_Node.parent,Array.Sensor)
        ← HistoryList.get(HistoryList.Size ();
        for (int i = 0; i < Num; i++);
        if (node [i].parent = = $N_p$ parent && node[i].type = = ZR
        i! ← $N_p$ parent;
        for (node [i].MaxDepth > 0);
        Cal_inf (Num,Next,Node.Array.$N_p$parent
        ←HistoryList.get(HistoryList.Size ( );
        Else
        $IP_{parent}$ = $N_p$parent;
        if (node [i].parent = = $N_p$ parent && node[i].type = =
        Sink_ODLS;
        $IP_{parent}$ = $N_p$parent;//
  end while
  Receive.receive (message_t* msg, void *payload, uint8_t len) {
  NodeTxnData* in = (NodeTxnData*) payload;
  NodeTxnData* out; //Ophan reconnection trigger mechanism
  in->sensorNodeparameter = call CC2420Packet.getparameter(msg);
  Return $IP_{parent}$
  End for
  End
  Return Sink_ODLS
```

The experimental setup employed a reliable field characterisation of the IoT nodes for optimal data transmission. This is because the reliability of the traffic/data streams strength S_t and its payload $L_{p,}$ is expected to be constant in an on-demand fashion. After developing the IP mapping, static and dynamic IP mapping were explored for the main deployment. The generic dynamic host configuration protocol (DHCP) of the sink-based station was used to allocate addresses, thereby optimising administrative costs/overheads. In the testbed, a combination of CIDR and MIHT gave scalable inter-domain routing addressing scheme (SIRAS) for Fig. 1 for all the TelosB nodes (i.e., Node 100, 200, 300, 400, $n+1$).

Now, assuming that a disposable IP block of 192.168.10.33/27 is needed for the service robot nodes, the network address can be given as 192.168.10.0, since this is a Class C address. Then, the subnet mask S_m is given by 255.255.255.244 (since the CIDR value of/27 = 244). The number of valid nodes for site_1, v_N_h are given by $2^N = 2^5 - 2 = 30$ nodes where N = off bits. The number of subnets for Site_1, N*site_1* are given by $2^X = 2^3 = 8$ subnets where X = on bits. The block size or valid subnets is given by $2^8 - 244 = 32$ i.e., [0 32 64 96 128 160 192 224], hence 8 subnets. The IP map identification subnet for the CC2340 TelsoB nodes is derived as shown in Table 1.

Hence, for site_1, the valid range for CC2420 is 192.168.10.**33**–192.168.10.**62** (subnet 32). In site_2, the valid range for CC2420 is 192.168.10.**65**–192.168.10.**94** (subnet 64). Additionally, in site_3, the valid range for CC2420 is 192.168.10.**97**–192.168.10.**126** (subnet 69). Hence, any address outside the valid node range becomes either a subnet or broadcast address which is unassignable in this design. In this

Table 1. Reliable log normality IP subnet mapping.

Subnets	0	32	64	96	128	160	192	224
First _ CC2420 IoT node	1	33	65	97	129	161	193	225
Last _ CC2420 IoT node	30	62	94	126	158	190	222	254
Broadcast CC2420 IoT node	31	63	95	127	159	191	223	255

work, broadcast addresses were used to register the initial presence of sensor nodes in the coverage deployment field. The remaining IP blocks from subnet 128 to 224 were not utilised because only a few sensor nodes were considered. However, this has no significant impact on the IoT node's baseline performance, considering the ORTM.

4.3. OutDoor testbed description

The Cisco LAB of the Federal University of Technology, Owerri (FUTO), situated in South-East Nigeria (FUTO-Cisco LAB), was used as the testbed for parametric outdoor scanning as shown in Fig. 3. The StreamRobot monitoring follows the LND, described previously. It accounts for both indoor and outdoor environmental deployments. The first testbed is the outdoor setup phase with the geographical coordinates (6° 14' 0" north, 7° 17' 0" East). In this testbed, four TelosB sensors were used in the experiment by calibrating the nodes at various distances, while attaching one of the sensor nodes to a laptop that served as the sink via an ODLS shown in Fig. 3a and 3b.

The remaining three sensor nodes were placed at 0°, 90°, and 180° from the sink with equal distances of 10 metres apart. The measurements were taken at several distances every 1 minute, and the data obtained were recorded. The nodes are assigned to ID 100, 200, 300, and 400 respectively.

The second testbed shown in Fig. 4 is an indoor setup, located at the School of Engineering building complex's overhead control corridor. Because of cost and convenience, these four-sensor nodes were deployed for data capture. However, the four-node setup was sufficient for the preliminary data acquisition needed for the study. One of the sensor nodes was attached to the server laptop as the sink while the other nodes were placed in the first testbed, as measurements were taken from the sink-analytics in Fig. 4.

The TelosB sensor node's characteristics are highlighted in [28] and [29]. The main composition of the node includes the CC2420 transceiver, a microcontroller with 3.3V CMOS compatible batteries. The ranges of the parametric variables such as temperature, humidity, and visible light sensors are highlighted. As shown in Fig. 4, the TelosB nodes have a smart USB slot with a programmed API driver in the Core i7 server machine, and TinyOS in DOS emulation mode maps with Cygwin libraries. With high-level TinyOS 2.x-Java GUI sockets developed on the IDE, the final API code was used to acquire real-time data as shown in Fig. 4, using Algorithm 1. Fig. 4b shows how the Sink SIRAS configuration palette is used to visually setup various nodes such as Node 100, 300, …, $N+1$, etc., regardless of their location (indoor or outdoor). Cases of TelosB captured parameters for analytics on the node cluster head (N#100), while node 200, 300, etc. were obtained as indicated in Fig. 4.

Using the setup shown in Fig. 4, three case scenarios involving IoT-TelosB captured parameters for analytics were determined. These include temperature, light intensity, humidity, RSSI, LQI, and frame sizes. It was used to determine indoor deployment locations for the IS-CRs.

4.4. Outdoor results analysis

In this section, the observations from the previous experimental testbed are discussed. From Fig. 5, the LQI depicts the amount of data stream packets/bytes received on the sink. For the COVID-19 ISCRs, moving data streams have implications on the LQI. As the placement distance increases, the LQI continues to drop. In the case of LS-ODC, network topological changes will drop the LQI, thereby showing log-normality distribution observation for the available nodes. The LQI status for all the attached nodes determines connectivity pathways. Any path with a satisfactory LQI profile is optimised to deliver messages/data streams to the sink reliably. Hence, Fig. 5 shows that to deliver data streams to the sink servers, the energy efficiency of the TelosB nodes must rely on defined short distances. Therefore, for the battery-powered nodes, a uniform trend was observed for LQI, starting from 105.5 dBm, considering the various placement distances. This implies that the measured and captured data streams under LS-ODC reached the sink server for possible analytics.

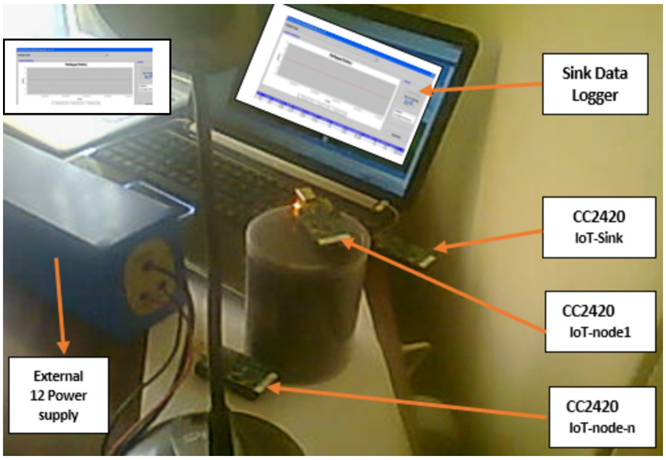


Fig. 4. TelosB Indoor Calibration with Sink SIRAS configuration palette.

Fig. 6 shows the obtained RSSI behaviour under LS-ODC. It depicts a considerable measure of data stream signal quality on the fly. For energy conservation during traffic routing, path selection can draw enormous power. Hence, the energy model positively impacted RSSI, using Algorithm 1. The total energy received for the physical TelosB nodes is −40 dBm. It was observed that given the various placement distances, the RSSI appears to be dropping as the node distances increase under the influence of LS-ODC. This means that the service robots need to be very close to the targeted objects to avoid a loss of signal with the central server.

This implies that the service-care robot's (SCR) RF can receive signals based on the connectivity profiled with RSSI. This reveals the true power level of the TelosB node when receiving traffic or when its sink head aggregator receives a similar data stream. As a thread-off, the throughput can be seriously impaired at a very low RSSI. From the measurement testbed, the various battery profiles (i.e., energy conservation usage) are highlighted in Fig. 7. It is important to note that a CMOS electrochemical battery (AA 3.3 V) was used to power the TelosB sensor node in the testbed. Clearly, with the LS-ODCI, the wake-up status with active power is made very small, thereby reducing the energy consumption and its discharge characteristics of TelosB nodes. The implication is that the nodes can be active for a longer duration of time, as shown in the various discharge rate profiles in Fig. 7. This result offers the basis for battery-efficient discharge and sensing capabilities.

Fig. 8 shows the influence of the log normality distribution on throughput parameters under LS-ODCI. All the joined nodes are represented with node four as a test case orphan that got disconnected. In reality, the nodes transmit data streams from the off-takers, but it was observed that node placement distances affect the network performance, especially for complex deployments. TelosB node 3, placed

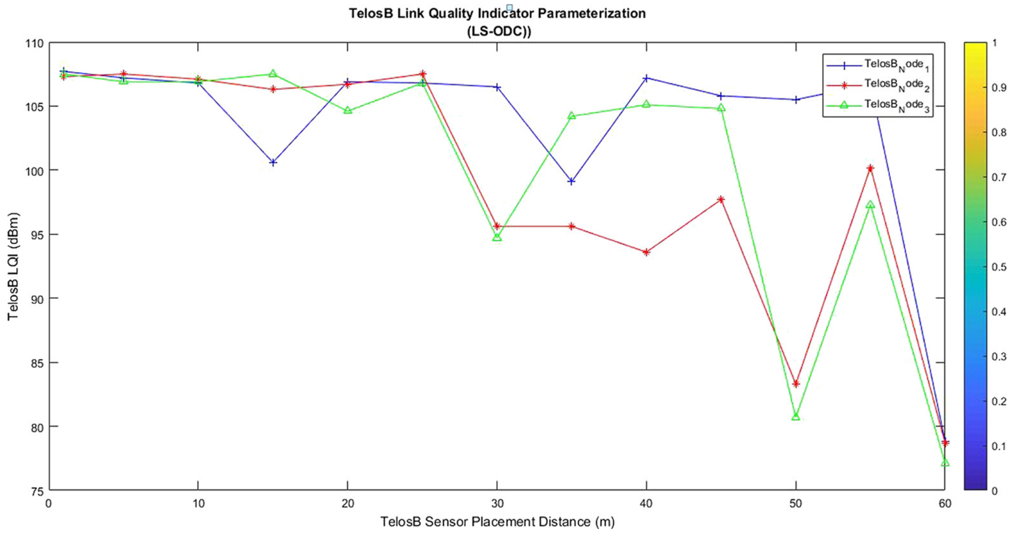


Fig. 5. Impact of LQI on TelosB sensor placement distance (m).

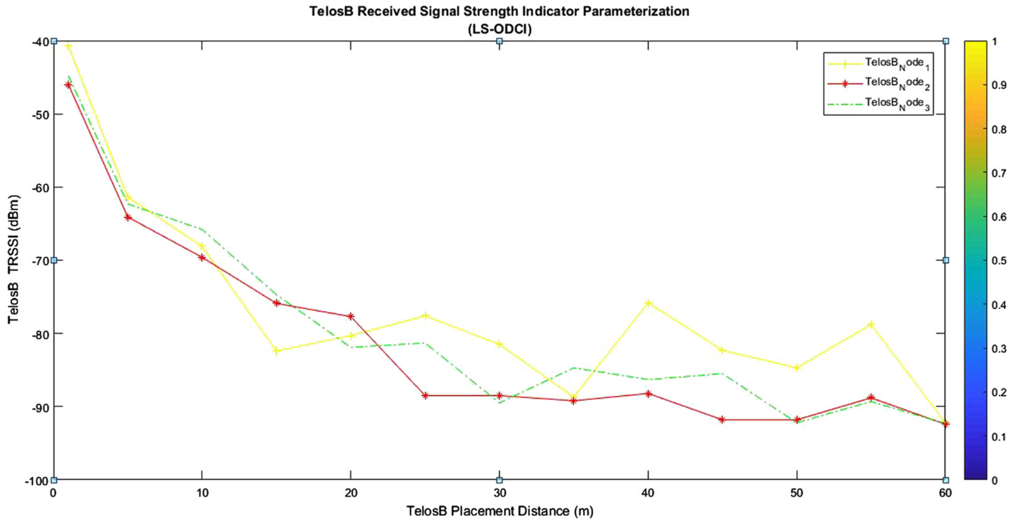


Fig. 6. Impact of RSSI on TelosB sensor placement distance (M).

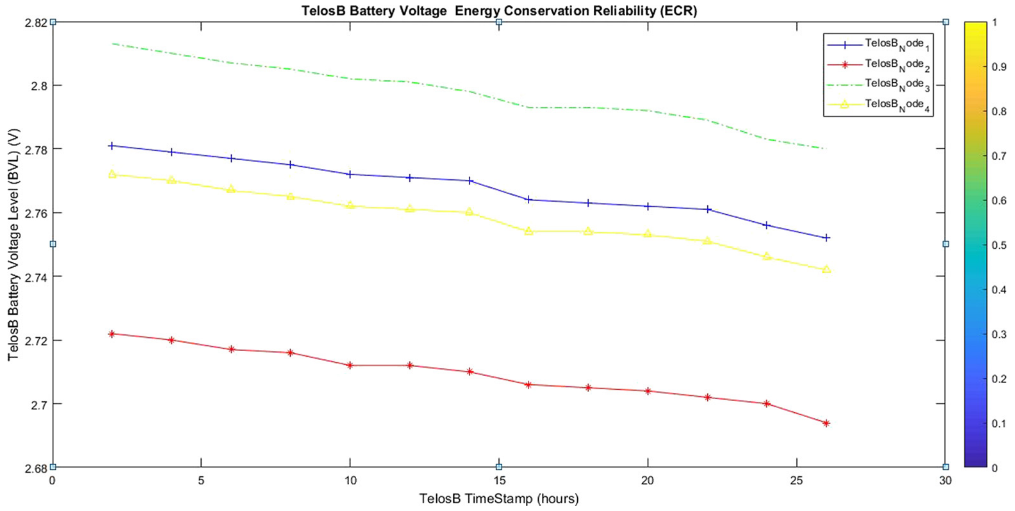


Fig. 7. Impact of battery life under LS-ODCI.

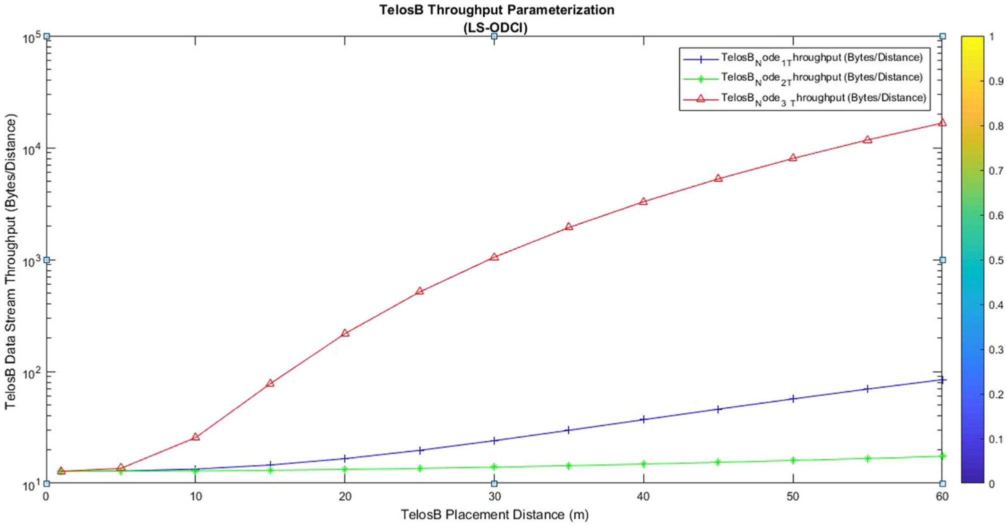
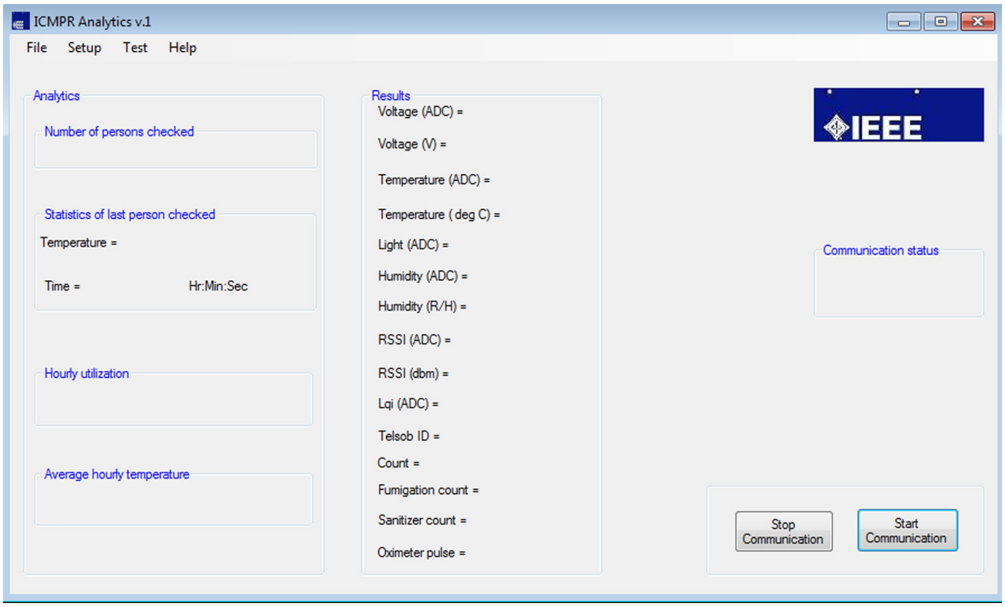


Fig. 8. Impact of the log normality distribution on throughput under LS-ODCI.



Analytics

Number of persons checked

Statistics of last person checked

Temperature =

Time = Hr:Min:Sec

Hourly utilization

Average hourly temperature

Results

Voltage (ADC) =

Voltage (V) =

Temperature (ADC) =

Temperature (deg C) =

Light (ADC) =

Humidity (ADC) =

Humidity (R/H) =

RSSI (ADC) =

RSSI (dbm) =

Lqi (ADC) =

Telsob ID =

Count =

Fumigation count =

Sanitizer count =

Oximeter pulse =

Communication status

Fig. 9. StreamRobot cloud application interface.

closer to the sink, experienced the highest throughput without any deviation. Similarly, nodes 2 and 1 were placed closer to the sink after node 3 – both experienced differential a throughput behaviour with LS-ODCI. Results shown in Fig. 8 indicate that the LS-ODC nodes closer to the sink analytic server experienced better throughput compared to those behind the lead node. This is a normal distribution confirmed from the experiment. Therefore, the introduced ORTM can trigger instant node recommendations for complex network setups [33]. This is good for the StreamRobot optimisation of data transmission. To migrate these parameters to the cloud, a set of fog devices can be employed for streams analytics. The insights gathered can be used for intelligent decision-making in the deployed context [34] and [35].

With such smart task placement in StreamRobot IoT TelosB nodes on a large-scale computing platform, workload optimisations will deliver sensitive data traffic, involving low power drain at the edge nodes. The fog will improve latency and can be applied in container orchestration domains, while machine learning heuristics could be applied to achieve quicker decisions in terms of scheduling. Future efforts will consider reinforcement learning and evolutionary schemes for dynamic IoT TelosB edge to cloud transactions.

In all cases, the throughput increases as the data stream increases from the off-takers to the sink. RSSI and LQI can affect the throughput, especially if the node placement distances are wrongly calibrated or if an orphan node exits. So far, using the proposed scheme, the results show that data stream transmission, considering LQI, RSSI, battery life, or energy conservation reliability, and throughput (under efficient node placements), can sustain optimal traffic flows through any wireless vector path. Therefore, reliable physical calibrations are necessary for the proposed ISCR.

Furthermore, with an increasing number of IoT edge nodes, the total number of the joined nodes with ORTM will remain higher than that of conventional IP TelosB nodes. Using ORTM, offers similar results even in a complex experiment. In this article, the network employed a smaller number of joined IoT TelosB nodes for parameter gathering in a service robot. The behaviour of the individual nodes was studied for a reliability analysis. As the number of nodes increases, the ORTM eliminates possible saturation effects. The implication is that more nodes can join in a subnet cluster using CIDR or MIHT. Also, in an unstructured deployment, ORTM can accommodate a massive node deployment, improve the overall joined ratio, and regulate all active deployment interactions.

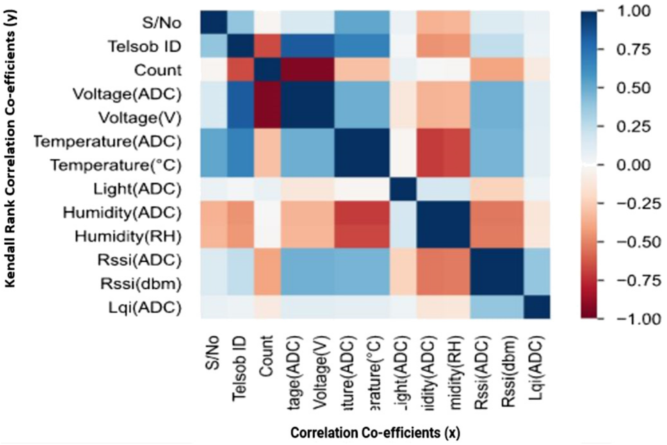


Fig. 10. StreamRobot Kendall's correlations map.

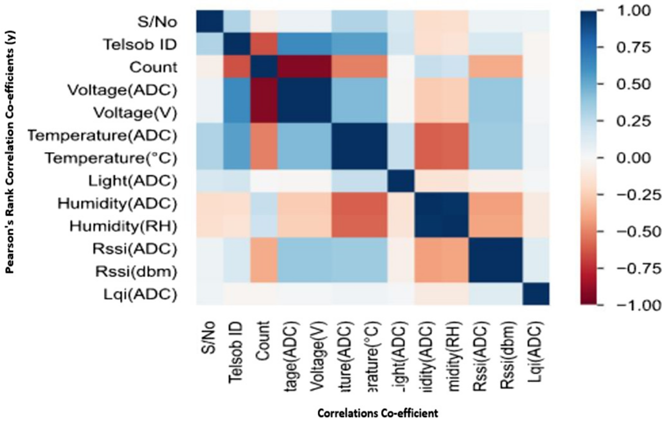


Fig. 11. StreamRobot Pearson's correlations.

It is difficult to notice the disjoined/unconnected state when the nodes are increased with ORTM. Consequently, an increase in the node placement will have less impact on the overall network at any given location, especially the battery lifespan. Fig. 9 shows the StreamRobot cloud interface for all the cyber-physical parametric captures.

Fig. 10 shows Kendall's correlations map for capturing the ordinal association between the measured quantities by the StreamRobot. Fig. 11 shows the Pearson's correlations for the StreamRobot multivariable. It depicts the statistical relation between the continuous variables. Fig. 12 shows the Phi_K correlation coefficient obtained from several adjustments to Pearson's hypothesis test of independence among multi-variables. Fig. 13 shows Spearman's correlations between the variables. The various test statistics depict a common similarity pattern in terms of all the captured variables by the StreamRobot. In Section 5, a discussion of the complex system-of-systems with OpenFlow SDN is presented.

5. OpenFlow SDN integration architecture

In most mission-critical environments, the use of service robots is unpopular, and the legacy computing systems depend on human agent operators who manually monitor and manage the system conditions. The efficiency depends on the overall system-of-systems integration. This ensures proper coordination and offers system stability at deployment sites.

Apart from the edge parameter transmission, critical intelligence is realised using self-monitoring and feedback control schemes in the StreamRobot. Such a system-of-systems model (cf. Fig. 14) can have an average lifespan of 10 to 30 years with SDN deep learning integrations [36, 37]. The issues of orphan node problems in the systems-of-systems architecture leads to the premature death of the StreamRobot. This

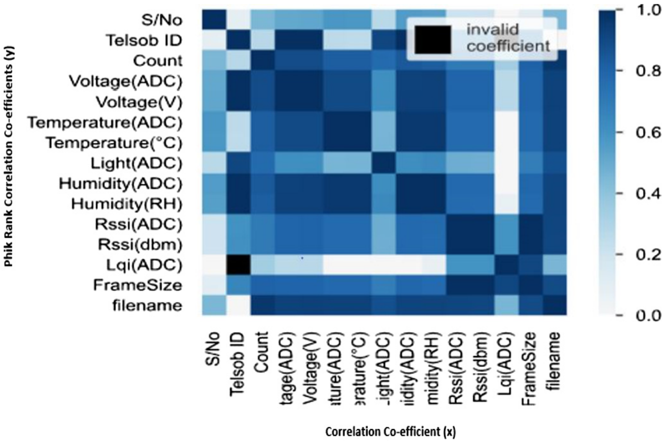


Fig. 12. StreamRobot Phik correlations.

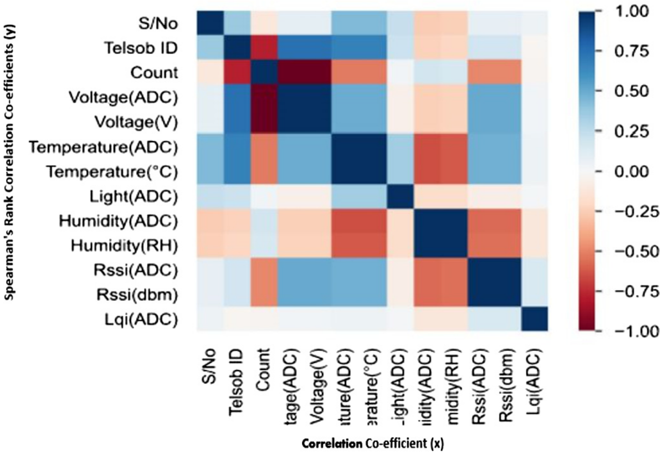


Fig. 13. StreamRobot Spearman's p correlations.

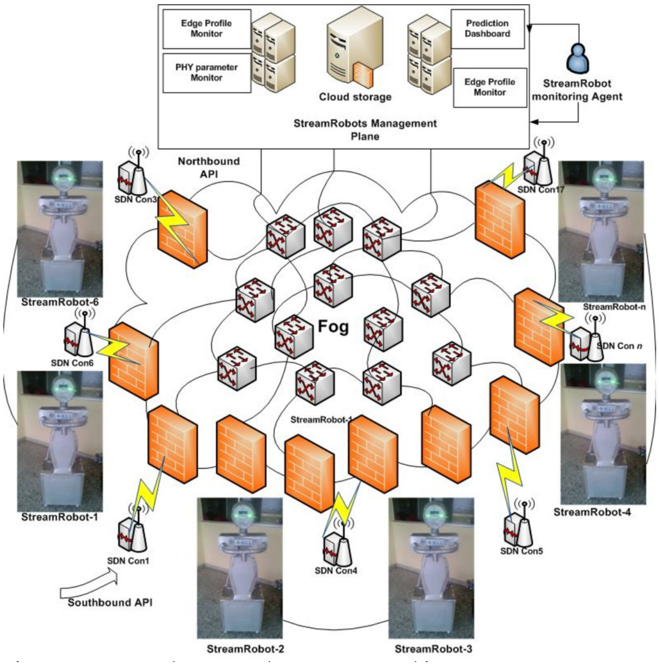
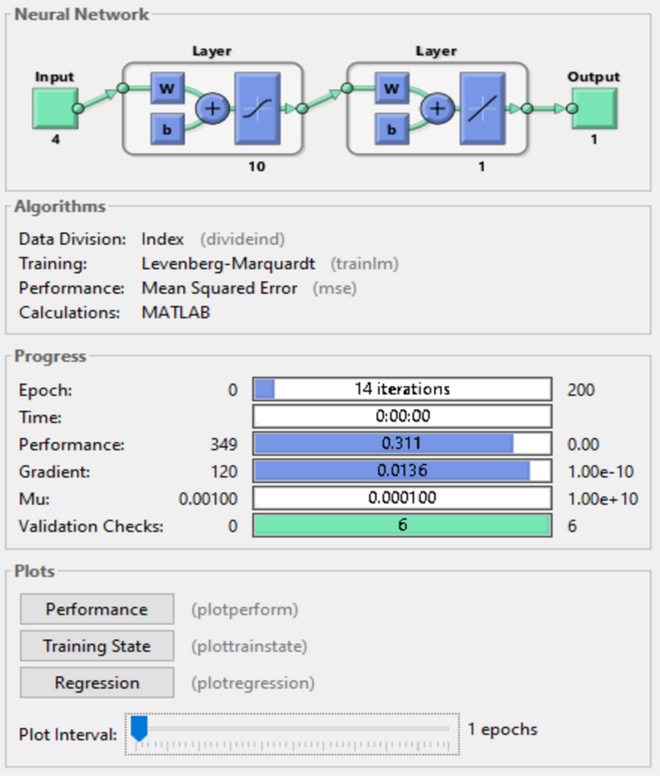


Fig. 14. Proposed StreamRobot SDN-ML architecture.



Algorithms

Data Division: Index (divideind)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

Progress

Epoch:	0	14 iterations	200
Time:		0:00:00	
Performance:	349	0.311	0.00
Gradient:	120	0.0136	1.00e-10
Mu:	0.00100	0.000100	1.00e+10
Validation Checks:	0	6	6

Plot Interval: 1 epochs

Fig. 15. FD-CM predictive model with computation neural controller.

abruptly ends its operational lifecycle. Since existing parameter monitoring systems for data acquisition in Nigeria has no smart way of accounting for the health status of the design components during active operation and fails to schedule routine maintenance based on analytic reports, another reliability optimisation is now introduced in StreamRobots to minimise orphan node failures as a result of non-derived system visibility.

In this section, robust monitoring and prediction systems based on a machine learning (ML) engine are introduced. This is used to achieve real-time monitoring of each StreamRobot deployment. In this case, the software-defined networking machine learning (SDN-ML) scheme is deployed, as shown in Fig. 15. With SDN-ML, smart network programmability is used to speed up management and data manipulation in StreamRobot networks. This was used to separate the data plane from the control plane. The data plane decouples the forwarding hardware for fog agent decision-making (routing and control algorithms). Similar to the work [37], the authors streamlined the plane separation to achieve flexibility, programmability, and cost-economy from the fog to the cloud infrastructure as indicated in Fig. 14. By infusing SDN-ML in the StreamRobot network, configuration policies are now executed on the controller rather than executing from network devices.

From Fig. 14, the SDN-ML controllers offer scaled visibility of the StreamRobot's topology, allowing for a unified configuration of the nodes from the management plane. The benefit of this optimal reliability approach is to offer granular and resilient systems-of-systems management of StreamRobots' large-scale deployment with a lower administrative overhead. As shown in Fig. 14, the forwarding table uses the SDN engine to complete the data transmission forwarding, based on lookup table traffic matching. The fog OpenFlow protocol is used to push transactions between the StreamRobot SDNs and fog OpenFlow multilayer switches. From the hardware controller (discussed in Section 8), the OpenFlow protocols comprise structured logic messages, exchanged among the controllers, and the switch in a secured connectivity channel. At this stage, the SDN controller moves updates/advertised logic messages with keys such as *add*, *remove*, *modify*, etc., from the lookup forwarding tables. Every time an incoming signal gets into the SDN OpenFlow switch, the signal-packet details are checked against the forwarding table for concurrence. Matched validity results in forwarding accent to the designated port. If this is invalid, a smart query request is to be sent to the SDN controller for advice updates, defining the exact location to forward the packetized signal. This is then achieved by allowing the SDN-controller to carry out quick topological database queries with deep learning and confirm new rules or notification states.

The proposed StreamRobot SDN-ML architecture in Fig. 14 fuses its forwarding plane optimally. It depicts the OpenFlow engine meant for the core IoT wireless network. Above this layer is the smart control plane. This gives a clear picture of the data manipulation and route management supports. A highly secure channel interface, the core SDN, and sink-SDN controllers were introduced in the architecture. The reason was to ensure security and optimal reliability. Finally, the management layer provides the compute storage and real-time deterministic sensing domain, housing all fault-error predictions in the system-of-systems network.

6. Deployment system testbed

A production deployment for the proposed StreamRobot SDN-ML comprises a Linux-based virtual server. The main data centre network executes the SDN design, having seven SDN controllers with their respective fail-over supports. The OpenFlow switch is used as the forwarding engine (i.e., OpenFlow multiplayer vswitch (3) [37]. An Ubuntu server was used with a 16-port NIC Intel ethernet [38], mininet [39], and a floodlight-controller [40] being deployed as SDN components. This was upgraded to fit the proposed network demands for the StreamRobot. The matching constructs determine the data stream forwarding verified from the lookup table. The SDN floodlight was used to determine which device should receive forwarded packets.

The TelosB sensor nodes provide a critical wireless tunnel for the StreamRobot. Bluetooth LoRa RF provides sensor capability for a variety of readings. The TelosB IoT wireless network works with the SDN controller. While the SDN sink-node forms the SDN controller, the other SDNs provide an OpenFlow forwarding engine. This article depends on the modified SDN algorithm [37]. This was used to achieve StreamRobot communication seamlessly (cf. Fig. 14). The system setup also has both virtualised and StreamRobot hardware prototypes employed to achieve the deployment. The virtualised domain supplies the baseline network with the smart SDN and OpenFlow switches. TelosB nodes for various sensing attributes are used to measure all the physical parameters as well as to trigger notifications on the application management plane via north- and southbound APIs. First, the physical variables were profiled on the analytic dashboards. These nodes capture analogue variables, mapped via signal conditioners into the Arduino UNO microcontroller. This processes the analogue-to-digital conversion while activating the LoRa module for data transmission. For failure predictions on the StreamRobot and the network conditions, this work introduced a reliability fault predict agent, leveraging past TelosB nodes historical datasets. Section 8 discusses the hardware proof of concept.

7. StreamRobot fog detection cloud predictive machine learning model (FD-CPMLM)

7.1. FD-CPMLM-MSE analysis

In this section, FD-CPMLM was introduced in Fig. 15 to train the accuracy of the StreamRobot's predictions. This is also called enhanced neural discriminant analysis (ENDA), based on SDN intelligence propagation (i.e., OpenFlow-SDN). In this case, the number of hidden neurons chosen is based on the mean square error (MSE) and regression (R). The prediction model design is the neural-based training data classification for a StreamRobot's IoT datasets. Ten neurons were chosen to design

Table 2. Neural classification tuning parameters.

Neural Specification	Specification Values
Network type	Predictive neural controller
Number of samples	6,000
Hidden layer	10
Sampling time	20 sec
Sampling interval	0.2
Plant input/output	2
Training samples	6,000
Training epochs	200
Training function	Trainlm
Control weighing factor (p)	0.00229
Search parameter	0.001
Control horizon (N_2)	2
Cost horizon	2
Reliability coefficient	1

the network prediction model due to the number of data complexity involved. In evaluating a model's performance, error measurements were considered. The MSE refers to the error that is to be minimised to realise an acceptable output from the artificial neural network (ANN). When the MSE between two consecutive epochs is less than the minimum error that is specified, then the training stops. Essentially, the training also stops when the validation resolves that overfitting occurs. The MSE also determines how well the network output fits the desired output. The R-value is considered, as it measures the correlation between outputs and targets. An R-value close to 1 means a close relation, while the reverse indicates a random relation. A comparison of the MSE and R-values for all numbers of nodes was carried out. The lowest MSE value was selected as the optimum number of nodes in the hidden layer, and also the R-value, which is the highest. MATLAB 17 is the multi-paradigm numerical computing environment employed in this regard. It has a network fitting tool referred to as NFTOOL used for solving data fitting problems. It maps a data set of numeric inputs and a set of numeric targets. Table 2 shows the tuning parameters for the predictive training classifications.

Fig. 15 shows a typical NN model with multiple hidden layers. The major consideration in computing the error level in the StreamRobot prediction model, including testing reliability of systems-of-systems prediction states, is the already mentioned mean square error (MSE) and root-mean-square error (RMSE). These represent the objective-cost function in the StreamRobot.

Essentially, the smaller the objective cost function in the system, the more reliable its fault prediction analytics becomes. Eq. (19) and Eq. (20) are used to express the MSE and RMSE in the proposed system. The differentials in both equations allow RMSE to give more computational weights to huge errors. This is certainly reasonable in the instance of unacceptable errors predicted [37].

$$Obj(k_1, k_2, k_3, k_4 \ldots \ldots .k_{n+1}) = \frac{1}{n}\sum_{n=1}^{n}(RL_{Pred} - RL_{targ})^2 \tag{19}$$

$$Obj(k_1, k_2, k_3, k_4 \ldots \ldots .k_{n+1}) = \sqrt{\frac{1}{n}\sum_{n=1}^{n}(RL_{Pred} - RL_{targ})^2} \tag{20}$$

From Eq. (19) and Eq. (20), the system is routinely trained in a repetitive fashion, accommodating new events on an hourly basis.

With the backpropagation model [41], cost-effective retraining is achieved in the SDN-ML/FD-CPMLM. The baseline MSE estimated reference used is $7.6 * 10^{-3}$ [42]. After using the deployed machine learning model (cf. Fig. 15), it was observed that it provides efficient accuracy in all of its predictions with similarity patterns in respect of the measured variables. This monitors the entire system status. In this case, a prediction status profile (PSP) was assumed as a significant tool for determining the operational health state of the StreamRobot. Added to that, the PSP is considered to be significant on the baseline scales of 0 to 1 where < 0 denotes 'no serious status' and 1 denotes a 'high health condition' without orphan nodes. The fuzzy states between 0 and 1 are denoted as the operational conditions of the SDN-powered StreamRobot. By mapping the TelosB sensor data streams into a sigmoid function, the optimal reliability of the StreamRobots can be deduced. Take X_i to denote the PSP of a TelosB sensor node. The PSP for multivariable inputs is now given as a typical logistic regression model in Eq. (21) [37]:

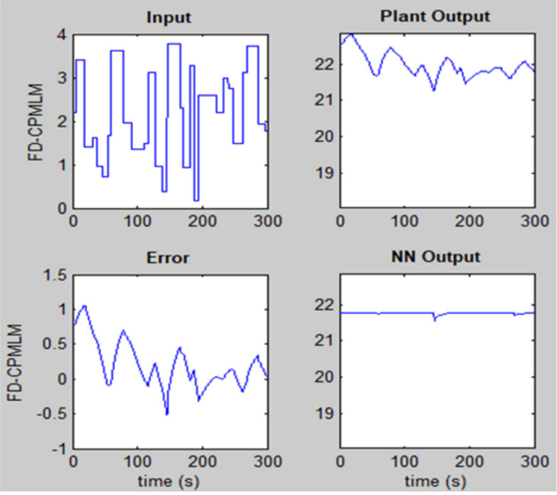


Fig. 16. Machine predictive model with slow rate data streams.

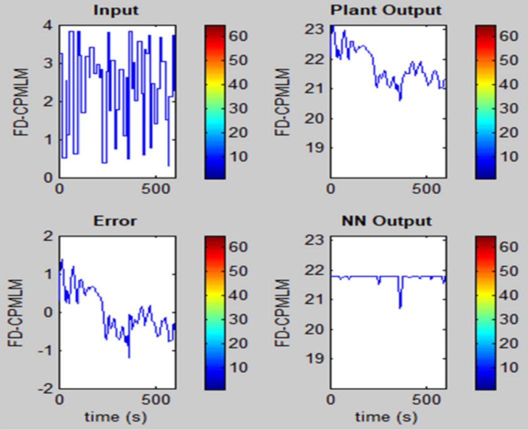


Fig. 17. FD-CPMLM/predictive model (SDN) with more accurate detection.

$$SI = \frac{1}{1 + e^{-\sum_{i=1}^{n}(\beta_{X_i})}} \times 100\% \tag{21}$$

When β denotes the computational weight effect of each IoT node variable, this typically ranges between 1 and 10. Fig. 16 illustrates a predictive machine model with slow rate data streams for a four-input sample over 200 simulation cycles. For the four input samples, the mean square error is between 0 and 0.5. The classified neural output shows no significant moving trend – hence a very small error was detected. It shows that a machine predictive model (OpenFlow-SDN) in which the four input samples have a mean square error, is between −0.5 and 1. The classified neural output shows a significant moving trend – hence some error states were detected. Fig. 17 shows a composite machine predictive SDN with improved detection. In this case, the mean square error is between −1 and 1. Similarly, the classified neural output shows a significant moving trend – hence internal state errors were detected.

Fig. 18 shows the completed StreamRobot schedule training data validation, using a SDN data mining predictive controller. To obtain the training residual of 0.19422, the target (T) must approximately correspond to the predictive output (Y) i.e., $Y \rightarrow T$. Hence, in the validation

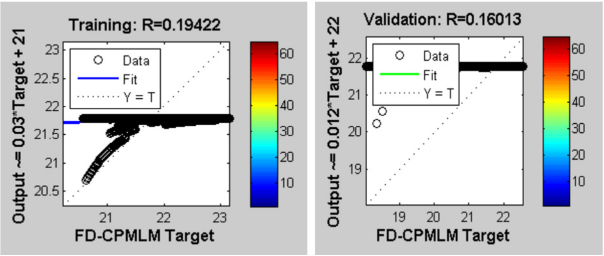


Fig. 18. StreamRobot SDN training data validation with a data mining predictive controller.

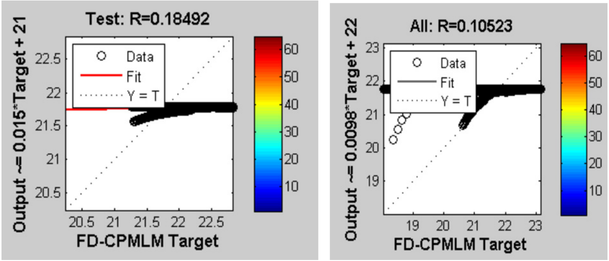


Fig. 19. StreamRobot SDN test data validation using a data mining predictive controller.

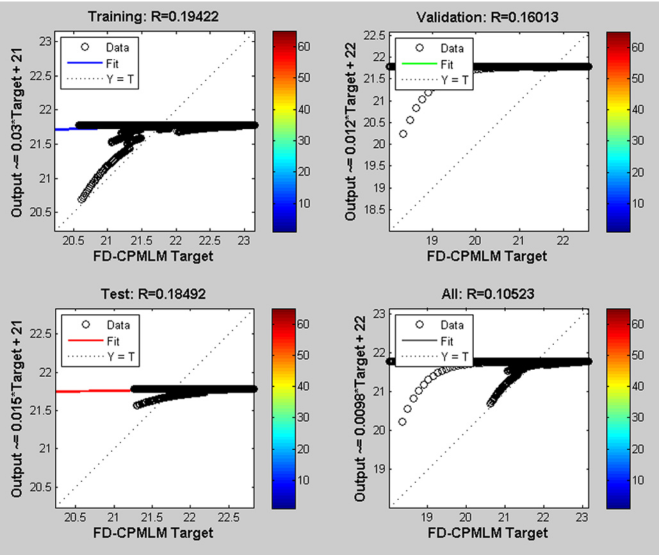


Fig. 20. Complete StreamRobot training data validation (FD-CPMLM).

plot of the training residual at 0.16013, the target completely approximates the predictive output $T(22) \rightarrow Y(22)$. Additionally, in Fig. 18, a complete StreamRobot test data validation using a data mining predictive controller, is demonstrated. For the test residual (R = 0.18492), the target (T = 22.5) linearly approximated the predictive output (Y = 22.5). Similarly, in Fig. 19, for the validated test residual, R = 0.10523. Here, the validated target (T = 23) has linearly approximated the predictive output (Y = 23). Fig. 20 shows the complete streamRobot training data validation (FD-CPMLM) as discussed above. It illustrates the regression plot of the training, which shows the relation existing between the outputs of the network and the targets. Essentially, the four plots represent the training, validation, testing, and general data for the model. The dashed line in each plot represents the targets or perfect result-outputs. The solid line represents the best fit linear regression line between the outputs and targets.

Fig. 20 illustrates the StreamRobot data mining predictive controller (FD-CPMLM) MSE validation. As shown in Fig. 21, the dotted path shows the best path. The best validation performance is experienced in which the dotted horizontal line and the dotted vertical line intersects. This was achieved after six iterations. The performance stopped increasing at this point, and the training was stopped. For this model, the best validation performance was observed at epoch 1 without further increase, so the training was stopped at the eighth epoch. It was observed that out of the 14 epochs for training, test, and best residuals, the best validation performance was obtained at the eighth epoch, at 1.7562 MSR error. The significance of these results is that the proposed system offers a very reliable/accurate region of failure/error classification. From these results, it is clear that the established data mining predictive controller has 1.7562 MSE for the discriminant function's 0.00229 critical value (threshold). This makes it difficult to have orphan nodes.

7.2. FD-CPMLM-MSE validation and testing

In this section, the performance evaluation of selected identified maching learning models is carried out. The idea is to discover the most optimal model that is accurate, consistent, and reliable in a StreamRobot. In this regard, three reliability optimisation schemes were studied for error/failure detection, namely decision tree (DT) [43, 44], logistic regression (LR) [45], and the deployed SDN scheme (i.e., ENDA).

Fig. 22 shows the validation plots for data clustering mean square latencies for the three optimisation algorithms. It was observed that three algorithms, namely decision tree, logistic regression, and SDN-ENDA (proposed) gave a latency hit-clustering of 40.00%, 33.33%, and 26.67% respectively, for the predictive fog model.

Fig. 23 shows the StreamRobot's linear predictive scalability behaviour of the selected algorithms on the network plane. These algorithms vary significantly in predictive scalability, especially as data samples are increased. These scalable algorithms, namely decision tree, logistic regression, and enhanced neural discriminant analysis (proposed) (ND) offer a clustering scalability of 30.12%, 33.73%, and 36.15% respectively.

From the analysis so far, it is obvious that the optimal SDN computational strategy (i.e., SDN-ENDA) offers a better prediction of errors than existing decision trees and regression schemes. However, identifying the proper attributes and mapping exact thresholds may even offer more precise results.

8. Experimental hardware prototype

Fig. 14 depicts the systems architecture powered by an optimal SDN packet flow engine. The integral part of the network is denoted as the fog cloud. This has multiplexed-multipath route connections driven by the SDN controllers, which activate and enforce off-taker forwarding engines to fix recurring query requests. All the location-based SDN controllers are linked via cascaded SDN with OpenFlow switches. This depends on TelosB IoT sensor data streams to the major SDN controller deployed at the cloud edge. The StreamRobot experimental cloud is the Linux-based virtualised hypervisor machine. In practical deployment, mininet and floodlight controllers are the SDN controller implementation agents. At the open vSwitch engine, the rule-base was deployed fully. The proof of concept for the systems-of-systems is presented at the fog control plane. Automatic encasement deployment for the StreamRobot gate is discussed below, as depicted in Figs. 24 to 26. This addresses the disaster issues in the COVID-19 incident response concerns in Nigeria and other African countries. The StreamRobot has the following specifications as depicted in Table 3.

In the deployment context, a facial recognition system powered by a deep learning neural convolutional script agent, coordinates the access control to the StreamRobot gate (Fig. 25). This is integrated with the temperature screening function. This quickly takes a skin-surface temperature and uploads abnormal temperature events to the network operating centre orchestrated in the private cloud network operating

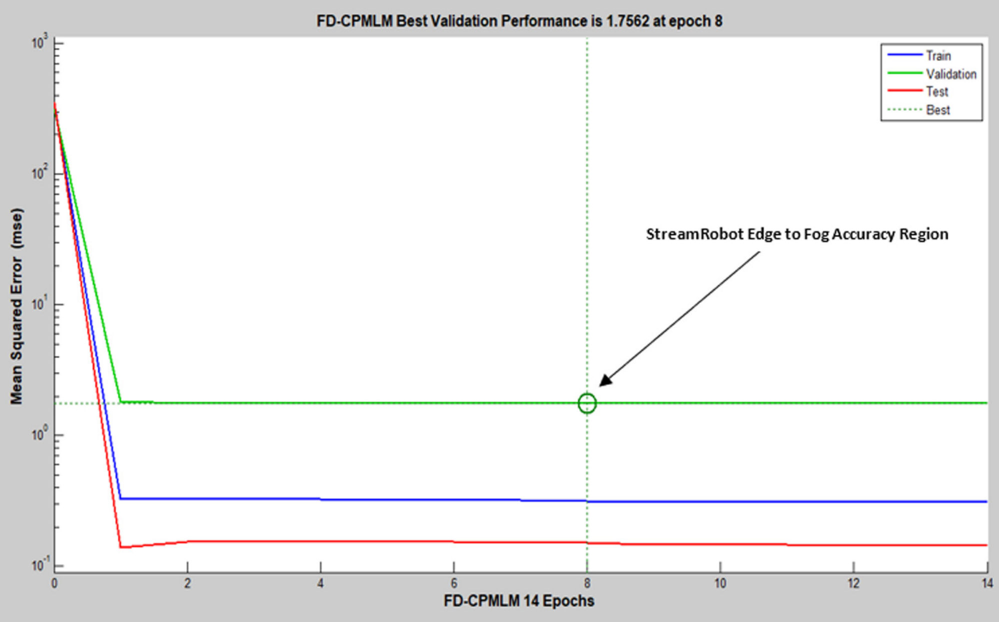


Fig. 21. Data mining predictive controller (FD-CPMLM) MSE validation.

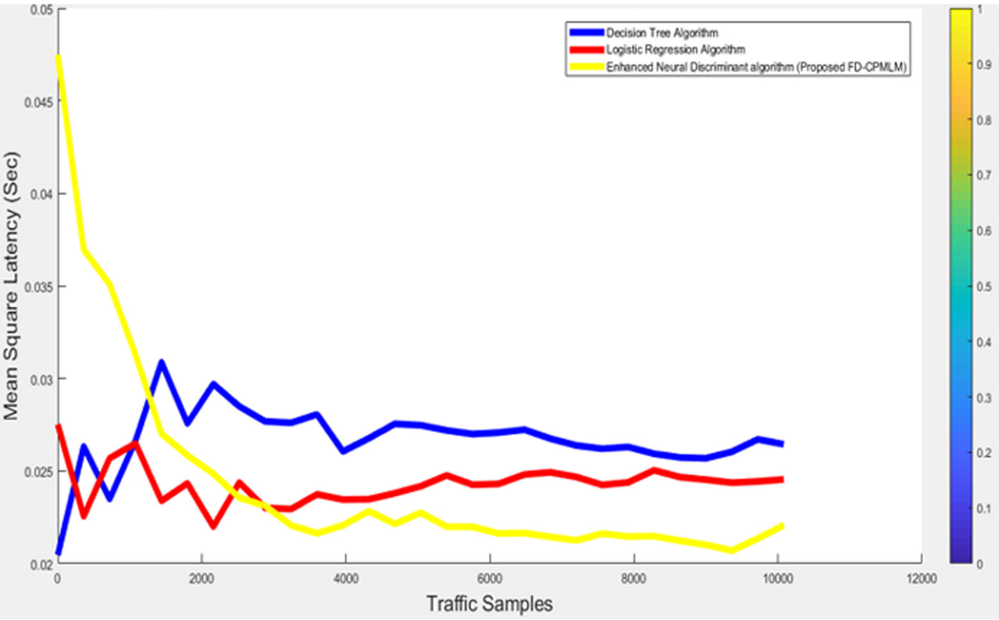


Fig. 22. Validation plots for data clustering latency.

Table 3. Features of the StreamRobot prototype.

StreamRobot gate features	Parametric values
Disinfection scheme	Aerosol-vent sprayer methods
Disinfectant molecular size when applied	0,1-0, 3-0,5 microns
Surface retention time after application	6 hours
Internal tank on full state	40 litres
Number of the points that apply disinfectant to the surface nozzle	12 pcs
Nozzle type	Aluminium
Nozzle angle	80 degrees/fan
Automation	Digital control
Tank internal capacity	75 litres
Pressure pump	35 bar
Main flow pump	1 litre/minute
Voltage	220-230v/60 Hhz
Device dimension (cm)	E: 130x180x220 cm
Weight	120 kg
Sensors (distance, tank capacity, temperature, oxygen level, etc.)	Default/Enabled

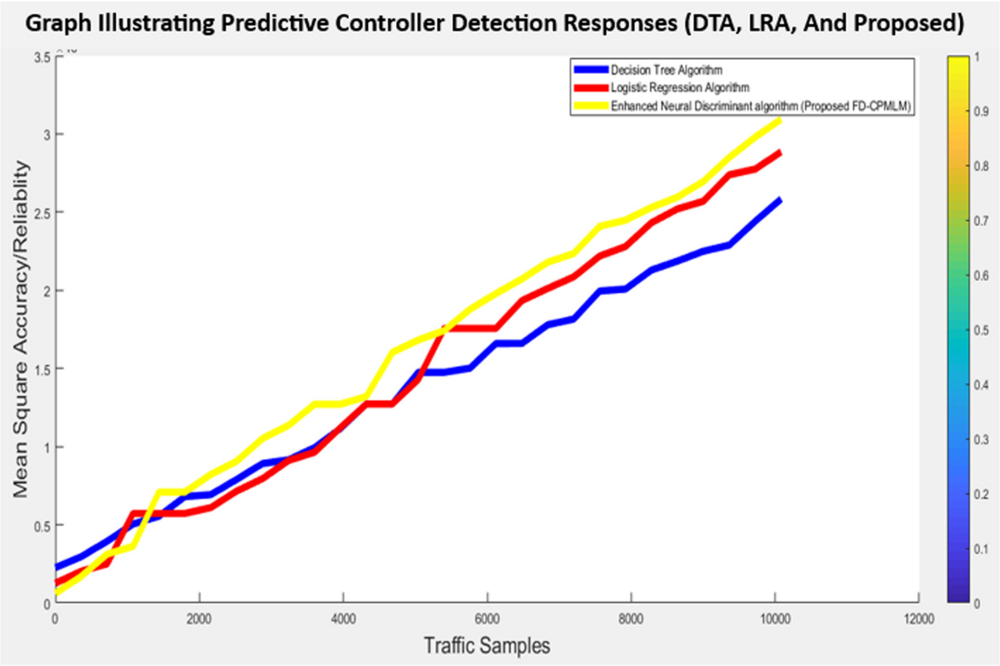


Fig. 23. Validation plots for data clustering latency.

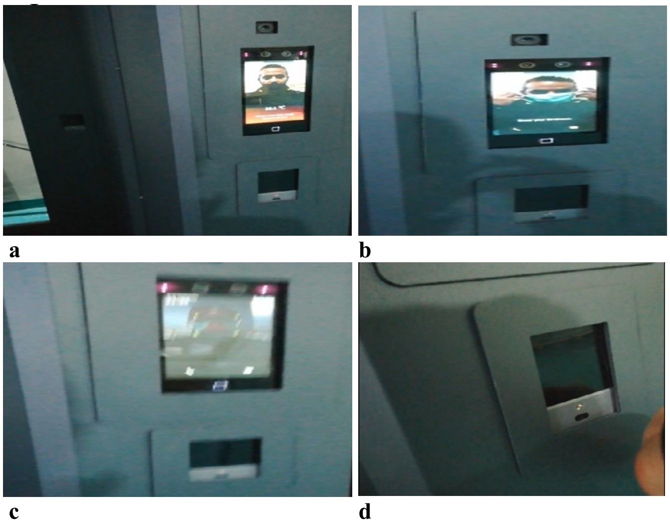


Fig. 24. a) StreamRobot gate face camera mode without face mask; b) Face-mask detection mode; c) StreamRobot gate deep learning scan for final verification to fog-cloud agents; and d) StreamRobot temperature mode.

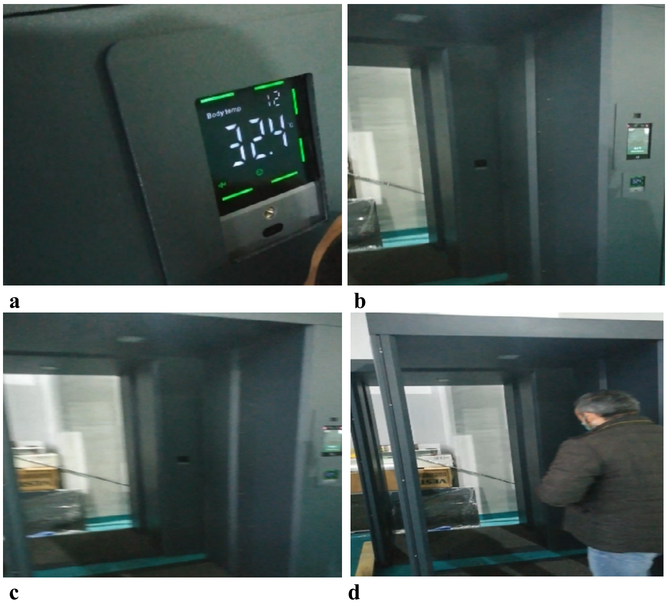


Fig. 25. a) StreamRobot gate face camera mode without face mask; b) Face-mask detection mode; c) StreamRobot gate deep learning scan for final verification to fog-cloud agents; and d) StreamRobot temperature mode.

centre (NOC). The StreamRobot deployment provides support for the Vanadium Oxide uncooled sensor that measures the target temperature range: 30 °C to 45 °C (86 °C-113 °C); accuracy: 0.1 °C; deviation: ±0.53 °C; recognition distance: 0.3 to 1.8 metre. The camera resolution is 120*160, with a framerate of 25fps and 2MP dual-lens. At deployment, the fast temperature measurement mode detects faces and takes temperature readings without identity authentication (Fig. 24).

However, multiple authentication modes are supported, such as card and temperature, face and temperature, card and face, and temperature. Another feature introduced is the face mask-wearing alert, foglet agent. In this case, the moment the face-recognition face module fails to identify the face mask, the device IoT voice reminder is activated. Fig. 26 shows the entry, standstill, and verification modes upon receiving control commands from the StreamRobot front-end depicted in Figs. 1 and 14.

Similarly, the authentication and attendance are flagged as valid, once the facial recognition engine verifies the captures. Forced mask alerting wearing alert can occur. In this case, if the recognising face does not wear a mask, the device prompts the IoT voice reminder. At the same time, the authentication or attendance will fail. Furthermore, the display temperature measurements are captured on the authentical page (cf. Fig. 25a). For accurate temperature measurement, the system is powered on for 90 minutes for warm-power-up. A voice prompt is triggered to indicate an abnormal temperature reading. The system works in both indoor and outdoor environments.

As shown in Fig. 24, the face recognition duration is less than 0.2 second per user. The face accuracy rate is ≥ 99%, with over 6,000 to 50,000 face capacity. The card capacity is 6,000 to 50,000 and can

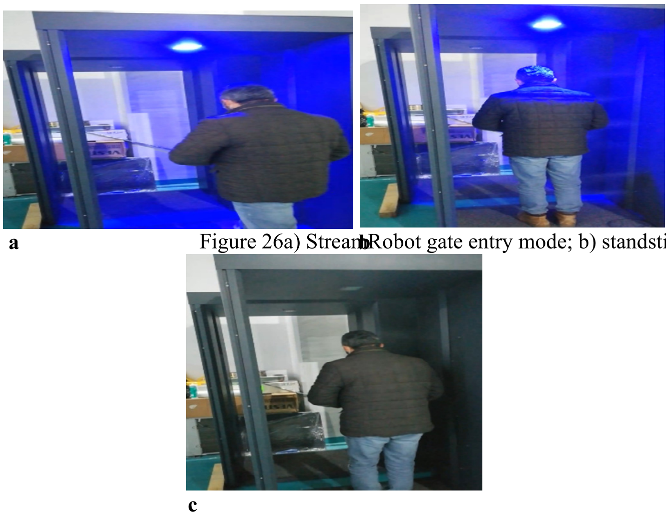


Fig. 26. a) StreamRobot gate entry mode; b) standstill mode; and c) final verification to fog-cloud agents.

handle an event capacity of 100,000 to 200,000 event capacity. The facial height for image recognition is between 1.4 and 1.9 metres. By default, the system supports over 6 attendance status, including check-in, check-out, break-in, break-out, overtime-in, overtime-out. At a high field programmable controller auto-synchronisation clock, the watchdog design and tamper functions are enabled. The audio prompt for authentication results is enabled for errors (cf. Fig. 3). The system connects to an external access control unit via a bluetooth radio and fog radio frequency of 2.4 GHz at 1 Mbps data rate. From the deployment, datasets are imported or exported from the StreamRobot via the foglets into the private cloud NOC. The StreamRobot's gate chamber uses its configurable status (open/close) when detecting abnormal temperature or orphan node problems. The front end equally displays the information via the TCP/IP communication APIs and stores it on its internal memory for next reference.

In Fig. 27, the StreamRobot SDN IoT bode prototype is depicted. This is constructed with a customised IoT core, programmed with SDN constellation scripts. The prototype hardware module is based on an Arduino board working as a programmable microcontroller device with SDN supports. The module has various sensors such as temperature, humidity, proximity distance sensor, humidity, and tank capacity. The TelosB sensor device-nodes are mapped with an OpenFlow orchestrator, which interacts with the terminal SDN controller, leveraging the long-range network connectivity LoRa network. This fog gateway SDN supports various heterogenous transactions and handles all sink-SDN controllers that gather sensor datasets meant for the forwarding aggregation to the cloud sink NoC. Immediately the data stream is processed and computed, and these are polled into the prediction dashboard for prediction analytics on cyber-physical real-timed sensed datasets. With reliable predictions, failure modes or errors can be identified in the deployed StreamRobot whose encasement gates are shown in Figs. 25 and 26 respectively.

This work has shown that with the reliability optimisation technique in the StreamRobot, it is feasible to predict and even analyse critical retention profiles from the global trained datasets. The SDN intelligence fully achieves this capability and eliminates the orphan node problem. The robot has multiple arrays of sensors and can be deployed in various scenarios, like enterprises, stations, residential buildings, factories, schools, hospitals, campuses, etc.

The developed intelligent service-care robots will benefit mission-critical sectors such as healthcare systems [46], transportation, manufacturing, and environmental applications. For instance, the system can be deployed as a smart surveillance model for predictive-early detection/notification of community-induced epidemics (SSMPEDCIE).

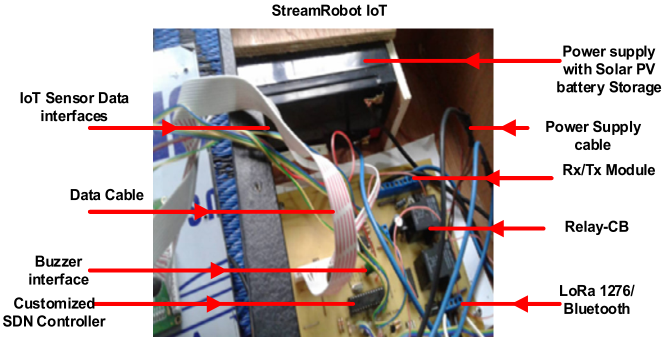


Fig. 27. OpenFlow SDN IoT substrate prototype.

The practical significance is that the system offers a predictive early warning, especially for strange/unusual health status conditions in data-aggregated captures. Such clinical (structured) and nonclinical (unstructured) data information can give insights to perceived syndrome or unconventional health updates. Other crisis areas include hospital sanitising, drug deliveries, and frontline healthcare workers' support to mitigate a COVID-19 exposure.

In summary, recent studies have justified the significance of ISCR, especially in collaborative sensing robots (cobots) [47, 48, 49, 50]. Other new areas include IoT-telemedicine robots [51], service robots [52], companion cloud-based robots [53], telepresence robots [54], social care geriatric robots [55], human-care robots [56, 57], telerobotics [58], and social assistive robots [59, 60]. These works offer greater potential for the internet of medical robots (IoMR). The adaptation of deep learning and unsupervised machine learning can create a huge computational workload for the edge controllers. Therefore, lightweight computation using dockers/containers at the edge appears preferable for ISCRs using LS-ODCI API and SDN.

In this article, LS-ODCI API offers a lightweight approach to processing ISCR wireless parameters, and draws less system resources in terms of computational complexity (space and time). On the other hand, the SDN offers a robust technique for a robotic network management and logically supports dynamic programmability and configurability, thereby improving system performance. It supplies a better approach towards monitoring networks, unlike other schemes that run on static architecture of traditional robotic network designs. When deployed on a robot to centrally manage all the traffic processes, this disassociates the wireless network forwarding process (data plane) from the routing process (control plane). The control plane is the brainbox of the ISCR-SDN. The only issue is that of scalability and elasticity within the OpenFlow remote communication network.

9. Conclusion

In this article, the physical design and experimentation of a StreamRobot have been carried out using reliability optimisation techniques. Unlike exiting service care robots, the proposed StreamRobot uses LS-ODCI and SDN to address reliable data transmission gaps in the literature. It achieved link resilience and log normality distribution with ORTM. LS-ODCI with ORTM handled performance issues relating to IP enhancement probability and standard deviation effects of edge nodes in the TelosB network field as a link connectivity scheme. Additionally, an investigation into the LND feasibility of a StreamRobot to enhance the connectivity concerns was discussed. Experiments were conducted using a real-world testbed. Added to that, an analysis into the use-case scenario (accurate data streams are seamlessly delivered from the IoT nodes to the sink servers) was carried out. The work analysed sensor nodes' deployment parameters such as LQI, RSSI, throughput, and energy consumption lifespan for sustained data stream transactions. An investigation into the connectivity scheme in respect of an efficient battery/energy consumption is considered. A satisfactory LQI, RSSI, and transmission throughput for a limited number of TelosB nodes in the StreamRobot were also observed. The work showed that a few selected metrics presented a normality distribution for operational efficiency with the IoT node placement distances. A smart SDN IoT engine was introduced to monitor errors and orphan node issues in the deployed environment. The customised controller programmability and its functional operations for the baseline and sink deployments were discussed. Using an intelligent SDN scheme, the IoT nodes were efficiently managed in the StreamRobot. An implementation of the core testbed on a virtualised Linux-based NOC, having multiplexed path redundancy, was also realised. Using the MATLAB neural prediction tool, the prediction accuracy was achieved beside the Python scripting on the OpenFlow virtual switch. Further validation results provided a prediction for data clustering mean square latencies. Considering the three optimisation algorithms, it was observed that decision tree, logistic regression, and SDN-ENDA (proposed) gave data streams latency as 40.00%, 33.33%, and 26.67% respectively for the fog predictive model. For the StreamRobot's linear predictive scalability behaviour of the selected algorithms on the network plane, these algorithms vary significantly, especially as data samples are increased. In this case, decision tree, logistic regression and SDN-ENDA (proposed) offer streams clustering scalability as 30.12%, 33.73%, and 36.15% respectively. Various experimental tests were carried out to validate the hardware prototype, constructed with IoT-SDN sensors and controller modules. The observation is that the proposed reliability optimisation scheme eliminates the orphan node problem during data stream transmissions. Finally, the proposed reliability optimisation schemes offer low-cost overheads and eliminate orphan node conditions while providing a satisfactory real-time management of the StreamRobot processes. Future work will fully investigate node-level optimisations, data stream error controls, and security constraints. Lastly, the impact of ORTM will be explored with other computational algorithms to improve connectivity in a very complex deployment.

Declarations

Author contribution statement

Dr. Kennedy Chinedu Okafor: Conceived and designed the experiments; Performed the experiments; Wrote the paper.

Dr. Omowunmi Mary Longe: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Funding statement

This work was supported by IEEE Humanitarian Activities Committee (HAC) and IEEE Special Interest Group on Humanitarian Technology (SIGHT) (#20-COV-16).

Data availability statement

Data will be made available on request.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

Acknowledgements

This paper and the research behind it would not have been possible without the exceptional support of University of Johannesburg, South Africa, IEEE Nigeria Section, and IEEE Consultants Network AG, Nigeria.

References

[1] C.J. Turner, J. Oyekan, L. Stergioulas, D. Griffin, Utilizing industry 4.0 on the construction site: challenges and opportunities, IEEE Trans. Ind. Inform. 17 (2) (Feb. 2021) 746–756.

[2] A. Mourad, A. Srour, H. Harmanani, C. Jenainati, M. Arafeh, Critical impact of social networks infodemic on defeating coronavirus COVID-19 pandemic: twitter-based study and research directions, IEEE Trans. Netw. Serv. Manag. 17 (4) (Dec. 2020) 2145–2155.

[3] X. Xu, Z. Wang, Z. Tu, D. Chu, Y. Ye, E-SBOT: a soft service robot for user-centric smart service delivery, in: IEEE World Congress on Services, Milan, Italy, 2019, pp. 354–355.

[4] T.J. Wiltshire, S.M. Fiore, Social cognitive and affective neuroscience in human-machine systems: a roadmap for improving training, human-robot interaction, and team performance, IEEE Trans. Human-Mach. Syst. 44 (6) (2014) 779–787.

[5] Z. Li, W. Yuan, Y. Chen, F. Ke, X. Chu, C.L.P. Chen, Neural-dynamic optimization-based model predictive control for tracking and formation of nonholonomic multi-robot systems, IEEE Trans. Neural Netw. Learn. Syst. 29 (12) (2018) 6113–6122.

[6] J. Mišeikis, P. Caroni, P. Duchamp, A. Gasser, R. Marko, N. Mišeikienė, F. Zwilling, C. de Castelbajac, L. Eicher, M. Früh, H. Früh, Lio – a personal robot assistant for human-robot interaction and care applications, IEEE Robot. Autom. Lett. 5 (4) (2020) 5339–5346.

[7] E. Schiller, E. Esati, S.R. Niya, B. Stiller, Blockchain on MSP430 with IEEE 802.15.4, in: IEEE 45th Conference on Local Computer Networks (LCN), 2020, pp. 345–348.

[8] N. Li, K. Liu, Z. Chen, W. Jiao, Environmental-perception modeling and reference architecture for cyber-physical systems, IEEE Access 8 (2020) 200322–200337.

[9] K. Mahmood, M.A. Khan, M. ul Hassan, A.M. Shah, S. Ali, M.K. Saeed, Intelligent on-demand connectivity restoration for wireless sensor networks, Wirel. Commun. Mob. Comput. (2018) 1–10.

[10] Y. Feng, L. Liu, J. Shu, A link quality prediction method for wireless sensor networks based on XGBoost, IEEE Access 7 (2019) 155229–155241.

[11] J. Flathagen, F.T. Johnsen, Integrating wireless sensor networks in the NATO network enabled capability using web services, in: MILCOM 2011 Military Communication Conference, Baltimore, MD, 2011, pp. 828–833.

[12] H. Smeets, T. Meurer, C. Shih, P.J. Marrón, Demonstration abstract: a lightweight, portable device with integrated USB-host support for reprogramming wireless sensor nodes, in: IPSN-14 Proceedings of the 13th International Symposium on Info Processing in Sensor Networks, Berlin, 2014, pp. 333–334.

[13] D. He, C. Chen, S. Chan, J. Bu, A.V. Vasilakos, A distributed trust evaluation model and its application scenarios for medical sensor networks, IEEE Trans. Inf. Technol. Biomed. 16 (6) (Nov. 2012) 1164–1175.

[14] R.A. Rashid, M.R.B. Resat, M.A. Sarijari, N.H. Mahalin, M.S. Abdullah, A.H.F.A. Hamid, Performance investigations of frequency-agile enabled TelosB testbed in Home area network, in: Proceedings of the IEEE 2nd International Symposium on Telecommunication, Technologies (ISTT), Langkawi, 2014, pp. 335–340.

[15] Y. Wang, R. Tan, G. Xing, J. Wang, X. Tan, Profiling aquatic diffusion process using robotic sensor networks, IEEE Trans. Mob. Comput. 13 (4) (2014) 880–893.

[16] S.M. Abd-Elazim, E.S. Ali, A hybrid particle swarm optimization and bacterial foraging for power system stability enhancement, Complexity 21 (2) (2015) 245–255.

[17] E.S. Ali, Optimization of power system stabilizers using BAT search algorithm, Int. J. Electr. Power Energy Syst. 61 (2014) 683–690.

[18] S.M. Abd Elazim, E.S. Ali, Optimal power system stabilizers design via cuckoo search algorithm, Int. J. Electr. Power Energy Syst. 75 (2016) 99–107.

[19] N. Rathod, R. Sundaresan, Relay placement algorithms for IoT connectivity and coverage in an outdoor heterogeneous propagation environment, IEEE Access 10 (2022) 13270–13289.

[20] Y. Lyu, Y. Mo, S. Yue, W. Liu, Improved beetle antennae algorithm based on localization for jamming attack in wireless sensor networks, IEEE Access 10 (2022) 13071–13088.

[21] L. Zhou, N. Lannan, G. Fan, Joint optimization of kinematics and anthropometrics for human motion denoising, IEEE Sens. J. 22 (5) (2022) 4386–4399.

[22] Y. Han, H. Hu, Y. Guo, Energy-aware and trust-based secure routing protocol for wireless sensor networks using adaptive genetic algorithm, IEEE Access 10 (2022) 11538–11550.

[23] M. Fujiyoshi, A. Kawamoto, S. Hashimoto, Y. Omura, H. Funabashi, Y. Ohira, Y. Hata, T. Ozaki, T. Akashi, Y. Nonomura, T. Nakayama, H. Yamada, Stress isolation suspension for silicon-on-insulator 3-axis accelerometer designed by topology optimization method, IEEE Sens. J. 22 (5) (March 2022) 3965–3973.

[24] D. Özbek, T.B. Yılmaz, M.A.İ. Kalin, K. Şentürk, O. Özcan, Detecting scalable obstacles using soft sensors in the body of a compliant quadruped, IEEE Robot. Autom. Lett. 7 (2) (April 2022) 1745–1751.

[25] C. Lin, G. Sun, L. Tan, B. Gong, D. Wu, Mobile LiDAR deployment optimization: towards application for pavement marking stained and worn detection, IEEE Sens. J. 22 (4) (2022) 3270–3280.

[26] C. Uysal, A. Onat, T. Filik, Non-contact respiratory rate estimation in real-time with modified joint unscented Kalman filter, IEEE Access 8 (2020) 99445–99457.

[27] J. Milošević, Implementation of LT codes on TelosB platform, in: 2011 19th Telecommunications forum (TELFOR), Proc. of Papers, 2011, pp. 1632–1635.

[28] E. Schiller, E. Esati, S.R. Niya, B. Stiller, Blockchain on MSP430 with IEEE 802.15.4, in: IEEE 45th Conference on Local Computer Networks (LCN), 2020, pp. 345–348.

[29] A. Makhoul, H. Harb, Data reduction in sensor networks: performance evaluation in a real environment, IEEE Embed. Syst. Lett. 9 (4) (Dec. 2017) 101–104.

[30] M. Rausand, A. Hsyland, System Reliability Theory Models, Statistical Methods, and Applications, 2nd edition, John Wiley & Sons, Hoboken, 2004.

[31] K.S. Senevirathna, L. Samaranayake, IPv6 multi-homing with structured CIDR, in: 6th International Conference on Industrial and Information Systems, 2011, pp. 453–456.

[32] C. Lin, C. Hsu, S. Hsieh, A multi-index hybrid trie for lookup and updates, IEEE Trans. Parallel Distrib. Syst. 25 (10) (2014) 2486–2498.

[33] M.A. Imtiaz, D. Starobinski, A. Trachtenberg, Investigating orphan transactions in the bitcoin network, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1718–1731.

[34] K.C. Okafor, G.C. Ononiwu, S. Goundar, V.C. Chijindu, C.C. Udeze, Towards complex dynamic fog network orchestration using embedded neural switch, Int. J. Comput. Appl. 43 (2) (2021) 91–108.

[35] A. Hussain, K. Zafar, A.R. Baig, Fog-centric IoT based framework for healthcare monitoring, management and early warning system, IEEE Access 9 (2021) 74168–74179.

[36] T. Wu, P. Zhou, B. Wang, A. Li, X. Tang, Z. Xu, K. Chen, X. Ding, Joint traffic control and multi-channel reassignment for core backbone network in SDN-IoT: a multi-agent deep reinforcement learning approach, IEEE Trans. Netw. Sci. Eng. 8 (1) (2021) 231–245.

[37] A.K. Al Mhdawi, H.S. Al-Raweshidy, A smart optimization of fault diagnosis in electrical grid using distributed software-defined IoT system, IEEE Syst. J. 14 (2) (June 2020) 2780–2790.

[38] Ubuntu 14.04.5 LTS (Trusty Tahr), 2018 [Online]. Available: http://releases.ubuntu.com/14.04/. (Accessed 8 October 2014).

[39] Mininet, 2018 [Online]. Available: http://mininet.org/. (Accessed 18 July 2021).

[40] Floodlight Controller, 2018 [Online]. Available: http://www.projectfloodlight.org/floodlight/. (Accessed 8 July 2021).

[41] J. Parab, M. Sequeira, M. Lanjewar, C. Pinto, G. Naik, Backpropagation neural network-based machine learning model for prediction of blood urea and glucose in CKD patients, IEEE J. Transl. Eng. Health Med. 9 (2021) 4900608.

[42] M.C. Mabel, E. Fernandez, Estimation of energy yield from wind farms using artificial neural networks, IEEE Trans. Energy Convers. 24 (2) (2009) 459–464.

[43] J. Wang, Y. Qian, F. Li, J. Liang, W. Ding, Fusing fuzzy monotonic decision trees, IEEE Trans. Fuzzy Syst. 28 (5) (May 2020) 887–900.

[44] L.C.M.M. Fontoura, H.W. de Castro Lins, A.S. Bertuleza, A.G. D'assunção, A.G. Neto, Synthesis of multiband frequency selective surfaces using machine learning with the decision tree algorithm, IEEE Access 9 (2021) 85785–85794.

[45] L. Wang, T. Wang, X. Hu, Logistic regression region weighting for weakly supervised object localization, IEEE Access 7 (2019) 118411–118421.

[46] R. Saher, M. Anjum, Role of technology in COVID-19 pandemic, in: Researches and Applications of Artificial Intelligence to Mitigate Pandemics, 2021, pp. 109–138.

[47] G. Pang, G. Yang, Z. Pang, Review of robot skin: a potential enabler for safe collaboration, immersive teleoperation, and affective interaction of future collaborative robots, IEEE Trans. Med. Robot. Bionics 3 (3) (2021) 681–700.

[48] Z. Ye, G. Pang, K. Xu, Z. Hou, H. Lv, Y. Shen, G. Yang, Soft robot skin with conformal adaptability for on-body tactile perception of collaborative robots, IEEE Robot. Autom. Lett. (2022) 1–8.

[49] S. Bharti, A. McGibney, CoRoL: a reliable framework for computation offloading in collaborative robots, IEEE Int. Things J. (2022) 1–13.

[50] H. Yu, Z. Zhou, Optimization of IoT-based artificial intelligence assisted telemedicine health analysis system, IEEE Access 9 (2021) 85034–85048.

[51] I. El Makrini, G. Mathijssen, S. Verhaegen, T. Verstraten, B. Vanderborght, A virtual element-based postural optimization method for improved ergonomics during human-robot collaboration, IEEE Trans. Autom. Sci. Eng. (2022) 1–12.

[52] M. Tröbinger, C. Jähne, Z. Qu, J. Elsner, A. Reindl, S. Getz, T. Goll, B. Loinger, T. Loibl, C. Kugler, C. Calafell, M. Sabaghian, T. Ende, D. Wahrmann, S. Parusel, S. Haddadin, S. Haddadin, Introducing GARMI – a service robotics platform to support the elderly at home: design philosophy, system overview and first results, IEEE Robot. Autom. Lett. 6 (3) (2021) 5857–5864.

[53] Z. Su, F. Liang, H.M. Do, A. Bishop, B. Carlson, W. Sheng, Conversation-based medication management system for older adults using a companion robot and cloud, IEEE Robot. Autom. Lett. 6 (2) (2021) 2698–2705.

[54] M. Wang, C. Pan, P.K. Ray, Technology entrepreneurship in developing countries: role of telepresence robots in healthcare, IEEE Eng. Manag. Rev. 49 (1) (2021) 20–26.

[55] H. Do Manh, W. Sheng, E.E. Harrington, A.J. Bishop, Clinical screening interview using a social robot for geriatric care, IEEE Trans. Autom. Sci. Eng. 18 (July 2021) 1229–1242.

[56] C.R.d. Cos, D.V. Dimarogonas, Adaptive cooperative control for human-robot load manipulation, IEEE Robot. Autom. Lett. 7 (2) (April 2022) 5623–5630.

[57] W. He, C. Xue, X. Yu, Z. Li, C. Yang, Admittance-based controller design for physical human – robot interaction in the constrained task space, IEEE Trans. Autom. Sci. Eng. 17 (4) (2020) 1937–1949.

[58] H. Zhou, G. Yang, H. Lv, X. Huang, H. Yang, Z. Pang, IoT-enabled dual-arm motion capture and mapping for telerobotics in home care, IEEE J. Biomed. Health Inform. 24 (6) (2020) 1541–1549.

[59] E. Martinez-Martin, M. Cazorla, A socially assistive robot for elderly exercise promotion, IEEE Access 7 (2019) 75515–75529.

[60] Y.-H. Byeon, D. Kim, J. Lee, K.-C. Kwak, Ensemble three-stream RGB-S deep neural network for human behavior recognition under intelligent home service robot environments, IEEE Access 9 (2021) 73240–73250.