2025 17th International Conference on Computer and Automation Engineering

Calculating System Health Metrics in Dynamic Environments

1st Shaine Christmas
School of Information Technology
Deakin University
Geelong VIC, Australia
schristmas@deakin.edu.au

2nd Robert Davidson
Defence Science and Technology Group (DSTG)
Edinburgh SA, Australia
rob.davidson@defence.gov.au

3rd Arkady Zaslavsky
School of Information Technology
Deakin University
Geelong VIC, Australia
arkady.zaslavsky@deakin.edu.au

4th Kevin Lee
School of Information Technology
Deakin University
Geelong VIC, Australia
kevin.lee@deakin.edu.au

***Abstract*—System Health management is increasingly important for distributed or remote systems management. While some System Health management techniques make inferences based on collected data, these inferences do not support situational awareness. This work presents a methodology and framework for calculating System Health, allowing for the separation of inference and health calculation, while still supporting the application of contextual information to System Health. This work explores this framework with a theoretical example, and application to real-world datasets. System Health largely depends on the weightings of components, which places importance on weight assignment.**

***Index Terms*—Situation-Awareness, Context, System Health, Resilience, Distributed Systems, IoT**

I. Introduction

Systems are becoming more intelligent by applying Situational Awareness technologies in many different problem domains [15]. Increases in edge devices, mobile systems and distributed systems, comes with a commensurate increase in data availability for context inference engines. In larger systems, System Health management can become an important tool for identifying, diagnosing, and addressing problems with the systems effectiveness and efficiency. While System Health is a concern for larger dynamic distributed systems, the marriage of these concepts to situation and self-aware systems has not been well explored.

This work defines a system's health as the capability to complete tasks to an expected standard. System Health must be calculated to inform decision support capabilities, and reduce the impact of environmental factors on mission success. Additionally, System Health may be utilised to track efficiencies in system operation, and further optimise decision support and resource allocation methods. System Health also applies to its resilience, as active monitoring of system resources can help identify and diagnose problems.

System Health management is most effective when supplied with accurate and up-to-date information. While raw data can aid in this understanding, System Health management may also benefit from contextual analysis. A rich understanding of a system's functionality and capabilities can lead to more robust decision support in complex environments where individual or groups of systems (e.g. autonomous systems) must make decisions. Search and Rescue (SAR) operations often occur in environments that are difficult to navigate and may pose dynamic and evolving risks to robotic and autonomous systems (RAS). This scenario would greatly benefit from the application of System Health, as decisions are likely to impact the operation of a system, especially in SAR operations with environmental hazards. Increasing the capability of individual system components within these environments enables more flexible decision support capabilities; and, hence, greater utility in these applications.

The contributions of the work presented in this paper are (i) a framework for identifying RAS health metrics, (ii) a method of calculating System Health based on data and contextual information, (iii) an application of the framework and method on a singular agent within a wider distributed RAS; (iv) an application of the framework and method to pre-existing datasets.

The remainder of this paper is as follows: Section II outlines the state of the art in System Health and Context Awareness; Section III outlines the proposed framework and applies it to a singular agent within a wider distributed system. Section IV explores pre-existing datasets and analyses the System Health calculated using the proposed framework. Section V concludes this work and discusses future applications.

II. State of System Health and Context Awareness

System Health covers the constant monitoring of the capabilities of system components [6] and functions based on locally available information and the utility of said components. System Health can be used in a number of different applications, from decision support technologies [7], routing algorithm optimisations [17], and node management for distributed

979-8-3315-3380-9/25/$31.00 ©2025 IEEE

architectures [18]. Types of data that can be used for System Health would include energy monitoring [3], communication failures [10] and application-specific information which may differ per deployment.

When approaching specific system types, such as RAS networks, criteria for calculating System Health must be selected based on the types of devices and nodes within the system. Domain specific feature selection is a common area of research, with researchers looking at space systems [12], aerospace and Unmanned Aerial Vehicle (UAV) applications [5], and self-driving vehicles [14].

Individual features can be used in various applications. Each application may use a number of different metrics, and techniques. These techniques include, but are not limited to:

- Machine Learning Models [16]
- Rule-Based decision support [19]
- Fuzzy inference methods [4]

One or more metrics may be used in these techniques.

System Health does not inherently support awareness functionality when dealing with Situationally Aware systems. Some applications that use System Health information build upon that information with context awareness [12]. As a result, context is inferred in-situ depending on the specific use of the data. Massé et al. [9] outlines how the OSA-CBM architecture tackles System Health management and prognosis. Context inference takes place during the data manipulation phase. This context inference is used for the sole purpose of the management and prognosis of System Health. However, this contextual information could be used in more advanced decision support processes. Situational awareness is an increasingly common area of research, a system's dynamic actions when aware of its environment [15]. To allow for situational awareness, a system must analyse its current state driven by data analysis techniques. Further, the analysis conducted must draw some inferences to enrich the information and provide a deeper understanding of the devices state within the environment. This state may pertain to the physical environment of the system and the internal state of the system. Contextual information can be used in decision support processes for more intelligent reactions to environmental and system changes.

When applied to System Health, situational awareness can give a better understanding of what a system or subsystem within a system is capable of given the environmental conditions. Understanding the mobility capabilities of a subsystem can provide a functional assessment of a subsystem for informing decision support analysis, and reallocation to a task that reduces the risk of further reduction in capability of the subsystem.

Contextual information can help increase understanding of what types of faults an in-field subsystem may encounter in unknown and un-mapped environments. This is especially prevalent in distributed systems, where locations could be dynamic and within variable environments.

III. Context Aware System Health

Three areas must be addressed to mitigate the concerns raised in Section II.

1) Understanding the metrics that support System Health for a given application.
2) Identifying what contextual information should be inferred to support the System Health metrics.
3) Weighting System Health metrics based on their classification when calculating an aggregate score for System Health

This section is separated into the following subsections: Subsection III-A outlines a possible method for defining and outlining System Health; Subsection III-B will apply this method to a distributed search and rescue scenario.

A. Calculating System Health

Health calculations for a system must consider a multitude of different pieces of information. These pieces of information include:

- The priorities or goals of the system.
- The subsystems (or components) of the system.
- Inferred and collected information.

Importantly, contextual information is not inferred within the System Health calculation. Any context that could be used in calculating System Health could also be used in decision support methods and other system objectives. As such, this decoupling means context is only inferred once per measurement, and decision support technologies can take advantage of externally and internally relevant contextual information.

$$Health = \sum_{i=1}^{n} w_i \times s_i \tag{1}$$

To account for the pieces of information above, the System Health can be calculated as a weighted sum of the utility of individual system components (as shown in (1)), where n is the number of total subsystems considered, w_i is the weight of the subsystem, and s_i is the total calculated utility of the subsystem.

This work uses three terms for describing the parts of a System Health calculation:

1) System: The largest unit for which System Health may be calculated within the current scenario.
2) Subsystem: A sub-part of the System, for which a value is required to calculate the larger System Health.
3) Component: A sub-part of the Subsystem, for which a value is required to calculate the Subsystem Health.

As these terms are relative to each other, a subsystem may also be defined as a system, or as a component, depending on the scenario. This model allows for a generalised depth of investigation. As such, the utility of a System (System Health), its subsystems, and the subsystems components are all restricted to values between 0 and 1. As a System may have an arbitrary number of subsystems (which in turn may have an arbitrary number of components), this model also allows for a generalised width of investigation. In both depth and width, the impact of individual subsystems and components is diminished with increasing the total number considered values.

Each subsystem contains its own weight, which may be changed to reflect a number of different criteria, including:

1) The importance of the subsystem to the overall system operation.
2) The significance of the subsystem to the execution of system tasks / mission tasks.
3) The priorities of the system.

For the first criterion, some subsystems will be inherently more important to system operation than others. In this case, subsystems that may require a higher utility score to operate may be weighted higher. At the same time, other systems may still contribute to an overall System Health calculation, but are less important to the operation of the system.

For the second criteria, some missions or tasks may require the use of particular subsystems within the systems capabilities. For these subsystems, weighting the System Health to reflect this perceived importance can give a better understanding of situational System Health, keeping in mind the overall system goals.

For the third criteria, a system may have specific goals that use specific subsystems. The calculation of System Health should consider active priorities to ensure that the inferred System Health is accurate with reference to the context of the current system status and tasks. Examples for each of these weighting components will be given in subsection III-B.

$$s_i = \sum_{j=1}^{m} w_j \times c_j \tag{2}$$

In order to use this method of calculating System Health, calculation of the subsystem utility must also take place. The proposed method of calculating subsystem utility also forms a weighted sum (as shown in (2)), where s_i is the individual subsystem, w_j is the weighting of a particular subsystem component, m is the number of components a subsystem contains, and c_j is the utility of the component. Defining a subsystem's components will rely on expert information depending on the individual subsystem. It may include raw, pre-processed, and contextual information if relevant to the subsystem in question. In this work, system depth of two is used (systems contain subsystems, which contain components). This methodology can be expanded to an infinite depth, as each constituent part of a system may be considered to be a individual system, subsystem or component. This use of the methodology is discussed in III-B. Each type of information used in a subsystem utility calculation must be normalised to a common scale. To normalise contextual information, a context model can be defined to 'rate' individual pieces of context [1].

For numerical context information, normalising the value for use in System Health calculation may be required. This can be done using a variety of different methods: distance from mean of possible values as a percentage; deviation of value from the standard deviation; Bernoulli Distribution analysis, and others. Any method may be used to create a representative value between 0 and 1. The accuracy of the calculated utility using a particular normalisation method may be severely impacted if the method does not appropriately represent the impact a reduction or increase of perceived utility has on System Health.

For categorical context information, any process of numeric encoding can occur to create a representative value. Some techniques include binning, feature classification and probabilistic reasoning. In the case of binning, ranking context categories based on perceived impact on system operation may occur. To further this, a context attribute may be treated as a component of its subsystem; allowing for features of the possible categories to be quantified and ranked in terms of impact on a favourable context category. For boolean contextual information, a probability of the context being true may also be assigned as its representative value. In each case, analysis of categorical context requires numeric encoding.

The utility of all components of a system should be a value between 0 and 1, with weightings causing the subsystem values to also fall between 0 and 1. To achieve this, the weights of all components of a system must add to 1. This should be the same for any high level system including the final System Health value.

B. System Health applied to Distributed Search and Rescue

SAR scenarios are a good example of systems that need to allocate resources, and complete tasks within a timely manner. Additionally, SAR scenarios can often occur in volatile environments, which may rapidly cause drastic System Health changes, requiring distributed nodes to prioritise specific actions over others.

For this work, the following example has been defined:

- 3 Robots are searching through a forest environment.
- The overall mission is to locate a missing person last seen within the search environment.
- A time limit of 2 hours has been defined due to an incoming wildfire.

Additionally, the following assumptions have been defined:

1) Each robot must track its own health.
2) Each robot can infer its own contextual information using locally available information.
3) Robots may communicate context with each other to inform System Health calculations and local context inference.
4) Robots will always aim to follow their goals and priorities.

Within this example, the following priorities have been defined:

1) Maintain communication
2) Complete goals
3) Optimise System Health

These priorities ensure that communication between nodes is maintained and that tasks may be completed at the cost of System Health. System Health should be optimised to ensure that system goals are completed in a way that minimises System Health reduction.

To calculate the health of the overall system at any given point, the following subsystems have been defined:

- Robot 1 Health.
- Robot 2 Health.
- Robot 3 Health.

In this case, each robot will calculate its Health, contributing to the overall System Health. To weight each robot, the starting Health may be considered, in addition to the robots understood capabilities. If robot 1 contains a module that robots 2 and 3 do not (such as a long range antenna), its weighting may be higher due to its perceived utility in expected conditions. For the remainder of this example, we will focus on the health calculation for Robot 1.

For an individual robot, System Health can be defined as the ability for a robot to complete its assigned goals, and hence the overall mission. A healthy robot should experience minimal System Health loss, and be able to complete tasks without requiring recalculation, or diversion from normal goals. An unhealthy robot would experience difficulty attempting to complete its goals due to environmental aspects beyond the robot's control, and/or physical hindrances to the robots capabilities. As all System Health values fall between 0 and 1, a point within this scale must be defined to classify the Health of robots and subsystems. For this work, a healthy system has a System Health greater than 0.6.

To calculate the Health of Robot 1, the following subsystems have been defined:

- **Environmental Sensing Subsystem:** What information has the robot obtained about its operating environment?
- **Robot Management:** What are the robots current capabilities?
- **Communication Subsystem:** How capable is the robot of communicating with other robots?

Each of these subsystems is important to the overall mission completion. Similarly to overall System Health calculation, the weighting of each subsystem will be influenced by the goals and priorities of the robot. In this case, the robot's and the system's priorities are aligned.

Each subsystem will be comprised of individual components. This example highlights how raw data, pre-processed data, and contextual information can all be used to aid in calculating a component's utility.

Fig. 1 outlines the example data that could be used for each of the three subsystems. Each piece of data falls into three data categories: raw data (such as humidity); pre-processed data (such as number of objects nearby); and contextual information (such as baseline robot utility). To this end, context must be inferred at some stage. Assumption 2 covers this inference. It can be assumed that context inference is separate from System Health calculation.

The weights for each subsystem and their constituent components are shown in Fig. 2. Within this scenario, weights have been defined based on perceived importance to System Health.

Subsystem	Component	Values
Environmental Sensing	Number of Objects nearby (as a fraction of max impact allowed)	Number = 5 Max = 10 Value = 5/10 = 0.5
	Temperature (as a placement on the range of values able to be read by sensor)	Number = 27 Range = 0 –> 60 Value = (27 - 0)/(60-0) = 0.45
	Humidity (percentage)	Number = 0.45 Range = 0-1 Value = 0.45
Robot Management	Robot Mobility Deviation (how standard is the velocity of the robot?)	Number = 0.85 Range = 0->1 Value = 0.85
	Battery Voltage (on the scale of battery max -> 0)	Number = 11.5 Range = 0 -> 12 Value = 0.958
	Battery Amperage (scale of battery max -> battery minimum)	Number = 45 Range = 30->50 Value = 0.75
	Baseline Robot Utility (Contextual Inference of usefulness of robot modules to completing tasks)	Number = 0.85 Range = 0->1 Value = 0.85
Communication Subsystem	Currently available nodes (as a fraction of total deployed nodes (excluding self))	Number = 1 Max = 2 Value = 1 / 2 = 0.5
	Available Bandwidth (bps, as a fraction of max supported by communication method)	Number = 900000 Range = 110 -> 921600 Value = 0.977
	Time remaining since last Communication (s, as a fraction of remaining Time-To-Live for communication packet)	Number = 246 TTL = 300 Value = 0.82

Fig. 1. Example data for a Search and Rescue Scenario

Subsystem	Component	Weight
Environmental Sensing		0.2
	Number of Objects nearby	0.6
	Temperature	0.2
	Humidity	0.2
Robot Management		0.5
	Robot Mobility Deviation	0.4
	Battery Voltage	0.2
	Battery Amperage	0.2
	Baseline Robot Utility	0.2
Communication Subsystem		0.3
	Currently available nodes	0.3
	Available Bandwidth	0.2
	Time remaining since last Communication	0.4

Fig. 2. Example weights of subsystems and components

Real-world applications may have dynamic weighting systems to change the weighting of subsystem and component utility functions. Additionally, weights that are informed by goals (the weights of each subsystem) may change if goals change.

Given these values and weights, the robot System Health for this snapshot in time may be calculated. Table I outlines the utilities as computed using the proposed weighted sum, using the data from Fig. 1 and Fig. 2.

Some interesting analysis can be done by looking at the utility of each subsystem. The communication subsystem seems quite healthy when looking at the available bandwidth and time since the last communication. However, as only 50%

TABLE I
DERIVED UTILITY OF EACH SUBSYSTEM USING THE EXAMPLE DATA AVAILABLE IN FIG. 1 AND WEIGHTINGS AVAILABLE IN FIG. 2.

Subsystem	**Utility**
Environmental sensing	0.48
Robot management	0.8516
Communication subsystem	0.6734

of the system nodes can be communicated with, the overall utility of the subsystem is drastically reduced. This reduction could be lessened by lowering the weight of the measurement of available resources within the communication subsystem. Alternatively, the robots within the system could be allocated differently to ensure that communication is maintained throughout the course of the mission.

$$\begin{gathered} Health = (0.2 \times env) + (0.5 \times robot) + (0.3 \times comms) \\ Health = (0.2 \times 0.48) + (0.5 \times 0.8516) + (0.3 \times 0.6734) \\ Health = 0.86782 \end{gathered} \tag{3}$$

Equation (3) outlines how the overall health of the robot may be calculated; with env representing the environmental sensing utility, $robot$ representing the robot management utility, and $comms$ representing the communication subsystem utility. The overall System Health was calculated as 0.86782, where only values between 0 and 1 are possible. This rates the robot as healthy and able to complete its goals.

IV. ANALYSIS OF REAL DATASETS

To show how System Health can change over time, the System Health measurement model has been applied to a fusion of 3 different pre-existing datasets, where each dataset is applied to one subsystem within the System Health calculation. These datasets and subsystem components are:

- Environmental Management Subsystem [8]
- Battery Utility [13]
- Communication Subsystem [11]
- Mission Management Subsystem: Manually generated

In addition to the externally sourced data, mission management data has been added to show how changes in goal completion may affect the overall System Health value. Fig. 3 shows each subsystem's calculated utility between 0 and 1 based on its constituent components. It should be noted that values outside this range may be considered outliers or anomalies, which could inform other System Health metrics.

After calculating each subsystems utility, the main Health calculation may take place. The weightings for each subsystem within this calculation will differ depending on the goals of the robot and system, and the perceived importance of the individual subsystems.

Fig. 4 demonstrates how System Health changes when prioritising battery utility. From this graph, it can be noted that the overall health of the system remains stable throughout the course of the scenario. This is despite the steady decline of battery utility shown in Fig. 3. As battery utility is part of a larger overall System Health measurement, this graph shows that the decrease in battery health does not outweigh the increase in other metrics, such as communication utility.

Fig. 5 outlines how the increase in communication health over time can affect the overall System Health calculation when more heavily weighted. This example shows that System Health increases significantly over time. The three other subsystems are fairly equal in their weighting, which may be noticed by looking at the similarities between the overall

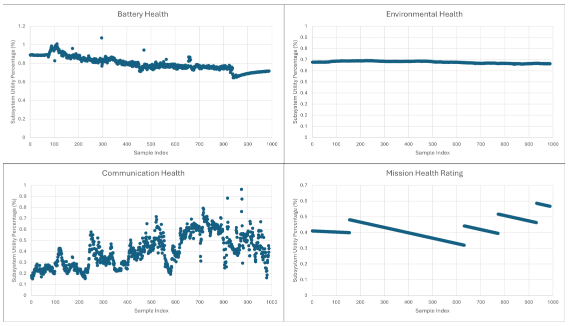
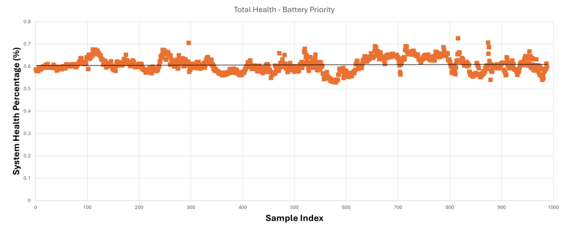
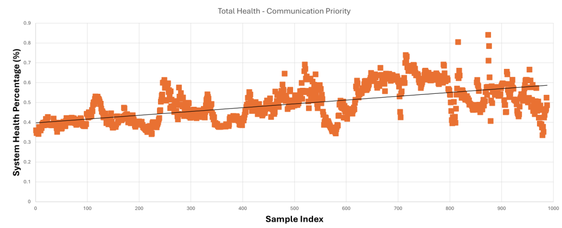


Fig. 3. The time series graphs of each subsystems utility (Battery Health, Environmental Health, Communication Health, and Mission Health respectively)



Subsystem	Weight
Environmental Subsystem	0.15
Battery Utility	0.4
Communication Subsystem	0.25
Mission Management Subsystem	0.2

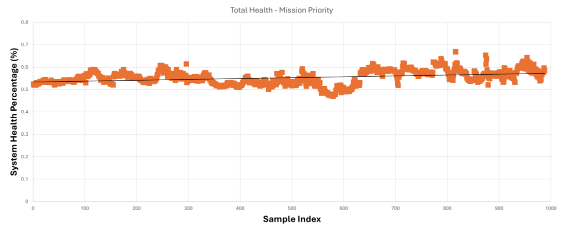
Fig. 4. System Health Graph and Weights for Prioritising Battery Utility



Subsystem	Weight
Environmental Subsystem	0.1
Battery Utility	0.15
Communication Subsystem	0.65
Mission Management Subsystem	0.1

Fig. 5. System Health Graph and Weights for Prioritising Communication Utility

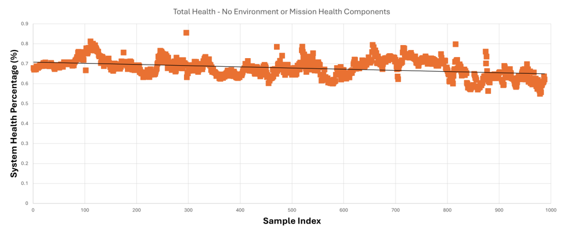
System Health and the communication utility graphs. The closer weightings for each subsystem are, the less a change in an individual subsystem calculated Health may have an impact on the total System Health.



Subsystem	Weight
Environmental Subsystem	0.15
Battery Utility	0.25
Communication Subsystem	0.2
Mission Management Subsystem	0.4

Fig. 6. System Health Graph and Weights for Prioritising Mission Utility

Fig. 6 shows how System Health can increase moderately based on the completion of system goals. Even while the battery utility negatively impacts the System Health, this measure gradually increases over time, as more goals are being completed. The increase identified at the beginning of the graph may also be attributed to the increase of the other subsystems at this point in the dataset. Changes in overall System Health are affected by the increase or decrease of values, as well as their relative weightings. The impact of mission tasks is highlighted by the fact that towards the end of the scenario, the measured health trends higher than in other graphs, where a falloff in System Health may be observed.



Subsystem	Weight
Environmental Subsystem	0
Battery Utility	0.7
Communication Subsystem	0.3
Mission Management Subsystem	0

Fig. 7. System Health Graph and Weights for only utilising Robot Metadata

Finally, Fig. 7 outlines what System Health could look like without using other variables, such as environmental and mission management subsystems. This may be closer to existing System Health calculation methods. In this case, System Health is steadily declining. Battery health is weighted more heavily within this calculation, as for this example maintaining node operation is more important than maintaining node communication. This decline outlines how environmental and mission data is important for understanding the total capabilities of a robot, within the context of its environment and mission. Similarly to individual robot System Health, Total System Health in multi-robot systems is a weighted aggregate of the System Health of each robot.

Weights may be assigned using multiple factors: number of subsystems a robot contains; importance of the individual robot to system goal completion; projected System Health decline given estimation of utility cost of robot tasks; and others. This weighting will not only change per deployment environment, but also with structure and size of the deployed system. Sufficiently large multi-robot systems may even be grouped into smaller multi-robot subsystems: increasing the depth of System Health calculation required to adequately measure the capabilities of all components of the system. Further, dynamic weight assignment can occur to increase accuracy in dynamic environments. Weightings may be changed according to anomaly detection methods, machine learning, environmental sensing, or manually by a system manager (human or coordinator device). Once changed, recalculation can occur determining the current System Health based on the new weight parameters. This will differ based on deployment, and may be useful for tuning decision support, state estimation, or other factors within a deployed system.

This weighted sum approach may not accurately encapsulate the nonlinear characteristics of complex systems by itself. However, as each component is calculated individually, these calculations may use unique methods, including but not limited to nonlinear techniques, averaged data-over-time, and machine learning techniques. This work presents this framework as a standardised method of calculating System Health, but allows for integrating unique and flexible component utility calculation depending on the deployed component, and deployed environment.

Overall, this is an effective method of analysing System Health, especially dependent on the context of this systems use and environment. The weighting of individual subsystems (as well as the weighting of their constituent components) is important to define, as it can drastically change the recorded value and trends of System Health over time. For these datasets, an accurate reflection of System Health will need to reflect the application of the system, and hence should be either decided by using experts opinion or automatically by considering system priorities. Machine Learning can be a ideal technology for furthering dynamic and real-time weight tuning, depending on identified patterns, or even other contextual information. Refinement of these weights is required for increased accuracy in real-world environments.

V. Conclusion and Future Work

This work presents a complete methodology and model for understanding System Health in situationally aware environments: by using raw data and contextual information to draw inferences on wider system capabilities. Analysis of system environments can lead to the identification of subsystems that may affect the health of the system. Each subsystem should have its own components, which may include raw data, contextual information or a subsystem which requires its own System Health calculation. All information used for

System Health description must be weighted according to its perceived impact on system capabilities and normalised to reduce data bias. Weight assignment based on perceived importance to priorities and system operation works well if the assignments originate from expert opinion. In the case of seemingly arbitrary assignments, aspects of the system that may not significantly affect the ability of the system to operate may have a higher impact on System Health than initially desired. Conversely, essential subsystems may have a reduced impact due to low weighting of the component or its related subsystem. Additional research will be required for domain-specific subsystems and components, depending on the individual system application and domain.

Creation of a singular, real-world system to collect data for interpreting System Health should be explored. In designing such a system, the health components should be considered before sensor and data selection to help refine the calculation of subsystem utility. Additionally, such a system should include some context inferences, to better inform System Health of the overall system and environmental contexts. Applying this System Health model to existing and developing situationally aware systems can help increase the available contextual information to support decision-making and create more dynamic reactions to system and environmental states. Future work will focus on context inference and application to the Lightweight Context Management Architecture (presented in [2]).

References

[1] A. M. Baddour, J. Sang, H. Hu, M. A. Akbar, H. Loulou, A. Ali, and K. Gulzar, *Cim-css: A formal modeling approach to context identification and management for intelligent context-sensitive systems*, IEEE Access, 7 (2019), pp. 116056–116077.

[2] S. Christmas, R. Davidson, A. Zaslavsky, and K. Lee, *Tackling network challenges in context aware environments: Lightweight context management architecture*, in 2023 24th IEEE International Conference on Mobile Data Management (MDM), 2023, pp. 334–339.

[3] S. D. Ghode and K. Bhoyar, *Nema: Node energy monitoring algorithm for zone head selection in mobile ad-hoc network using residual battery power of node*, in 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 1999–2004.

[4] D. T. Ho and J. M. Garibaldi, *Context modelling in fuzzy systems*, in 2012 IEEE International Conference on Fuzzy Systems, 2012, pp. 1–7.

[5] D. Jing and W. Haifeng, *System health management for unmanned aerial vehicle: Conception, state-of-art, framework and challenge*, in 2013 IEEE 11th International Conference on Electronic Measurement & Instruments, vol. 2, 2013, pp. 859–863.

[6] R. Kothamasu, S. H. Huang, and W. H. VerDuin, *System health monitoring and prognostics — a review of current paradigms and practices*, The International Journal of Advanced Manufacturing Technology, 28 (2006), p. 1012–1024.

[7] Q. Liang and Q. Ren, *Energy and mobility aware geographical multipath routing for wireless sensor networks*, in IEEE Wireless Communications and Networking Conference, 2005, vol. 3, 2005, pp. 1867–1871 Vol. 3.

[8] A. L. Majdik, C. Till, and D. Scaramuzza, *The zurich urban micro aerial vehicle dataset*, The International Journal of Robotics Research, 36 (2017), pp. 269–273.

[9] J.-R. Massé, B. Lamoureux, and X. Boulet, *Prognosis and health management in system design*, in 2011 IEEE Conference on Prognostics and Health Management, 2011, pp. 1–5.

[10] M. Panda and P. M. Khilar, *Energy efficient soft fault detection algorithm in wireless sensor networks*, in 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012, pp. 801–805.

[11] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, *Beyond throughput: a 4g lte dataset with channel and context metrics*, in Proceedings of the 9th ACM Multimedia Systems Conference, MMSys '18, New York, NY, USA, 2018, ACM, p. 460–465.

[12] K. Reichard, E. Crow, and T. Bair, *Integrated management of system health in space applications*, in 2007 Annual Reliability and Maintainability Symposium, 2007, pp. 107–112.

[13] T. A. Rodrigues, J. Patrikar, A. Choudhry, J. Feldgoise, V. Arcot, A. Gahlaut, S. Lau, B. Moon, B. Wagner, H. S. Matthews, S. Scherer, and C. Samaras, *Data Collected with Package Delivery Quadcopter Drone*, 7 2020.

[14] N. Salowitz, Z. Guo, S.-J. Kim, Y.-H. Li, G. Lanzara, and F.-K. Chang, *Bio-inspired intelligent sensing materials for fly-by-feel autonomous vehicles*, in SENSORS, 2012 IEEE, 2012, pp. 1–3.

[15] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, *Context-aware computing, learning, and big data in internet of things: A survey*, IEEE Internet of Things Journal, 5 (2018), pp. 1–27.

[16] A. Stief, J. R. Ottewill, and J. Baranowski, *Relief f-based feature ranking and feature selection for monitoring induction motors*, in 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR), 2018, pp. 171–176.

[17] G. Wang, M. Shao, R. Li, Y. Ma, and B. Wang, *Spray and wait routing algorithm based on transfer utility of node in dtn*, in 2015 IEEE International Conference on Progress in Informatics and Computing (PIC), 2015, pp. 428–432.

[18] C. Xiao, N. Chen, and K. Wang, *Design and implementation of geospatial sensor web node management prototype system*, in 2013 21st International Conference on Geoinformatics, 2013, pp. 1–5.

[19] X. Zhang, H. Xia, and S. Cai, *An decision-support system based on hybrid reasoning architecture*, in 2010 2nd IEEE International Conference on Information Management and Engineering, 2010, pp. 335–339.