Comparison of Agent Deployment Strategies for Collaborative Prognosis

Maharshi Dhada
Institute for Manufacturing
Department of Engineering
University of Cambridge
Cambridge, U.K. CB3 0FS
Email: mhd37@cam.ac.uk

Marco Pérez Hernández
Institute for Manufacturing
Department of Engineering
University of Cambridge
Cambridge, U.K. CB3 0FS

Adrià Salvador Palau
Department of Physics and Technology
University of Bergen
Allégaten 55, 5007 Bergen, Norway

Ajith Kumar Parlikad
Institute for Manufacturing
Department of Engineering
University of Cambridge
Cambridge, U.K. CB3 0FS

***Abstract*—Collaborative prognosis is a technique that enables the industrial assets to learn from similar other assets in a fleet, and improve their data-driven prognosis models. When collaborative prognosis is implemented in a computationally distributed framework, each asset is monitored by its corresponding Digital Twin agent. Distributed collaborative prognosis is particularly beneficial for high value assets where the communication and the processing costs are negligible compared to the maintenance costs. This paper analyses the effects of Digital Twin deployment strategies on the effectiveness of predictive maintenance activities relying on distributed collaborative prognosis. Distributed and heterarchical multi-agent system architectures are analysed for large fleets of assets, with varying failure rates and noise levels in the failure data. The results show that no single architecture or deployment strategy can be deemed best across all failure rates and noise levels. The conclusion derived in this paper provides guidance to the asset owners to choose the most suitable combination for a given application.**

***Index Terms*—Collaborative Learning, Digital Twins, Prognosis, Operations Research**

I. Introduction

Predictive maintenance is characterised by real-time maintenance scheduling when the corresponding asset failure hazards rise over a certain threshold. It minimises unwanted preventive maintenance for healthy assets, and also expensive corrective maintenances by avoiding asset failures. As a result, the predictive maintenance policies have proven to be more cost-effective for the service providers, compared to conventional fixed maintenance plans. Predictive maintenance policies primarily rely on data-driven prognosis, where the statistical models are used to learn from historical failure data and predict the oncoming asset failures [1].

Nevertheless, the industrial processes are non-ergodic systems. In the sense that the average behaviour of independent assets in the fleet is not representative of any single asset. As a consequence of this non-ergodicity, an asset's failure behaviour can be best described by a model trained using its own failure data. But an individual asset often does not experience sufficient failures for its statistical model to learn the failure behaviour. In these circumstances, collaborative prognosis is a technique that enables the assets with insufficient failures to identify similar other assets in the fleet and learn from their data. State-of-the art collaborative prognosis involves clustering similar assets, followed by exchanging the failure data within these clusters [1], [2].

The literature presents multiple instances where researchers have proposed collaborative learning for industrial asset operations. [3] showed that in a system comprising of multiple assets and historical failures, prediction of a given asset is improved by identifying similar historical behaviours from a library of past failure data, and evaluating the best fit for the current failure's degradation curve. [4] used a genetic algorithm to identify clusters of the most similar historical failure trajectories, which in turn improved the prediction accuracy of models corresponding to each of those identified clusters. An example implementation of this was shown for fatigue crack growth, drilling bit degradation, and degradation of turnout system applications. [5] relied on collaborative learning to tackle the lack of sensing resources for the overall cohort of units, for the cases of both medical patients and industrial assets. Markov models and selective sensing were used to address the problem of incomplete data per individual units. The reader may refer to [1] for further insight about collaborative prognosis.

Collaborative prognosis can be computationally distributed across the asset fleet using industrial multi-agent system (MAS) frameworks, where each asset is controlled by its corresponding Digital Twin agent. The distributed nature makes collaborative prognosis more adaptable, scalable, resilient, flexible, and lean compared to a centralised application [6], [7].

Operational costs of MAS architectures with varying extents of computational distributions were compared by [8] to identify the suitable architecture for a given asset value. [8] concluded that distributed collaborative prognosis is particularly beneficial for a fleet of high value assets. This is because the resulting increase in communication and computation costs due to data exchange and model training, are more than compensated by the reduction in corrective and preventive maintenance activities.

However, apart from the underlying MAS architecture, asset

This research was supported by the Next Generation Converged Digital Infrastructure project (EP/R004935/1) funded by the Engineering and Physical Sciences Research Council and BT.

failure characteristics also affect the overall costs of the operations. For example, the severity of asset failure determines the cost of its preventive or corrective maintenance. In most applications, an asset has multiple failure profiles, each mode with its own corresponding downtime [9]. This is shown in [10] where the maintenance cost is evaluated as a function of time to repair. Failure frequency is another characteristic that determines the total maintenance cost. [11], [12] highlight the use of frequency of failure events and downtime as key criteria for maintenance policy decision.

Similar to the failure characteristics, several strategies for deploying the assets' Digital Twin also exist. Cloud, Fog and Edge computing infrastructures are the main strategies for deploying distributed computing systems [13]. Cloud strategy provides almost unlimited computing and storage resources, away of the source where data is generated [14]. Whereas the Fog and Edge computing strategies push the computations to the end points of the network, which in our case are the assets. The distinction between Fog and Edge is not always clear, however both strategies aim to bring computation closer to the data generation, with benefits such as reducing communication latency [15].

To the best of the authors' knowledge, the literature does not present analyses for the implications of the asset failure characteristics and the Digital Twin deployment strategies on predictive maintenance activities relying on collaborative prognosis. This paper analyses the implications of asset failure characteristics and Digital Twin deployment strategies on a predictive maintenance plan relying on distributed collaborative prognosis. Simulated asset fleets were used to study the aforementioned effects, which allowed analysing the effect of increasing asset failure frequencies. The logic used for the simulations enable the industries to gauge the implications of the agent deployment strategies suitable for their systems.

Section II describes the simulation parameters used for the experiments, in line with the standard predictive maintenance policies and collaborative prognosis technique. Description of the experiments conducted and the results obtained are presented in sections III and IV respectively. The results are discussed in Section V. Finally, Section VI summarises the key conclusions and the potential future research directions.

II. Simulation Description

Open source MAS simulation software Netlogo was used to model the temporal evolution of a fleet of 500 high value assets, according to the parameters described in this section. The prognosis and clustering algorithms were run at the corresponding agents using Python extension for Netlogo. The dynamic simulation model used for the experiments is detailed in this section.

A. Simulation Setup

It must be noted that the simulations were performed for the cases where distributed collaborative prognosis has been deemed beneficial, i.e. for the fleets comprising of high value assets. MAS architectures analysed in the experiments were the heterarchical and distributed MAS architectures, presented in Fig. 1 and Fig. 2. Characteristically, the presence of a central agent overlooking the asset fleet network differentiates the heterarchical architecture from the distributed. The following agent types comprise the distributed and heterarchical MAS architectures:

- **Digital Twins**, which are the components that run the predictive and maintenance planning algorithms for their corresponding assets.
- **Social Platform,** wherever present, is responsible for enabling communications across the system.

Further information about these architectures can be found in [8].

An asset's health was represented using a synthetic health index, assumed to be evaluated using various operational parameters like in [16], [17]. A white noise was added to this health index to represent noisy data, based on which the Digital Twins would learn the corresponding failure behaviours/ function parameters. Asset health indicators degraded according to inverse exponential functions, described as:

$$HI_i(t_{li}) = a_i \left(1 - e^{-b_i(t_{fi} - t_{li})}\right) + \epsilon_{0,\sigma}, \tag{1}$$

where, t_{li} is the local time of the i^{th} asset, i.e. the time since the last corrective repair or installation. (a_i, b_i, t_{fi}) are the parameters that determine the shape of Health Indicator function. b_i is the curvature parameter, and a_i determines the expected value of HI_i at $t_{li} = 0$. t_{fi} is the average (or expected) time of failure. $\epsilon_{0,\sigma}$ is a white noise term with standard deviation σ and 0 mean, conforming to Gaussian distribution.

It must be noted that the true health of an asset was a monotonously decreasing function, represented only by the first term in 1. The noise term was added only for the data analysed by the Digital Twin, to represent the noisy sensors. Assets failed when (true) $HI_i \leq 0$.

Asset failures corresponded to one of the three possible failure modes, each representing mild, moderate, and high severity of the failures. Whenever an asset failed, a weighted random choice with probabilities 0.6, 0.3, and 0.1 for mild, moderate, and high severity failures respectively was used to determine the failure mode. The failures were also associated with their own corrective maintenance downtimes, being 2, 4, and 6 time-steps respectively. This brought the simulated fleet closer to reality where an asset experiences diverse failures during its operations, each one with its own individual time to repair (TTR) [10]. The preventive maintenance activities on the other hand were associated with a downtime of only one time-step, which is later detailed in Section II-B.

The goal for a given Digital Twin was to estimate the function parameters (a_i, b_i, t_{fi}), and predict the impending failure of its corresponding asset. Collaborative prognosis was used for estimating the model parameters across the fleet, and a predictive maintenance policy was used to prevent the asset failures, details of which are presented later in Section II-B.

Apart from the failures in the assets, the Digital Twins were also simulated to fail. Digital Twin failures however were relatively straightforward, as these were software components that could fail suddenly without any noticeable degradation. Digital Twin failures followed a Bernoulli distribution at each time-step, with a failure probability of 0.1. The implication of a Digital Twin failure was their inability to communicate with other agents as well as to make any decisions.

B. Collaborative Prognosis and Maintenance Policy

Collaborative prognostics was implemented by clustering the assets, and subsequently sharing health indices data within the clusters. k-means clustering algorithm was used for clustering the assets, which was run by the Social Platform and the Digital Twins for the heterarchical and distributed architectures respectively. The health indices observed over the most recent 200 time-steps of the assets were used to cluster the assets. More information about the distributed k-means clustering implementation can be found in [8], [18]. The function parameters were subsequently estimated based on the cluster's data corpus using non-linear least squares fitting algorithm.

The assets were preventively repaired when their time since installation or last repair surpassed the predicted time of failure multiplied by a factor, η: $t_{li} > \eta t^{e}_{fi}$, $\eta < 1$, and corrective maintenance was conducted upon asset failure. Asset failures were associated with downtimes according to the severity. For the experiments discussed here, t^{e}_{fi} was the estimated time of failure of the asset i in the fleet, and η was set to a fixed value of 0.7. The preventive maintenances were associated with a downtime of one time-step.

C. Deployment strategies

Two Digital Twin deployment strategies were analysed. These strategies can be thought of as cloud or edge deployment, and are illustrated in Figures 1 and 2. Their behaviours and corresponding deployments are explained as follows:

- **Cloud Deployment**: This strategy is depicted in Fig. 1, where the Digital Twins of the corresponding architectures are deployed away from the assets they monitor. In this strategy, a failure in the asset would not affect the corresponding Digital Twin capabilities. While simulating the cloud deployment strategy, the agent failure and the asset failure were simulated independently.
- **Edge Deployment**: In this strategy it was believed that the assets possess sufficient computing infrastructure for the agents to be deployed directly on board. Such assets could have built-in computing infrastructure or have it retrofitted [19]. This strategy is depicted in Fig. 2, where the Digital Twins are deployed along with the assets that they monitor.

  Any asset failure in the case of edge deployment would propagate to the corresponding Digital Twin, causing it to cease its operation and hence the prognostics system capabilities for that asset. For the simulations, it was believed that any asset failure would cause its corresponding Digital Twin to fail. Since the Digital Twin had failed, the operators would not know what caused the failure, and would resort to the conservative estimate of a severe failure. Therefore, only severe asset failures were simulated for this strategy.

Similar assets in Figures 1 and 2 are shown with same symbols, and are connected to one another. However, when the assets have failed (marked with cross), it is shown how for the case of cloud deployment the agents are still working and the network connection is not severed. Opposite is the case for edge deployment, where asset failure implies agent failure as well because the Digital Twin agent is deployed on the asset.

III. Experiments

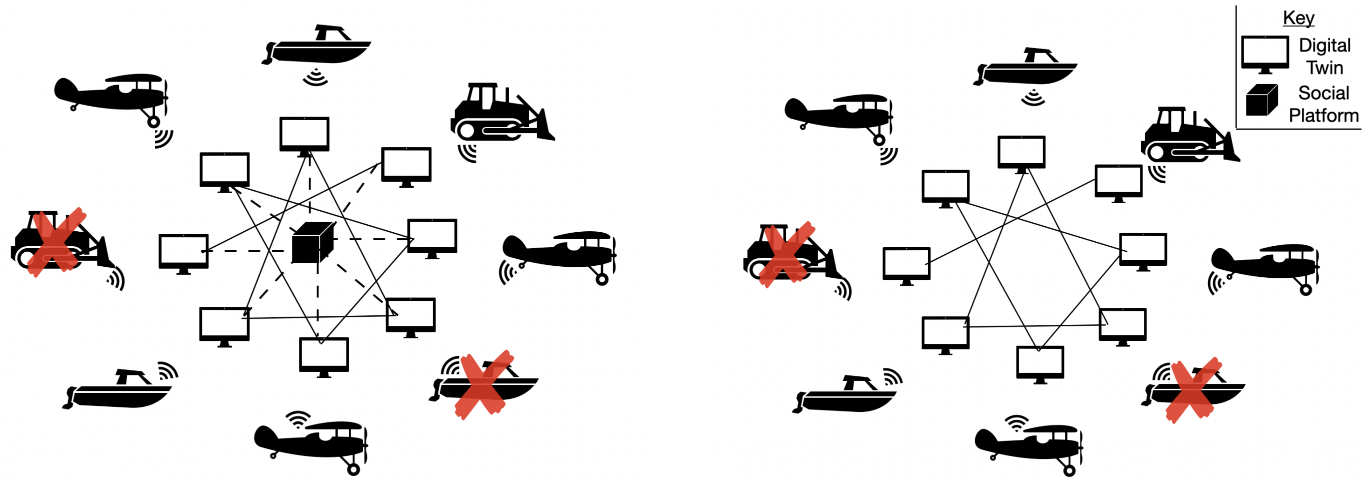
The parameter values for the various experiment cases are described in this section.

The simulated asset fleet was divided into four asset clusters. The clusters were each characterised by a corresponding set of (a_i, b_i, t_{fi}) parameters. The true cluster of any given asset was not known beforehand at the start of simulation.

Parameter t_{fi} represents the reference time to failure for the simulated fleet (t_{ref}). Assets with a higher reference time to failure would experience less number of failures (lower failure frequency), and vice versa. t_{fi} was varied across the experiment cases, with values in the ranges of $50, 100, 150, 200$. Apart from the (a_i, b_i, t_{fi}) parameters, the asset health degradation was also governed by the Gaussian noise term $\epsilon_{0,\sigma}$ as shown in (1). Since the first term in (1) was normalised to the a_i value, the Gaussian noise was characterised by mean 0, and a standard deviation $\sigma \in (0, 1)$ that was varied across the experiment cases along a range of values as $0.05, 0.1, 0.15, 0.2$.

Figure 3 shows examples of the degradation functions describing a simulated asset fleet across the range of noise variations, and also two extreme asset failure frequencies (i.e. $t_{fi} =$ 50 and 200). The functions are shown for the case where the four asset clusters are represented by different ranges of t_{fi} mentioned in the sub-captions. The noisy input used by the Digital Twins to estimate the true parameters is coloured red, whereas the dashed lines are the true health degradations. Digital Twins in the simulations discussed in this paper use the non-linear least squares fitting algorithm for estimating the (a_i, b_i, t_{fi}) parameters. The noisy function is illustrated for a single cluster and across the range of noise values. Moreover, the shaded green region in Figure 3 show the threshold for preventive maintenance, given the true parameter values. This time is shown for the function plot in black.

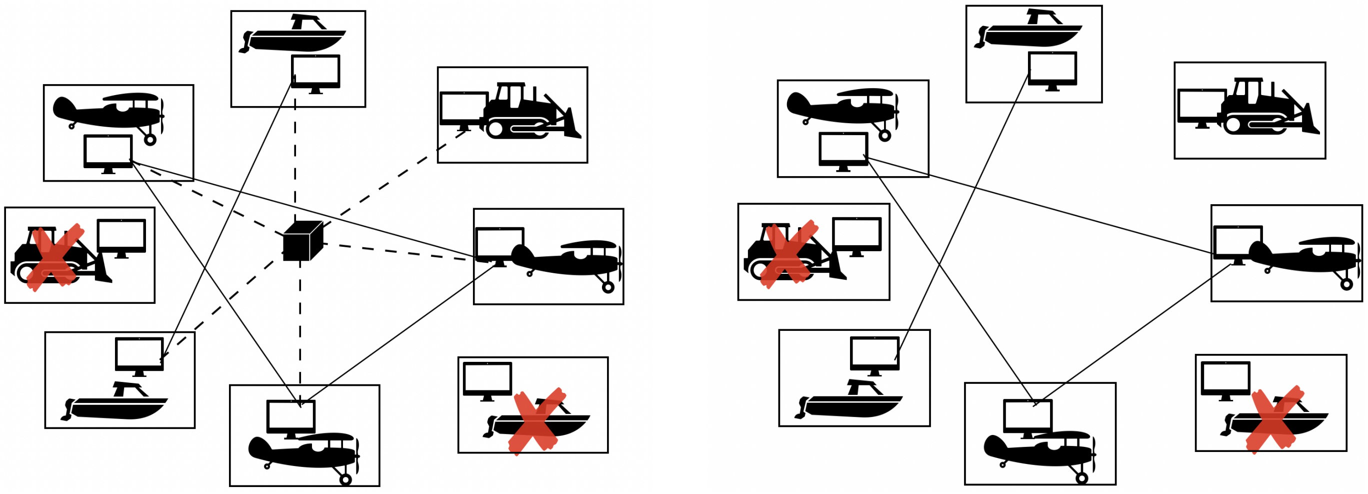
The simulations were run until either total number of failures exceeded 5000 or the total simulation time was greater than $t_{fi} * 10$ for the given simulation case. The simulation time was chosen as the expected number of failures per asset multiplied by its reference time to failure. Table I summarises the asset down times and the weights associated with each of the failure modes, including other governing parameter values



(a) Heterarchical architecture

(b) Distributed architecture

Fig. 1: Cloud Deployment Strategy



(a) Heterarchical architecture

(b) Distributed architecture

Fig. 2: Edge Deployment Strategy

discussed yet. Using these values, heterarchical and distributed architectures were simulated for the two agent deployment strategies. Each experiment case was simulated ten times with different randomness seeds.

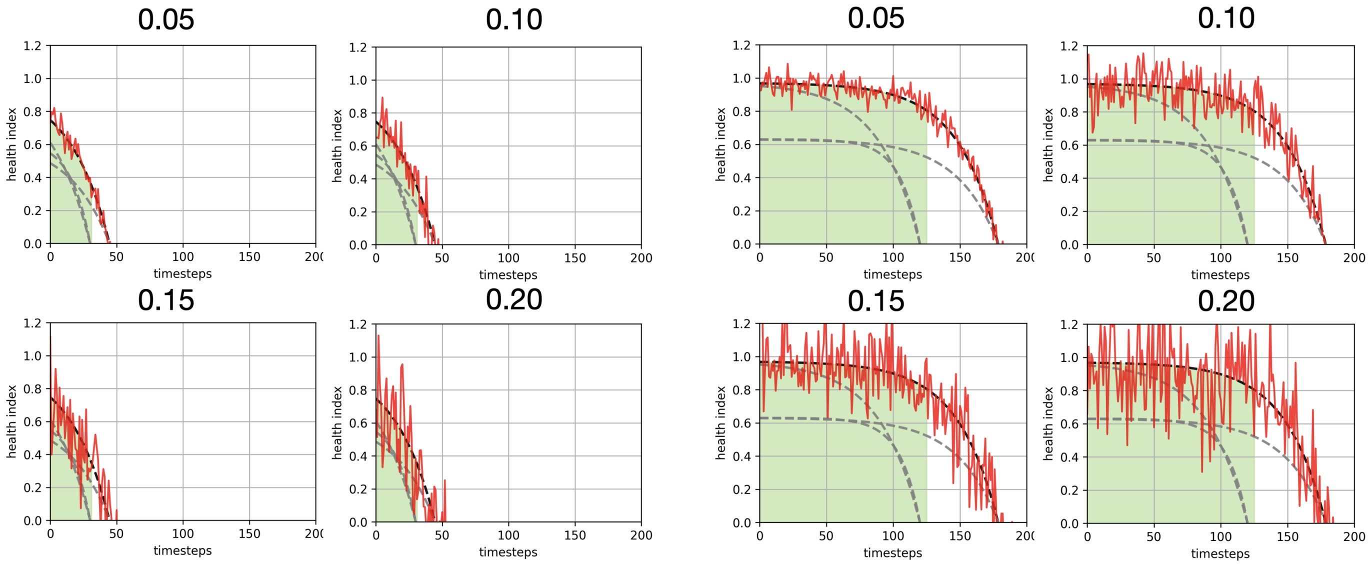
TABLE I: Simulation parameters

Number of assets	500
probability of agent failure	0.1
Runs per architecture	10
Noise standard deviations $(\epsilon_{0,\sigma})$	0.05, 0.10, 0.15, 0.20
Reference times to failures (t_{fi})	50, 100, 150, 200
Stopping condition:	
Total failures, or	5000
Simulation time	$t_{fi} * 10$
Corrective maintenance downtimes (and probabilities):	
Severe	6 time-steps (0.1)
Moderate	4 time-steps (0.3)
Mild	2 time-steps (0.6)

IV. Results

Results of experimental cases are presented in Figures 4 and 5 in the form of box plots. The box plots represent total corrective and preventive maintenance activities performed for each experiment case. Since these are high value assets, the determining factor of the overall operations costs are the maintenance activities. Each box plot is generated based on ten replications of the simulation case.

Common to both Figures 4 and 5, the x-axes indicate the agent deployment strategy, where Dist() and Het() indicate distributed or heterarchical architectures. The letters 'e' or 'c' in the parentheses denote edge or cloud Digital Twin deployment strategy. The reader must note that 'edge' and 'cloud' terminology here merely indicates whether the Digital Twin is deployed on the asset (in which case the Digital Twin fails when the asset fails) or not. The y-axes indicate



(a) t_{fi} in the range of 50

(b) t_{fi} in the range of 200

Fig. 3: Asset degradation function plots for high and low failure frequency assets. The noise levels are shown at the top of the corresponding plots. True health indices are shown in black and grey, whereas the noisy data are shown in red.

the corresponding maintenance activity counts. Furthermore, the reference times to failures increase from top to bottom subplots, and the data noise increases from left to right. The reader must be aware that the axes in Figure 4 are not evenly scaled, this is to improve the readability as the corrective maintenance counts vary greatly across the cases.

V. Discussion

The box plots in Figures 4 and 5 show that the maintenance activities are substantially higher when the noise (σ) is higher. Furthermore, the following conclusions can further be derived:

1) **Based on overall trends observed in the corrective and preventive maintenance activities**
   - Going from left to right in Figures 4 and 5, it is observed that the corrective maintenances (CMs) increase but no significant increase is observed in the preventive maintenances (PMs). This is because the white noise causes early or late predictions of asset failures with equal chances. As a result the number of PMs stay nearly constant throughout the noise variations. However, early predictions have no effect on the CMs because the failures can be prevented with early PMs. But late predictions cause an increase in CMs. The effectiveness of the PMs reduce with increasing noise, and are rendered almost ineffective in the extreme case seen in the top right plot in Figure 4.
   - The above effect is mitigated with decreasing failure frequency (going from top to bottom) because the assets spend more time in the high health region/ above the PM threshold (refer the green shaded region Figure 3). This enables the cumulative data across the asset clusters to be closer to the actual function, therefore better estimates of the function parameters can be derived by the corresponding Digital Twins.

2) **Based on the trends in corrective maintenance activities (the following conclusions are observed from Figure 4 only)**
   - With less noise, the cloud Digital Twin deployment strategy performs better than the edge Digital Twin deployment. This is because only severe failures are observed in the case of edge Digital Twin deployment, and the assets have an overall lesser uptime compared to the cloud counterparts. As a result the Digital Twins in cloud strategy are able to collect more data from the operating time of the assets. However, when the noise is high, the parameter estimation is challenging. This problem is amplified when the failure frequency is high and the assets are often wrongly clustered, leading to incorrect estimation of the parameters (the case of 50, 0.2 and to some extent 100, 0.2). The results show that the connected architectures perform better in these cases.
   - For higher failure frequencies (upper row), the distributed architecture performs better than the heterarchical, the reason for which is attributed to system resilience. A failed central agent in the heterarchical case cannot update asset clusters, leading to suboptimal preventive maintenance activities. This disadvantage becomes clear as the noise increases as the PMs become increasingly ineffective.
   - However, the above mentioned conclusion does not hold for lower failure frequencies because the time

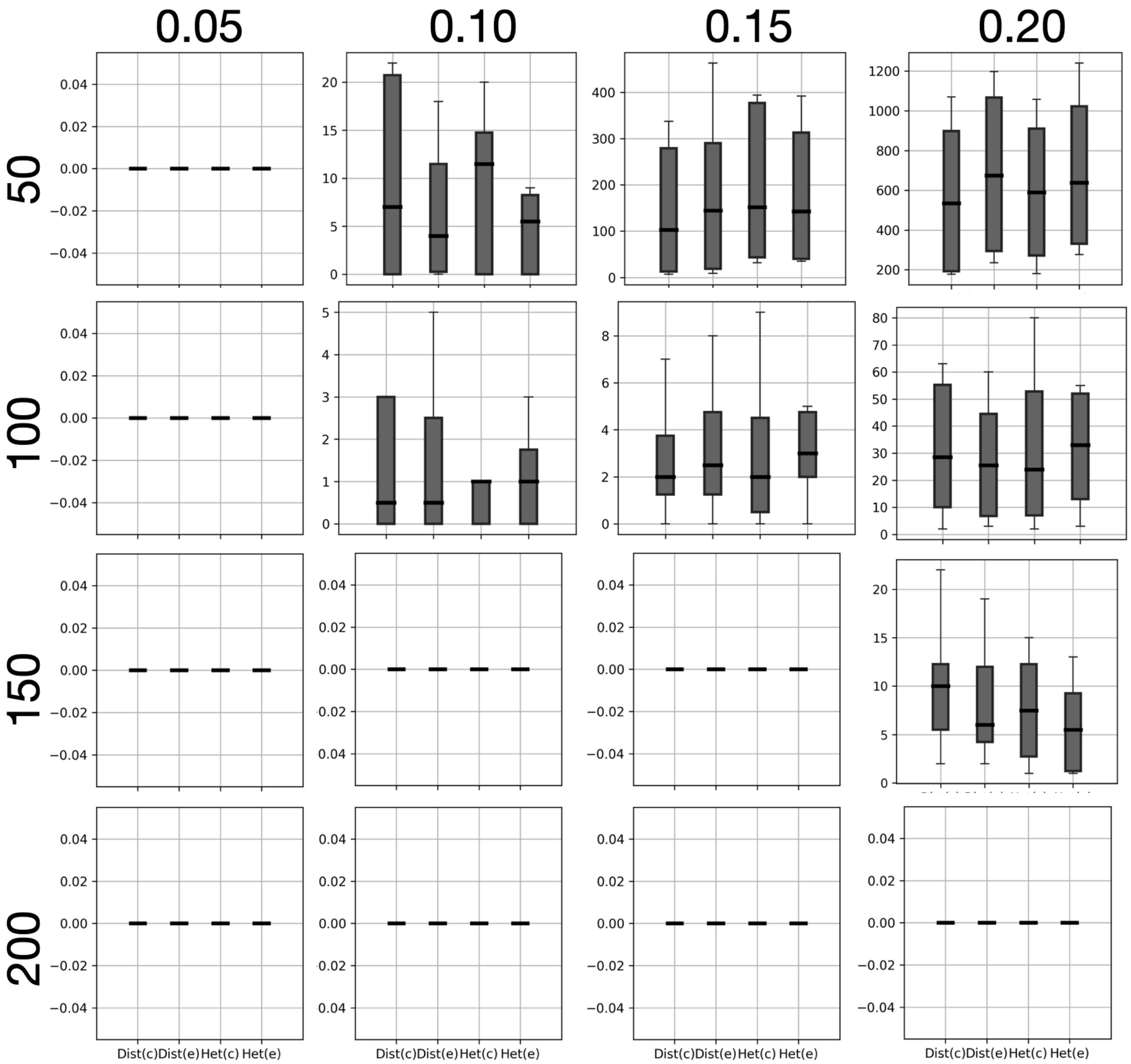


Fig. 4: Corrective maintenance counts across the experiment cases. The noise levels are mentioned at the top, and the T_{fi} ranges are mentioned on the left

lost upon agent failure is negligible with respect to the average time between failures of the assets. In these cases, the heterarchical architecture has shown to perform better.

In summary, it is concluded that the choice of architecture and the Digital Twin deployment strategy is governed by the failure frequencies and the noise levels observed in the operations data.

VI. Conclusions and Future Work

A. Conclusions

This study analysed the effect of Digital Twin deployment strategies and failure characteristics on the operational cost of collaborative prognosis based predictive maintenance. Results help industries determine suitable Digital Twin deployment strategies for their asset failure profiles and agent architectures. Key conclusions from the discussions are reiterated in Figure 6.

B. Future Work

Following research stems from the very limitations of this research, including:

- **Granularity of failures:** Degradations and the failures of the entire assets were simulated in the experiments discussed here. This granularity could be reduced to simulate particular cases where the assets experience

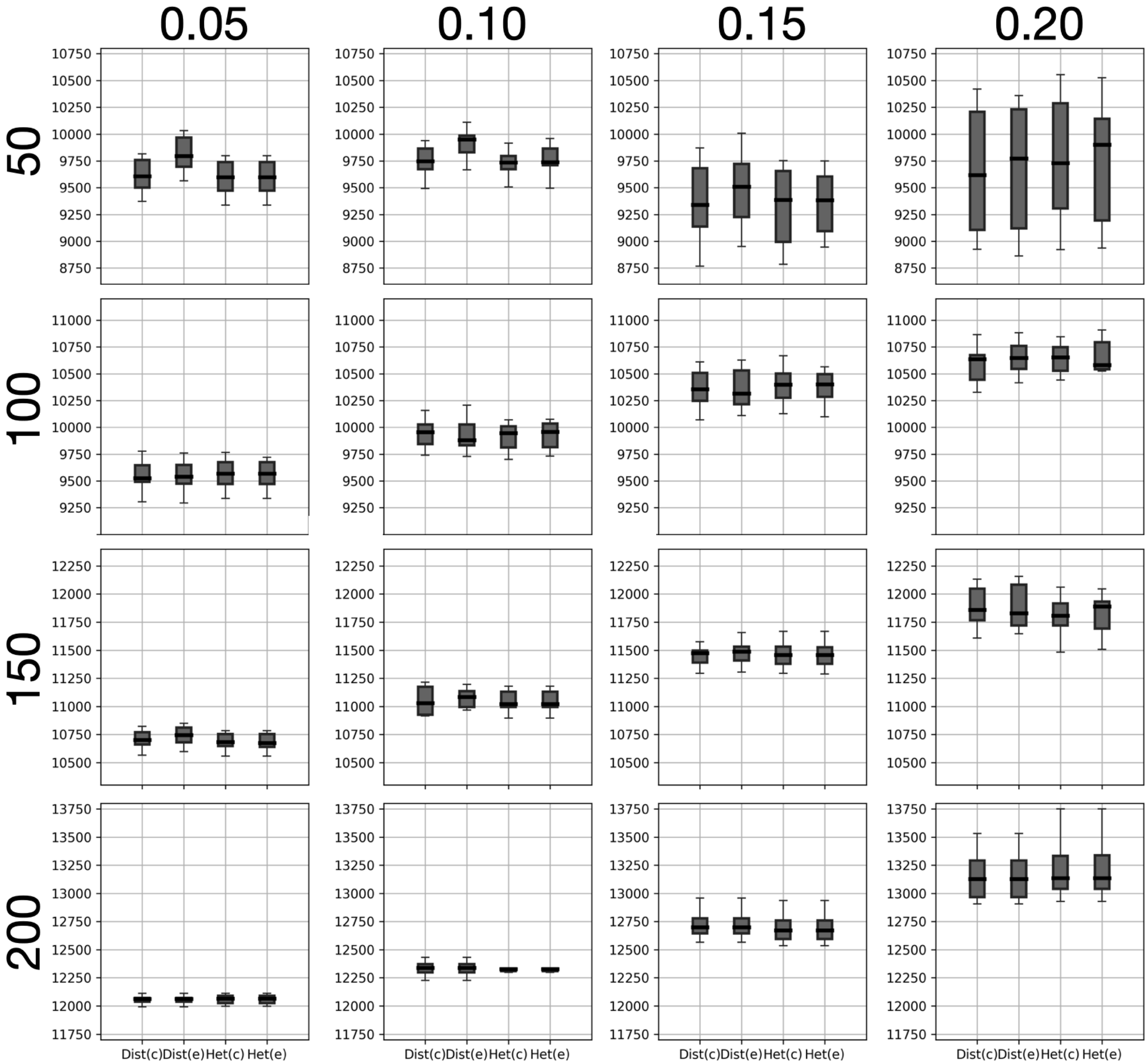


Fig. 5: Preventive maintenance counts across the experiment cases. The noise levels are mentioned at the top, and the T_{fi} ranges are mentioned on the left

partial failures, or keep operating under a sub-optimal performance.

- **Cost analyses:** Current analyses only compares the maintenance activities. Depending on the type of assets, the maintenance costs can be used as another comparison metric for the evaluations. The cost can also include the total uptime or production output of the assets.
- **Analysing a fleet of real assets:** Simulated assets were used for the current analyses, enabling only comparative conclusions across the architectures and the agent deployment strategies. Future work can extend this comparison for the fleets comprising of real assets.

Acknowledgment

The authors thank Dr Manuel Herrera at the Institute for Manufacturing for his feedback and help.

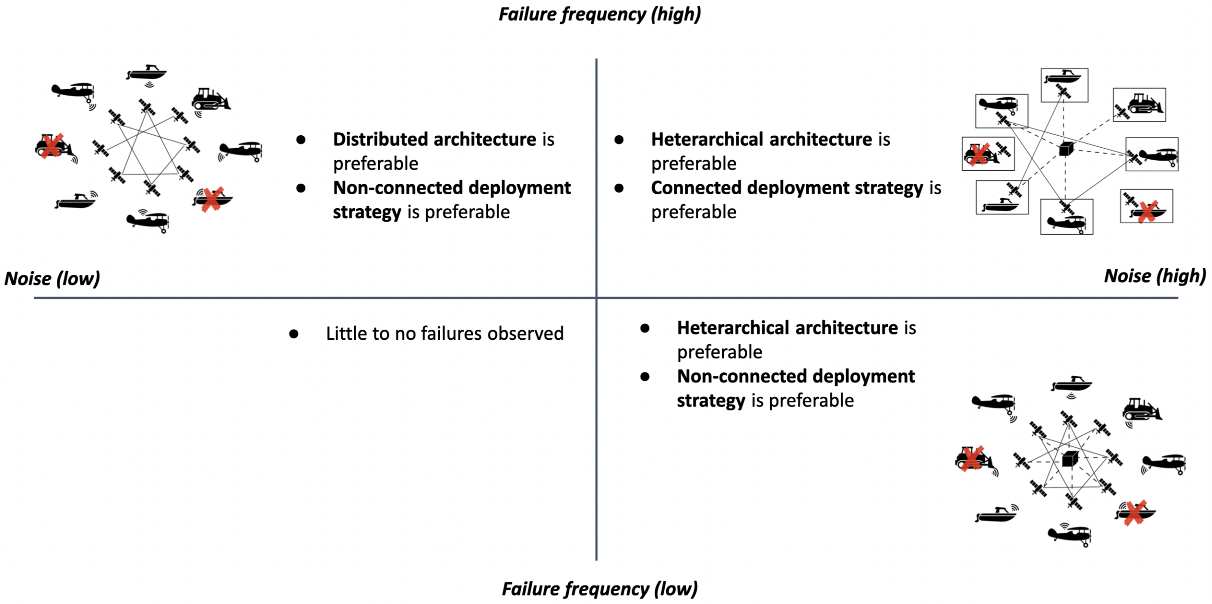

Failure frequency (high)
Distributed architecture is preferable
Non-connected deployment strategy is preferable
Heterarchical architecture is preferable
Connected deployment strategy is preferable
Noise (low)
Noise (high)
Little to no failures observed
Heterarchical architecture is preferable
Non-connected deployment strategy is preferable
Failure frequency (low)

Fig. 6: Summary of conclustions

References

[1] A. Salvador Palau, "Distributed collaborative prognostics," Ph.D. dissertation, University of Cambridge, 2020.

[2] M. Dhada, A. K. Jain, M. Herrera, M. P. Hernandez, and A. K. Parlikad, "Secure and communications-efficient collaborative prognosis," *IET Collaborative Intelligent Manufacturing*, 2020.

[3] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," in *2008 International Conference on Prognostics and Health Management, Denver, USA*, 2008.

[4] O. F. Eker, F. Camci, and I. K. Jennions, "A Similarity-based Prognostics Approach for Remaining Useful Life Prediction," in *Second European Conference of the Prognostics and Health Management Society, Nantes, France*. PHM Society, may 2014.

[5] Y. Lin, S. Liu, and S. Huang, "Selective sensing of a heterogeneous population of units with dynamic health conditions," *IISE Transactions*, vol. 50, no. 12, pp. 1076–1088, dec 2018.

[6] A. S. Palau, M. H. Dhada, K. Bakliwal, and A. K. Parlikad, "An industrial multi agent system for real-time distributed collaborative prognostics," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 590–606, 2019.

[7] M. Herrera, M. Pérez-Hernández, A. Kumar Parlikad, and J. Izquierdo, "Multi-agent systems and complex networks: Review and applications in systems engineering," *Processes*, vol. 8, no. 3, p. 312, 2020.

[8] A. S. Palau, M. Dhada, and A. Parlikad, "Multi-Agent System architectures for collaborative prognostics," *Journal of Intelligent Manufacturing*, 2019.

[9] J. A. Nachlas, *Reliability engineering: probabilistic models and maintenance methods*. Crc Press, 2017.

[10] K. Upasani, M. Bakshi, V. Pandhare, and B. K. Lad, "Distributed maintenance planning in manufacturing industries," *Computers and Industrial Engineering*, vol. 108, pp. 1–14, 2017.

[11] A. Labib, "A decision analysis model for maintenance policy selection using a CMMS," *Journal of Quality in Maintenance Engineering*, vol. 10(3), 2014.

[12] A. Rastegari and M. Mobin, "Maintenance decision making, supported by computerized maintenance management system," in *2016 Annual Reliability and Maintainability Symposium (RAMS)*, 2016, pp. 1–8.

[13] P. Escamilla-Ambrosio, A. Rodríguez-Mota, E. Aguirre-Anaya, R. Acosta-Bermejo, and M. Salinas-Rosales, "Distributing computing in the internet of things: cloud, fog and edge computing overview," in *NEO 2016*. Springer, 2018, pp. 87–115.

[14] D. C. Marinescu, *Cloud computing: theory and practice*. Morgan Kaufmann, 2017.

[15] I. Sittón-Candanedo, R. S. Alonso, J. M. Corchado, S. Rodríguez-González, and R. Casado-Vara, "A review of edge computing reference architectures and a new global edge proposal," *Future Generation Computer Systems*, vol. 99, no. 2019, pp. 278–294, 2019.

[16] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–6.

[17] J. Yan, M. Koc, and J. Lee, "A prognostic algorithm for machine performance assessment and its application," *Production Planning & Control*, vol. 15, no. 8, pp. 796–801, 2004.

[18] J. Qin, W. Fu, H. Gao, and W. X. Zheng, "Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multiagent consensus theory," *IEEE transactions on cybernetics*, vol. 47, no. 3, pp. 772–783, 2016.

[19] D. H. Arjoni, F. S. Madani, G. Ikeda, G. D. M. Carvalho, L. B. Cobianchi, L. F. Ferreira, and E. Villani, "Manufacture equipment retrofit to allow usage in the industry 4.0," *Proceedings - 2017 2nd International Conference on Cybernetics, Robotics and Control, CRC 2017*, vol. 2018-Janua, pp. 155–161, 2018.