Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

MESSAGE PRIORITIZATION IN CONTESTED AND DYNAMIC TACTICAL NETWORKS USING REGRESSION METHODS AND MISSION CONTEXT

Rohit Gopalan
Md Hedayetul Islam Shovon
Benjamin Campbell

Defence Science and Technology Group
Third Avenue
Edinburgh, SA 5111, AUSTRALIA

Vanja Radenovic
Kym McLeod
Leith Campbell

Consilium Technology
147 Pirie Street
Adelaide, SA 5000, AUSTRALIA

Dustin Craggs
Claudia Szabo

University of Adelaide
Ingkarni Wardli Building
North Terrace Campus
Adelaide SA 5000, AUSTRALIA

ABSTRACT

Military communications at the tactical edge consists of unreliable, disrupted, and limited bandwidth networks, which can lead to the delay and loss of critical information. These networks are increasingly being used for the transmission of digital command and control (C2) information, requiring timely and accurate transmission, and play a vital role in the outcome of military operations. Machine Learning (ML) techniques have the potential to improve operational outcomes by autonomously prioritizing the delivery of the most important information through these networks, using observations of the current mission and network state. This paper covers the experimental process and the operational metric used for comparison between the ML and a non-ML approach that sorts messages in a fixed order. We present two regression-based supervised-learning methods that were shown to be more effective in both medium and high congested networks than the non-ML approach.

1 INTRODUCTION

Command and control (C2) information management between the commanders and assigned forces plays a vital role for a successful defence operation. When voice-only tactical radios were used for communication, commanders exchanged C2 information through this medium ranging from urgent calls for evacuation to routine resupply requests. Commanders controlled the information flows based on their own interpretations and knowledge of the current operational status, network load and status, information priority and the vocal responses of other network users (Judd et al. 2018).

The advent of digital messaging enabled the transfer of different types of messages over radio networks. Counter intuitively, if managed poorly, this digitization can degrade the timely flow of C2 information. Firstly, human decision makers can be overloaded by the vast range of messages which can come through digital media. Secondly, the network itself can be overloaded due to the restricted and intermittent performance of mobile networks operating in congested and contested radio frequency (RF) environments in complex terrain without fixed infrastructure. Without efficient utilization of both the cognitive capacity of human operators and the information delivery capacity of networks, overload of either can result in compromising the timely delivery of important C2 messages (Judd et al. 2018).

Effective cognitive capacity is important for friendly soldiers as they need to be dynamically adaptive, agile and robust within a very challenging environment, subject to significant and disruptive dynamic changes (Beautement et al. 2005). Moreover, given the limited nature of military communications networks there is a need for prioritization of messages in order to deliver important information in a timely manner and improve mission outcomes. More specifically a system is required that can automatically prioritize the messages that have a high impact on the current mission and can reduce the potential cognitive burden.

Previous work (Judd et al. 2018; Judd et al. 2019; Szabo et al. 2020) presents Semantically Managed and Resilient Tactical Networks (SMARTNet), a distributed middleware that prioritizes C2 information based on network and mission context. An instance of SMARTNet operates on every network node, gathering information on network status, mission context and the environment to determine what information should be prioritized and delivered. Section 2 will discuss in depth how SMARTNet has evolved in solving the prioritization problem using contextual information.

Our primary contribution in this paper is the application of regression-based supervised learning techniques using mission context for solving the distributed prioritization of message delivery. The use of regression for prioritizing information has been demonstrated in various domains such as healthcare (Bagula et al. 2016) and network security (Renners et al. 2017). We demonstrate that these regression-based techniques can effectively assist the message prioritization better than a non-ML approach that relies on a fixed ordering to sort messages.

The metric used to assess the performance of these techniques is covered in-depth in Section 3. Section 4 discusses the generation of both training data and the applied methods. The effectiveness of the ML based regression techniques is presented in Section 5. Section 6 provides a brief conclusion and future work.

2 BACKGROUND AND RELATED WORK

2.1 Background

Humans are good at understanding and applying context from various information sources to make decisions that can potentially affect outcomes (Schaefer et al. 2019). Commanders, as the example we discussed in Section 1, effectively demonstrated this during the era of the voice-only tactical radios. The context they applied was based on the current mission status, how military networks work, what information gets prioritized over others and the vocal response received from subordinates. However, the same thing cannot be said of Artificial Intelligence (AI) systems especially within high uncertainty and unstructured operations (Schaefer et al. 2019). Nevertheless, integrating such AI is especially important considering the increasingly complex and dynamic technological environments that modern soldiers are expected to operate within (Judd et al. 2018; Schaefer et al. 2019).

Schaefer et al. (2019) have integrated context-driven AI into the human-machine teaming research program, Robotics Collaborative Technology Alliance (RCTA). They developed a module that facilitates the collection of information from the environment, mission and social context; encouraging joint decision making and collaborative operations between humans and AI powered robots. The AI approaches that have been applied include Speech/Gesture Recognition and Natural Language Processing (NLP). By providing various sources of communication such as audio, visual and tactile via these AI approaches, interactions between humans and robots can be made more robust while ensuring message delivery and shared situational awareness (Schaefer et al. 2019).

The RCTA program suggests that it is possible to utilize context-driven AI within complex and dynamic military operations. Szabo et al. (2020) and Judd et al. (2019) proposed artificial intelligence approaches that aim to facilitate communications across digitized tactical networks within contested and congested networks. These solutions aim to apply both mission and network context for prioritizing and delivering C2 information.

2.2 SMARTNet for Using Context for Prioritization

In its most basic mode of operation, SMARTNet uses fixed rules to control the rate at which position information messages (blue spots) are generated while sorting all messages in the dissemination queue based on fixed prioritization rules. These fixed rules will be referred to from here on as the baseline SMARTNet. Judd et al. (2018) enhanced SMARTNet so that it dynamically adjusted priorities based on mission context. This enhancement was tested against the baseline in simulations where two platoons carry out operational phases of Advance, Assault and Pursuit. Based on the location error metric (see section 2.3), this enhancement achieved a lower location error in most cases. However certain corner cases resulted in unexpected interactions of the dynamic rules causing higher location errors than the static prioritisation baseline. This suggests that while dynamic prioritisation based on human designed rules has the potential to improve network performance, creating the rules is non-trivial.

As an alternative to the dynamic rule based approach, Machine Learning (ML) techniques have been investigated to solve the prioritization problem. Judd et al. (2019) proposed a solution that addresses the prioritization problem. This solution applies features derived from the current mission context and the contents of the queue, to determine the need to re-prioritize the message dissemination queue. Triggered every time a node's mission or network context changes, this solution uses Support Vector Machines (SVM) to return a binary value. As the classification measures such as accuracy, precision and recall returned very good results for at least 10,000 training samples, it shows that supervised learning can be used for solving similar problems. However, this method simply determines when re-prioritization should occur but does not carry it out due to the limitations of the SVM classifier. An additional mechanism is required to conduct the re-prioritisation task. This provides the motivation towards the exploration of regression based techniques.

Szabo et al. (2020) applied Evolutionary Algorithms (EA) to learn optimal network bandwidth for sending messages across the network. Using network context, the EA returns a ratio of how much bandwidth should be allocated for friendly node positions, enemy detection, text and tactical graphic messages. EAs were demonstrated to be effective over the baseline which dictates when friendly node positions are sent across the network while sending messages in a fixed order. The improvement shown by the EAs over the baseline was approximately 49%. While the EAs were effective in applying network context, these algorithms may have observed greater improvement if current mission context was also applied.

2.3 Metrics used within SMARTNet

Judd et al. (2019) applied classification based metrics of accuracy, precision, recall and F1-score to measure the effectiveness of the SVM model that determines when to re-prioritize the message dissemination queue. F1-score is the primary metric used to assess the performance of classification-based supervised learning models. This value is obtained by combining both precision and recall (Fujino et al. 2008). Even though the model was demonstrated to be effective, no metrics were applied for comparing the model with the baseline. For comparing a ML solution against the baseline, suitable metrics are required.

Both Judd et al. (2018) and Judd et al. (2019) attempted to resolve this issue by applying metrics to measure the effectiveness of their proposed solutions. These metrics consist of average location error, max position error, message latency and total position location information (PLI) messages sent. While Judd et al. (2018) measured the effectiveness of the dynamic rule-based approach to prioritization, Judd et al. (2019) measured the effectiveness a Double Deep-Q Network (DDQN) based reinforcement learning (RL) solution that aims to control the rate at which updated PLI messages get sent across the network. These applied metrics though, were limited to PLIs; to be more general, other message types need to be considered. This will necessitate developing and applying a metric that assesses the performance of prioritizing various C2 information types. This metric can be then used for comparing ML solutions against the baseline.

This limitation was resolved by Szabo et al. (2020) through the extension of the prioritization problem towards Text Messages, Tactical Graphics and Enemy Detection messages. Moreover, a generic metric, the Mid-level Metric (MLM) was introduced to assess the performance of C2 message dissemination based on information type, timeliness, identity of the receiving node and the current state of the operational environment within a SMARTNet scenario. This metric was used for comparing the EA algorithms against the baseline. The MLM is more generic than the metrics used by Judd et al. (2018) and Judd et al. (2019), which was more focused towards the PLIs. The MLM will be expanded upon in Section 3 and will be used for comparing the regression methods against the non-ML approach in Section 5.

3 MEASURING THE EFFECTIVENESS OF C2 DISSEMINATION

To be able to compare the performance of different C2 information management strategies in prioritizing C2 information across tactical networks we first need to have an understanding of what constitutes good performance. This has always been a challenging area of research, as C2 information management performance is dependent on a range of factors including, importantly, its impact on the operational outcomes (Alberts et al. 2002). Simply measuring the effectiveness of information delivery itself provides an incomplete picture as delivering more information does not necessarily improve performance - the usefulness of the information to the recipient is a critical consideration (Baroutsi 2015). Likewise, issues of simulation fidelity will occur while attempting to simulate the impacts of C2 information management on operational outcomes. As a consequence, these issues will greatly impact the usefulness of the simulation results.

Szabo et al. (2020) propose the mid level metric(MLM) framework to mitigate these issues. The MLM combines message delivery metrics with an SME-derived ruleset and outputs a score to indicate the expected impact of network utilization on the overall operational outcome. The MLM framework is comprised of two parts - a scenario generator that generates random simple unclassified tactical scenarios for SMARTNet that require the dissemination of C2 information between nodes, and a scoring system to score the performance of a C2 information management strategy over a generated scenario. For this work, we have extended the MLM framework to include a concept of operational context. This extended framework will be used for experimentation in Section 5.

3.1 Mid Level Metric Scenario Generation

The MLM scenario generator creates simple tactical scenarios for use within SMARTNet, based on an unclassified version of SME and Australian Army doctrine derived rules. The scenarios can be generated with any number of nodes and duration, and each node within the scenario will steadily generate C2 information required to be transmitted to other nodes. Table 1 shows the five C2 information types, and their generation rules. Within the scenario the nodes utilise a simple tactical movement algorithm, which is a modified random walk that keeps nodes close to other nodes within echelon-appropriate distances, and vehicle-appropriate velocities. For this work, we have also added enemy movement and detection to the scenario generator. The enemy nodes utilise a similar movement algorithm to friendly nodes and will keep within the highest echelon appropriate distance within the scenario to the friendly nodes. To simplify data generation patterns, every 10 seconds each of the enemy nodes is assigned to be detected by a random friendly node.

3.2 Mid Level Metric Scoring

In addition to generating scenarios, the MLM provides a scoring system that evaluates the C2 information management performance on the generated scenario. The scoring system works by recording information

Table 1: Mid-level metric scenario generation.

Information Type	**Destination / Type**	**Generation Rate**	**Generation Rules**	**Payload Size** ($m_{payload}$)
Friendly Node Position	Multicast to all	Each node generates 1 each second	Simple tactical movement	36 bytes
Enemy Detection	Multicast to all	Each enemy detected once per 10 seconds by random node	Simple tactical movement	81 bytes
Text	Unicast to random node hierarchically adjacent	Each node generates on average 1 per 5 minutes	Each node equal weight	500 bytes
Tactical Graphics	Unicast to units one level down in hierarchy	Each command node Platoon Leader (PL) and above generates on average 1 per 1 minutes	Each node equal weight	50 bytes
SOS	Multicast to all	Each node generates on average 1 per 20 minutes	Leaf nodes only	1 byte

delivery statistics of the five scenario-generated C2 information types, and scoring it using an SME-derived ruleset based on four factors: information type, the importance of the information to the receiver, the time taken to receive the information, and how the current operational context changes the importance of the information. Table 2 shows how the factors are calculated and Table 4 shows how operational context is calculated and how it impacts on the scoring at the node level. The combined local score, which is the basic MLM score multiplied by the operational context modifiers, is applied as a penalty to the node responsible for disseminating that information. The global MLM penalty is the average MLM score, calculated by summing up the MLM scores of all nodes divided by the number of nodes in the network. The global MLM score will be used for experimentation in Section 5. The MLM scoring system can give scores in real time (once per second) as the scenario is progressing in order to support reinforcement learning, or after completion of the scenario. The MLM uses the knowledge described in Tables 1, 2 and 4 to generate its scores. As SMARTNet is a distributed system, the MLM is unsuitable for controlling prioritisation, however it can be used in training an ML model for distributed prioritisation.

4 PROTOTYPE IMPLEMENTATION

This section covers the implementation of the prototype used in our investigation into using regression methods to apply context for message prioritization. Firstly, we discuss the scenarios we have simulated for both training and experimentation. Then we outline the applied methods for generating the models using the obtained comma-separated values (CSV) data from the training scenarios.

4.1 Simulation Scenarios

For both training and experimentation, we have generated 20-minute scenarios with hierarchical network topologies consisting of 13 friendly nodes (as shown in Figure 1) and 40 friendly nodes. These nodes will move through the environment, detect enemies (six enemies for 13-node networks and 20 enemies for 40-node networks) and periodically generate messages to be sent across a basic network simulation environment with network congestion being low, medium and high. To simulate congestion we control

Table 2: Mid-level metric scoring implementation at source node level.

Information Type	**Score Measurement**	**Time modifier**	**Receiver Modifier**
Friendly Node Position	0.1 point per meter of distance error from source node to all other nodes in the network	None	Penalty reduced by 20% for each 100 meters after the first 100 meters (ground truth distance to node) / Penalty reduced by 20% for each node distance in hierarchy
Enemy Detection	0.2 point per meter of distance error of each enemy the source node is supposed to detect	None	Penalty reduced for each 100 meters after the first 100 meters, Penalty Reduction starts at 20%, reduced by 4 percentage points for echelon (chain of command hierarchy) level
Textual Messages	5 points per second	Decays by 0.5 points per minute	None (either 100% or 0% if addressee or not)
Tactical Graphics	5 points per second	Decays by 0.1 points per minute	None (as above)
SOS	20 points per second	Penalty decays by 20% every 1 minute	None (as above)

T_p, which is the period in seconds between each node sending a packet. At every T_p, the highest priority messages in the queue, up to a total of 1500 bytes, are added to a packet and broadcast. T_p in training equals 0.1, 1 and 10 seconds for low, medium and high congestion scenarios respectively. To prevent model overfitting, T_p for testing in low, medium and high-congested scenarios is set to 0.4, 2 and 15 seconds. For 13-node networks, we generate 100 scenarios for each network congestion level, giving a total of 300 generated scenarios. For 40-nodes, we generate 50 scenarios for each congestion level, giving a total of 150 scenarios. The data generated from the training scenarios, is saved as CSV data for each message and context type to be used for training the models (discussed in Section 4.2). Each CSV data file contains the features and labels defined in Tables 3 and 4.

4.2 Model Generation

Table 3: Message types with features and labels.

Message Type	**Features**	**Label for $C_{message}$**
Friendly Node Position	Distance Travelled Since Last Update, Average Node Distance (average distance from one node to other nodes) & Average Hierarchical Distance (average number of hops from one node to other nodes)	Score
Enemy Detection	Distance Travelled Since Last Update, Average Node Distance & Average Number of Hops in Network	Score
Text	Number of Seconds Since Created Message	Score
Tactical Graphics	Number of seconds since created message	Score
SOS	Number of seconds since created message	Score

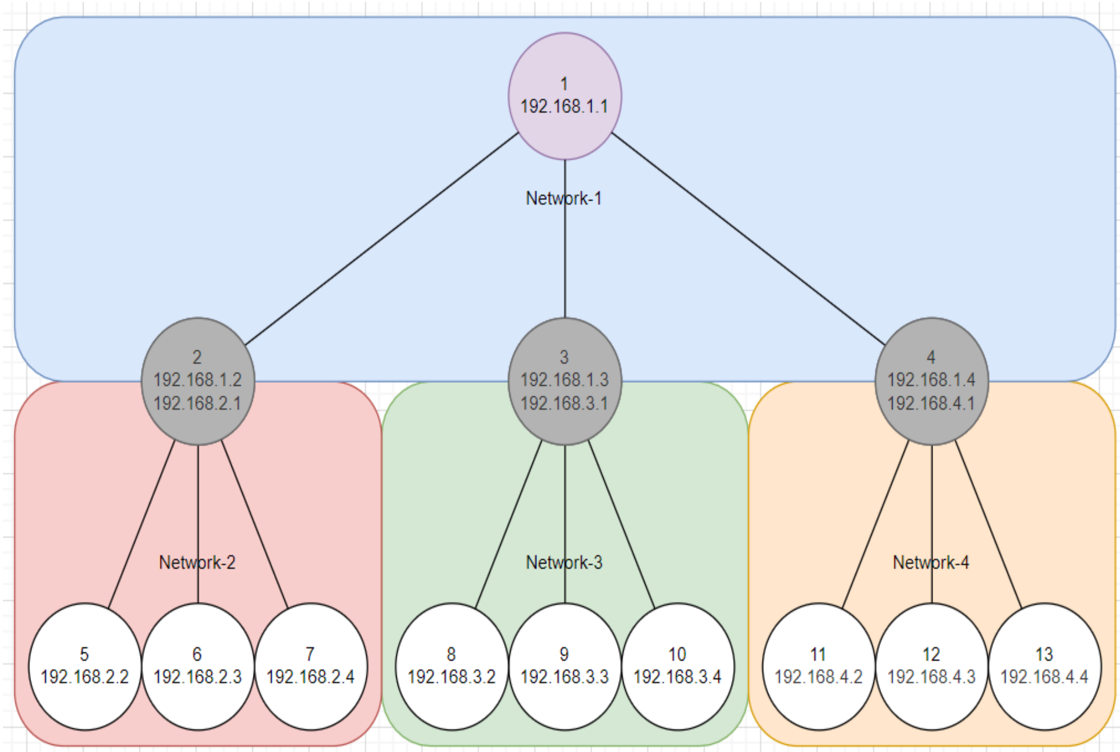

Figure 1: A 13-node hierarchical network topology resulting in four interconnected networks.

Two exemplar regression methods have been applied to generate models for distributed message prioritization. They are Multivariate Regression (Bagula et al. 2016) and Decision Trees (Renners, Heine, and Rodosek 2017). These two methods represent a basic linear regression method and a hybrid method, respectively (that can identify a set of rules for predicting specific outputs). Both algorithms utilize the Scikit-Learn API, a Python-based ML library (Pedregosa et al. 2011; Buitinck et al. 2013). We have developed individual models for all message and context modifier types using the generated CSV data from the 300, 13-node and 150, 40-node scenarios described in Section 4.1. The CSV data is neither standardized or normalized prior to model fitting. The functions that compute the outputs for the labels for each message and context type at training time are based on the MLM scoring functions described in Section 3.2. At deployment time, each node will have eight (five message type and three context type) models of either Linear Regression or the Decision Tree method as the features and the label of all message and context types differ from each other.

Table 4: Context modifier types with features and labels.

Context Modifier Type	**Features**	**Notation for Multiplier Label**
Distance to last Enemy Detection	Distances of nearest 5 enemies from friendly node	$M_{distanceToEnemy}$
SOS Modifier	Number of Seconds since Last sent SOS Message	M_{SOS}
Aggregate Modifier	Number of other enemies within a particular enemy's radius of 600 metres	$M_{aggregate}$

Initially we considered the Random Forest method (Ho 1995). However, based on the findings by Smith, Ganesh, and Liu (2013) in the field of neuroscience, Random Forest was found to be computing inaccurate concentrations of nine neurochemicals compared to the Linear Regression method. Through the application of the Coefficient of Determination metric, known as the R^2, these predictions were also found to vary greatly unlike the Linear Regression method. Therefore, we came to the conclusion that a method like Random Forest may not suitable for prioritizing C2 information.

5 EXPERIMENTATION & RESULTS

The objective of this experimentation is to investigate the effectiveness of both Linear Regression and Decision Tree models in understanding context for developing predictions that assist message prioritization. These models will be embedded within a SMARTNet module that will generate Friendly Node Positions every second and will sort messages (in descending order) based on their cost by message size ratio defined in (1).

$$\frac{C_{message}M_{total}}{m_{payload}}, \text{ where } M_{total} = \begin{cases} 1, & \text{if SOS Message} \\ M_{SOS}M_{aggregate}, & \text{if Enemy Detection Message} \\ M_{SOS}M_{distanceToEnemy} & \text{otherwise} \end{cases} \tag{1}$$

Each node will have five message type and three context type models of either the Linear Regression or Decision Tree method that will compute predictions for (1). All message type models will be predicting $C_{message}$ while all context modifier types will be predicting M_{total}. The key question from our experimentation is whether our regression methods can assist in message prioritization based on the features from both the message and operational context.

Across 300, 13-node scenarios (100 for each congestion level of low, medium and high) and 150, 40-node scenarios (50 for each congestion level), these models were compared against our baseline, a non-ML approach that generates and sends Friendly Node Positions every second (as long as there is nodal movement of at least a metre) and sorts messages in a fixed order where SOS messages are prioritized first followed by Text, Tactical Graphics, Enemy Detection and Friendly Node Positions. This baseline, previously used in (Judd et al. 2018; Judd et al. 2019; Szabo et al. 2020), was derived through input from military SMEs.

Figure 2 depicts a comparison of both the regression-based methods against the baseline. Based on this graphic, it is observed that both methods achieved a lower global MLM penalty compared to the non-ML approach (our baseline) across all medium and high congested scenarios. In addition, the Decision Tree method also achieved a lower global MLM penalty compared to the baseline in the low congested 40-node scenarios. This indicates that we have observed improvement by at least one ML method within these scenarios.

Table 5: Results showing level of improvement by ML methods over the non-ML approach.

# Nodes	Congestion	Linear Regression	Decision Tree
13	Low	-326.49%	-9.05%
13	Medium	9.27%	24.16%
13	High	3.07%	2.61%
40	Low	-67.17%	26.70%
40	Medium	9.50%	18.58%
40	High	11.58%	7.10%

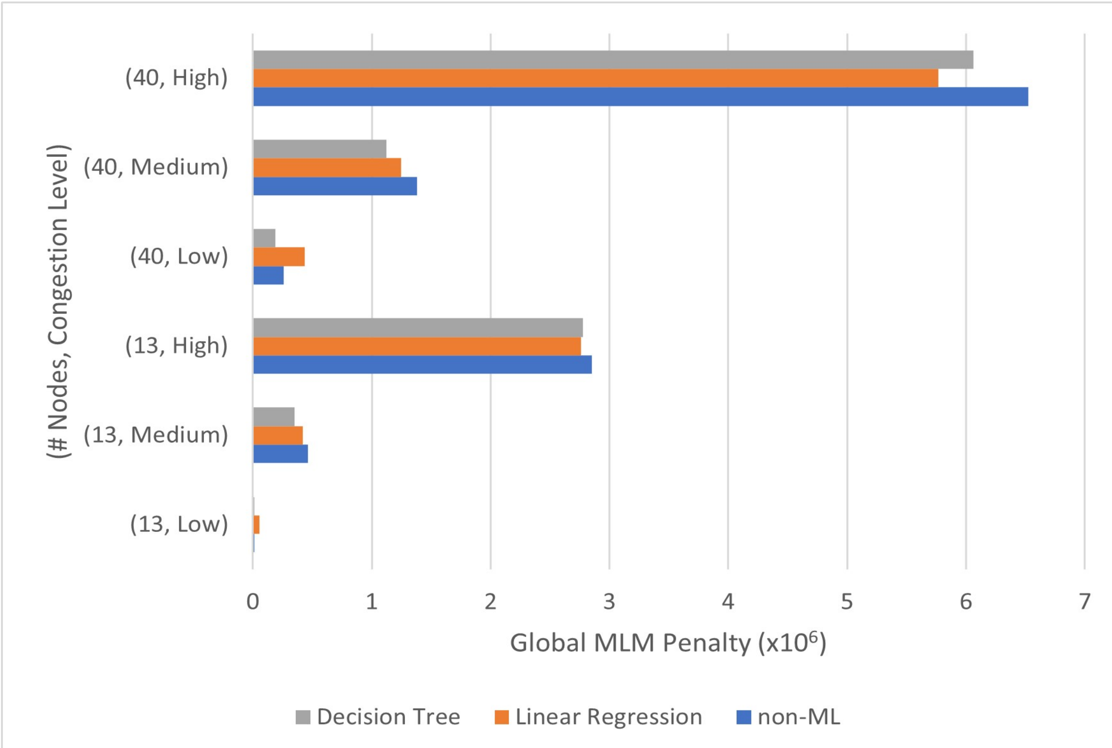

Figure 2: Global MLM penalty comparison between non-ML and ML approaches.

For calculating the amount of improvement of both methods from the baseline as shown in Table 5, we have used equation (2),

$$\left(1 - \frac{MLM_{Regression}}{MLM_{Baseline}}\right) * 100 \tag{2}$$

where $MLM_{Regression}$ is the global MLM penalty (as discussed in Section 3.2) by Linear Regression or Decision Tree methods and $MLM_{Baseline}$ is the global MLM penalty by the non-ML approach.

These results demonstrate the effectiveness of our models by accounting for both the features in the message and the operational context. There are two reasons why. First of all, the gateway nodes (that belong to two networks) have the delicate balancing act of prioritizing the Friendly Node Position, Enemy Detection and SOS messages that are to be forwarded with their own generated Friendly Node Position, Enemy Detection, Text and Tactical Graphic messages. The gateway nodes effectively maintained that balance in all scenarios except for those involving the low-congested 13-node networks as these nodes were found to give lower priority towards their own messages. Secondly, if a node's queue contains multiple Friendly Node Position and Enemy Detection messages from the same sender, the messages with the highest cost per message size ratio will be sent over the network, while the other messages will be set to expire.

In terms of which ML method achieved greater improvement over the non-ML approach, we have found that the Linear Regression method achieved greater improvement in the high-congested scenarios while the Decision Tree Method improved the most across the low to medium congested scenarios. Since Friendly Node Positions are being generated every second, it is crucial to send enough updates about a Friendly Node's position while also ensuring the timely delivery of all other message types. Figure 3 is a comparison of the average distance (in metres) a friendly node has travelled before it sends an update of its current position across the network. This shows that the Linear Regression method has the highest average distance for all scenarios. As a consequence, this method suffers a greater distance error penalty in the low to medium congested scenarios for not sending enough updates. But delaying the updates was

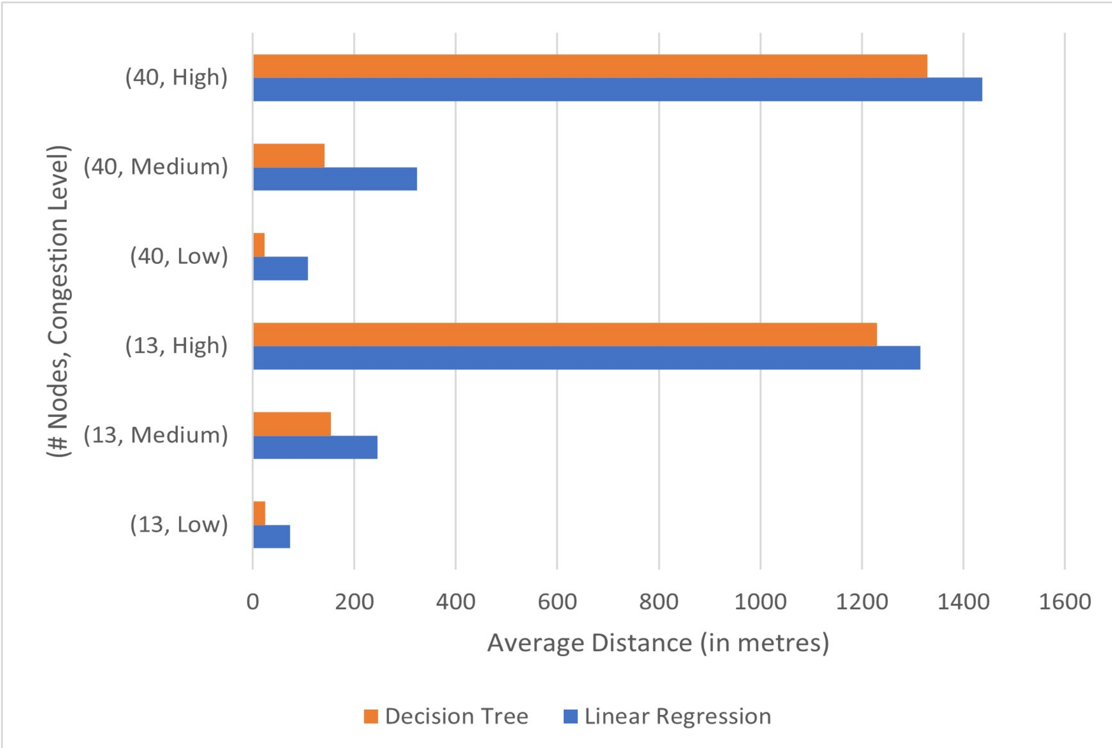

Figure 3: Average distance travelled in metres before friendly node sends current position.

proven to be beneficial in the high congested scenarios, which led to a greater improvement within those scenarios over the Decision Tree methods. A key reason why Decision Tree methods have lower average distance in all scenarios and a greater improvement over the baseline in both low and medium congested scenarios is mainly its ability to compute accurate predictions (by minimizing the error between actual and predicted values) across all message and context models. These results were omitted for brevity.

6 CONCLUSION

In this paper we have explored regression-based supervised-learning for understanding context for message prioritization. We have successfully demonstrated that linear regression (Bagula et al. 2016) and decision trees (Renners, Heine, and Rodosek 2017) can improve message prioritization when compared to a fixed ordering based prioritization approach. Critically, these methods have been shown to be particularly effective in medium and high congestion scenarios, where prioritization is expected to have the most value.

In this study we have explored only the mission context. The network context (i.e data capacity, actual packet send times, etc) is also a key component for effective message dissemination and prioritization. Future work should incorporate both network and mission context for message prioritization. This should support the learning of a compromise between overall message latency and position error. We expect that this inclusion will result in further improvements in network performance, reflected in MLM scores.

REFERENCES

Alberts, D., T. Bailey, P. Choinard, A. Tolk, G. Wheatley, and J. Wilder. 2002. *NATO Code of Best Practice for C2 Assessment*. Washington, DC: CCRP Press.

Bagula, A., C. Lubamba, M. Mandava, H. Bagula, M. Zennaro, and E. Pietrosemoli. 2016. "Cloud Based Patient Prioritization as Service in Public Health Care". In *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, 1–8. Manhattan, New York: Institute of Electrical and Electronics Engineers.

Baroutsi, N. 2015. “Measures of Command & Control Performance”. In *Proceedings of the 20th International Command and Control Research and Technology Symposium: C2, Cyber, and Trust*, Volume 21. June 16th-19th, Annapolis, Maryland, USA.

Beautement, P., D. Allsopp, M. Greaves, S. Goldsmith, S. Spires, S. G. Thompson, and H. Janicke. 2005. “Autonomous Agents and Multi–Agent Systems (AAMAS) for the Military–Issues and Challenges”. In *International Workshop on Defence Applications of Multi-Agent Systems*, 1–13. Springer.

Buitinck, L., G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. 2013. “API Design for Machine Learning Software: Experiences from the Scikit-Learn Project”. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Fujino, A., H. Isozaki, and J. Suzuki. 2008. “Multi-label Text Categorization with Model Combination Based on F1-score Maximization”. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*. January 7th-12th, Hyderabad, India.

Ho, T. K. 1995. “Random Decision Forests”. In *Proceedings of 3rd international conference on document analysis and recognition*, Volume 1, 278–282. Manhattan, New York: Institute of Electrical and Electronics Engineers.

Judd, G., R. Lorke, P. Boyd, V. Radenovic, and K. Chan. 2018. “Upping the IQ of Army’s Digital Communications”. In *International Conference on Science and Innovation for Land Power*. September 5th-6th, Adelaide, South Australia, Australia.

Judd, G. B., C. Szabo, K. S. Chan, V. Radenovic, P. Boyd, K. Marcus, and D. Ward. 2019. “Representing and Reasoning over Military Context Information in Complex Multi Domain Battlespaces using Artificial Intelligence and Machine Learning”. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, Volume 11006, 1100607. International Society for Optics and Photonics.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. “Scikit-Learn: Machine Learning in Python”. *Journal of Machine Learning Research* 12:2825–2830.

Renners, L., F. Heine, and G. D. Rodosek. 2017. “Modeling and Learning Incident Prioritization”. In *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Volume 1, 398–403. Manhattan, New York: Institute of Electrical and Electronics Engineers.

Schaefer, K. E., J. Oh, D. Aksaray, and D. Barber. 2019. “Integrating Context into Artificial Intelligence: Research from the Robotics Collaborative Technology Alliance”. *AI Magazine* 40(3):28–40.

Smith, P. F., S. Ganesh, and P. Liu. 2013. “A Comparison of Random Forest Regression and Multiple Linear Regression for Prediction in Neuroscience”. *Journal of neuroscience methods* 220(1):85–91.

Szabo, C., V. Radenovic, G. Judd, D. Craggs, K. L. Lee, X. Chen, and K. Chan. 2020. “Optimizing Communication Strategies in Contested and Dynamic Environments”. In *2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS)*, 197–205. Manhattan, New York: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

ROHIT GOPALAN is a Research Scientist in Land Division at DSTG. He holds a bachelors degree (with honours) in Software Engineering from the University of Western Australia. His research interests are in multi-agent reinforcement learning, computer vision and supervised learning. His e-mail address is rohit.gopalan@defence.gov.au.

HEDAYETUL SHOVON is a Research Scientist in Land Division at DSTG. He holds a PhD in Computer and Information Science from the University of South Australia in 2016. Previously, he was a Research Associate at the University of South Australia and was a lecturer in Rajahshi University of Technology. His research interests are on Machine learning, Deep Reinforcement Learning, Computer Vision and Artificial Intelligence. His email address is md.shovon@defence.gov.au.

BENJAMIN CAMPBELL is the Discipline Leader autonomous C5ISREW in Land Division at DSTG and is the DSTG technical lead for the Semantically Managed and Resilient Tactical Networking (SMARTNet) research activity. He has research interests in distributed control of land vehicle mission systems, distributed autonomy and RF signature management. His email address is benjamin.campbell5@defence.gov.au.

VANJA RADENOVIC is a Software Chapter Lead at Consilium Technology and has over a decade of experience in defence research and software engineering projects. His research interests are in command and control (C2) systems and tactical networking. His email address is vanja.radenovic@consilium.technology.

KYM McLEOD is a Senior Solutions Architect and Senior Software Engineer at Consilium Technology. Kym has more than 20 years experience in the Information Technology industry, and has worked in Commercial, Government and Defence sectors. He has a strong focus on System Architecture and Design with an extensive understanding and appreciation of good systems development life-cycle practices. His email address is kym.mcleod@consilium.technology.

LEITH CAMPBELL is a Software Engineer at Consilium Technology. With his Bachelor degree in Computer Science (AI) from the University of Adelaide, he has interests in streamlining and automating processes for the benefit of people. This includes working with basic and complex systems across a range of architectures and technologies. His email address is leith.campbell@consilium.technology.

DUSTIN CRAGGS is a researcher and software developer in the Complex Systems Research Group at the University of Adelaide. His email address is dustin.craggs@adelaide.edu.au.

CLAUDIA SZABO is an Associate Professor in the School of Computer Science and the Lead of the Complex Systems Research Group at the University of Adelaide. Her research interests lie in the area of complex systems and on using simulation to identify and validate their emergent properties. Her email address is claudia.szabo@adelaide.edu.au.