

Distributed Decision Contextualization via Machine Learning-based Reverse Parametrization

Stanley D. Hicks

Universities Space Research Association, Washington, DC, 20024, USA

Aditya Das

NASA Ames Research Center, Moffett Field, CA, 95035 USA

Husni Idris

NASA Ames Research Center, Moffett Field, CA, 95035 USA

Abstract – This paper presents a smart-machine-based decision-support framework that facilitates contextualization of individual decisions in multi-agent operations in shared airspace usage. In specific, the presented research aims to enhance the strategic decision-making from the different smart vehicles and operators looking to use shared resources in the federated airspace management architecture. Typical challenges addressed by the presented work are the high demand for decisions in a distributed and data-centric operational environment, and the information sparsity therein due to the lack of data and intent sharing among the competitive business operators. The presented research utilizes artificial intelligence and machine learning based reverse parametrization of the decision-making factors, and, thereby, offering a viable pathway towards holistic contextualization of operational intents and conflict resolution through inter-vehicle or inter-operator negotiation for resource sharing. This paper discusses the feasibility study and the utilization of this reverse parametrization approach through a cargo delivery scenario with multiple service providers and different cargo priorities. The preliminary results presented in the paper demonstrate the comparison of different machine learning models in predicting the undisclosed parameters and their parametric weights, both for cases with and without correlations between the disclosed and undisclosed parameters. The initial findings are encouraging as they show very good conformity between the predicted and actual parameters and preferences.

I. Introduction

In the Concept of Operations version 2.0 for Unmanned Aircraft Systems (UAS) Traffic Management (UTM) released in August 2022 [1], the Federal Aviation Administration (FAA) discusses a federated airspace management structure for UASs in the future, supporting the smart vehicles and their operators to conduct multifarious business operations, such as passenger transportation, cargo delivery, emergency response, etc., autonomously while sharing the airspace with other users. Many of these operations are on-demand, short-range/duration, and commercially competitive in nature with multiple business operators. The rapidly changing state of the airspace due to such unscheduled operations and the uncertainties around operational intents can lead to frequent conflicts in accessing shared resources such as airspace corridors, delivery zones, landing sites, electric charging points, etc. Resolving such conflicts in a decentralized manner remains suboptimal in the absence of information about decision-making parameters for all concerning vehicles and operators, values, and preferences, for many of such parameters are not shared openly by these entities.

Nevertheless, distributed decision-making in autonomous multi-agent systems is critical for enabling scalable operations, whereby each vehicle and/or vehicle operator coordinates with other vehicles and/or vehicle operators for its independent decision-making rather than depending on centralized coordination services for the usages of shared resources. The success of this self-operation and collaborative resolution of common operational conflicts in multi-agent systems relies on the completeness and timely availability of the information about each agent's situational awareness, operational intent, and business constraints. In real-world implementations, however, such holistic access to information is hindered by barriers such as limited communication bandwidth and non-cooperative behavior due to

lack of equipage or privacy concerns, respectively. To mitigate such barriers, each autonomous agent may augment its sensing and perception with learning and predicting the behavior of other agents through precedence and preference modeling. Such learning can enable anticipation of the intent of other agents, which helps effective negotiation and coordination for resource sharing. Each agent's decision-making model, driven by its business preferences, can consist of various parameters and corresponding parametric weights. Artificial intelligence and machine learning (AIML) methods can be utilized to build generalizable models of the other agents' decision-making that is based on their underlying business objectives and utility functions. Such a method is envisioned to achieve rapid convergence in multi-agent negotiation – a use case discussed in this paper in the context of a cargo delivery scenario.

This paper reports the findings from our ongoing research on a functional reverse parametrization approach that utilizes machine learning techniques to deliver rapid estimation of a specific agent's preferences with respect to a set of common aviation parameters. The individual preferences, or the parametric weights, towards the final decision are assumed to be not known, because these are decided based on how the business chooses to run its operations, and thus, can vary from business to business. Values for a subset of the parameter set are known, as the vehicle and/or vehicle operator is required to provide this information as per regulatory requirements. For example: current position, heading, fuel status, payload type, etc. are considered as “*Public*” parameters, values for which are shared by the vehicle and/or vehicle operator. On the other hand, values for the remainder of the parameter set are unknown and thus considered as “*Private*” parameters. For example: information on price paid for fuel, preference for on-time arrival, preference for passenger comfort, etc. are not shared in a public domain. Regardless, both *public* and *private* parameters contribute to the final decision, and, therefore, it is imperative to gather understanding of the driving functions to interpret the context for the decision.

This paper is organized as follows: section II discusses the background including our past work in the presented research area. Section III describes the cargo delivery use case. Section IV presents the reverse parametrization approach, followed by experimental findings summarized in section V. Finally, section VI concludes the paper with information on future directions.

II. Background

Decentralized decision-making is the process where the decision-making authority is distributed among individual entities in a group. Numerous studies have been conducted to implement the decentralized decision-making in multiple contexts including cooperative robots and sensors [2], auction-based task allocation [3], negotiation platform for multi-agent task allocation [4], swarm intelligence for multi unmanned aerial vehicles (UAVs) [5], surveillance and monitoring applications [6] and many more [7]. Multi-agent systems suffer from computational complexity, partial observability, and reliance on incentivization for task completion. As discussed in [8], opponent modeling and inter-agent communication are some of the most common multi-agent challenges. Contemporary approaches do not take into consideration the opponent's level of intelligence and/or assume access to the opponent's parameters, which is unlikely in real-world business settings. While the opponent modeling and communication can reduce the non-stationary behavior and improve the observability, real-world scenarios rarely benefit from these due to reluctance to share information. Several research studies in this direction can be found in literature that present spatial action mapping [9] and spatial intent mapping [10] [11].

In the previous reporting [12] of our research, we introduced the general governing dynamics of the distributed decision-making, in the context of machine-to-machine negotiation in shared airspace, as shown in the Eq. (1).

$${}^rU_k = \left[\sum_{i=1}^m ({}^rP_i \cdot {}^rW_{ki})_{norm} + \sum_{j=1}^n ({}^rQ_j \cdot {}^rW_{kj})_{norm} \right] - \left[\sum_{s=1}^N \mu_{rs} \cdot \left(\sum_{i=1}^m {}^sP_i \cdot ({}^sW_{ki} + \Delta {}^sW_{ki}) + {}^s\phi_{kj} \right) \right] - {}^gU_k + {}^tU_k \quad (1)$$

Where:

$U_{1 \text{ to } l}$: Strategy utilization cost corresponding to the decision taken by a smart (autonomous) agent

$P_{1 \text{ to } m}$: Public parameter values that contribute to the decision-making

$Q_{1 \text{ to } n}$: Private parameter values that contribute to the decision-making

W_{ki} : Preference (or parametric weights) towards the public parameter P_i for strategy or decision S_j

W_{kj} : Preference (or parametric weights) towards the private parameter Q_j for strategy or decision S_j

r : Smart agent identifier (1 to N)

The expression inside the first square bracket on the right-hand side of Eq. (1) represents the agent's utilization cost based on its own private and public parameters. The expression inside the second square bracket represents the agent's estimation of the other agents' utilization cost. The term ' μ_{rs} ' is non-zero for all other agents impacted by the decision and zero for all other agents not impacted by the decision. The term ' ϕ_{kj} ' represents the uncertainty associated with

such an estimation by one agent about other agents' utilization costs, as for those agents the weights ' W 's and private parameter values ' Q 's are not known. The private parameters and preferences may be known or not known due to enforcement of aviation authorities, or due to operators not wanting to reveal strategies that are business essential. The term ' gU_k ' represents global reward for good decisions. Lastly, the term ' U_k ' represents the time penalty for delaying the decision. So as time progresses, this penalty increases gradually. In summary, we hypothesize that overall better decisions depend on better estimates about other impacted agents' decision factors. This constitutes the basis of the work presented in this paper, which is to predict the Q and W values.

To create a framework to predict these Q and W values, we created a synthetic dataset consisting of generated parameters and weights. The P and Q values representing the public and private parameters, respectively, are sampled from a normal distribution with a mean and standard deviation sampled from a uniform distribution.

$$P_i \sim N(\mu_i, \sigma_i), \quad \mu_i \sim U(\mu_{min}, \mu_{max}), \quad \sigma_i \sim U(\sigma_{min}, \sigma_{max}),$$

$$Q_j \sim N(\mu_j, \sigma_j), \quad \mu_j \sim U(\mu_{min}, \mu_{max}), \quad \sigma_j \sim U(\sigma_{min}, \sigma_{max}),$$

The W values representing the public and private parameters, respectively, are sampled from a uniform distribution between 1 and 0 to determine the utilization cost for strategy k , based on: $W_{ki}, W_{kj} \sim U(0,1)$.

From this synthetic dataset of parameters and weights, we can then calculate the public part of the utilization cost, U_k^{pub} , which is the inner product of the vector of public parameters and the vector of public weights. Since we assume that the total utilization cost, U_k , is known, we also assume that the private part of the utilization cost, U_k^{pri} , is known, since U_k is simply the sum of U_k^{pub} and U_k^{pri} . We can then concatenate each sample for a single negotiation of the public parameters, P_j , with the public weights, W_{ki} , the total utilization cost, U_k , the public cost, U_k^{pub} , and the private cost, U_k^{pri} . This establishes a single input sample for strategy k . Since we want to predict the private parameters, Q_j , and the private weights, W_{kj} , we create a single output sample through the concatenation of each sample for a single negotiation of the public parameters, P_j , and the private weights, W_{kj} . This similarly establishes a single output sample for strategy k .

III. Implementation Scenario

To evaluate the utilization of the reverse parametrization for distributed decision-making, we picked a cargo delivery scenario as depicted in Fig. 1.

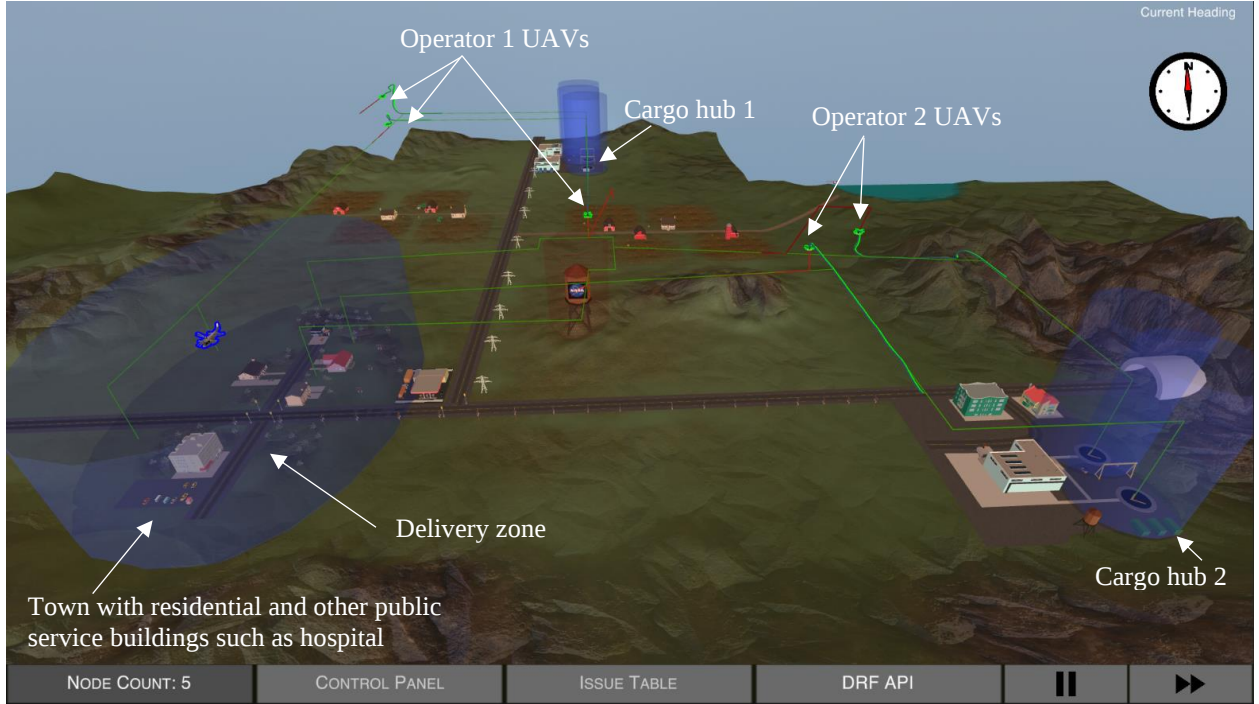


Fig. 1 Cargo delivery scenario for reverse parametrization

In this scenario, there are two cargo delivery operators with cargo hubs in two distinct parts of the town. Three UAVs from operator 1 and two UAVs from operator 2 are tasked to deliver packages to a residential zone that includes several houses and a medical clinic. The packages being delivered have different priority levels such as normal delivery, rushed order, and urgent medical supplies. Due to safety and noise concerns, only one UAV is allowed to occupy the delivery zone at any given time. Furthermore, the operators 1 and 2 are not aware of each other's UAV flight details. Publicly available information about the vehicles includes current positions (latitude, longitude, altitude, and timestamp), current speed and heading, remaining fuel/battery and flight time, and operator identifier. Aside from these public parameters there are several privately held information such as package type (heavy, standard, fragile, etc.), delivery type (normal, rushed, emergency, etc.), number of deliveries per sortie, fuel conservation preference, etc. Finally, for conflict resolution the vehicles are allowed to carry out one of the three strategies: (a) hold position (speed alteration), (b) reroute (position alteration), and (c) no changes (continue flying on the original course and flight plan). For the first two strategies the vehicle/operator receives a credit, and for the third strategy it pays a credit.

The entire concept of operation for the scenario is simulated in our custom developed advanced air mobility (AAM) simulator [13]. This simulator allows decentralized multi-agent operation where the agents are independently controlled by smart edge nodes [14]. The simulator also enables real-world environmental data access from NASA's Data and Reasoning Fabric (DRF) platform [15] to bring in realism to the simulation environment. DRF serves as an online marketplace for data and analytics, packaged as services that aviation stakeholders, such as flight operators, smart vehicles etc., can subscribe to and receive operational and environmental data and analytics to build situational awareness, share information, and receive alerts.

In the absence of any inter-vehicle/inter-operator coordination, as shown in Fig. 2, we observed a grid lock in accessing the delivery zone. When the vehicle that is first to reach the delivery zone, enters it, the maximum occupancy for the delivery zone is reached and any other vehicle is denied access to the delivery zone until the vehicle that is currently occupying the delivery zone exits the zone. However, at that point all the remaining vehicles waiting outside the delivery zone are ready to enter, and without any coordination they get into a race condition that prevents any of them from entering the delivery zone. Note that, given the short time window for delivering the cargo at the recipients site that does not include any parking spot (or even considering that the cargo could be dropped or tethered down from the hovering vehicle), it is assumed that the occupancy status for the delivery zone does not change until the air vehicle flies out of the zone. So, the case that would consider the vehicle exiting the zone when landed in the zone, is excluded.

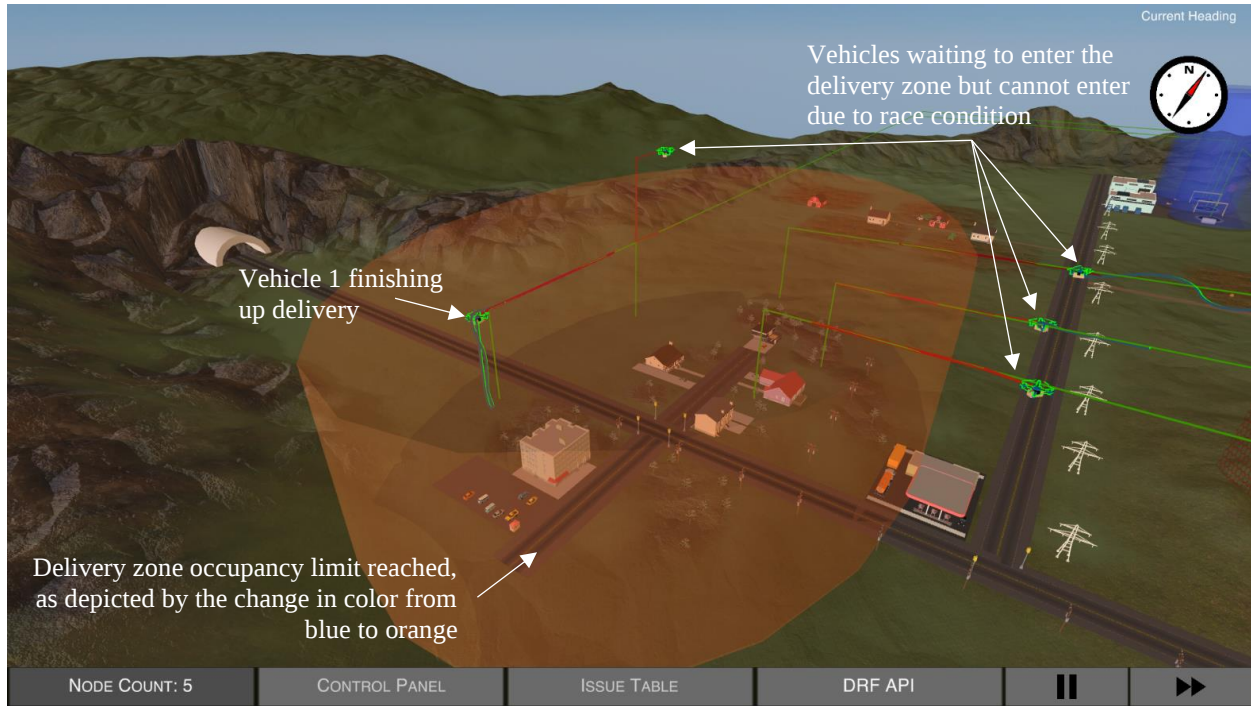


Fig. 2 Grid lock due to no coordination among vehicles and operators

In the next section, we discuss the reverse parametrization approach to enhance the distributed decision-making process for active negotiation-based conflict resolution.

IV. Reverse Parametrization Approach

The objective of the reverse parametrization approach is to determine the private parameters and corresponding preferential weights associated by the vehicle/operator, to expedite the convergence during negotiation. To evaluate machine learning approaches for the reverse parametrization due to their ease of use and prevalence, we chose the four following machine learning models for the multiple regression of the private parameters:

- i. *Decision tree* – A supervised machine learning algorithm using binary tree graph
- ii. *Linear Model* – A linear regression with ElasticNet regularization [16]
- iii. *MLP-NAS (MLP- Neural Architecture Search)* – A multilayer perceptron for regression that utilizes a neural architecture search for determining the number of layers and neurons in each layer
- iv. *XGBoost (Extreme Gradient Boosting)* – A scalable, distributed gradient-boosted decision tree (GBDT) machine learning library that provides parallel tree boosting [17]

The Decision tree and Linear Model algorithms were implemented using Scikit-Learn, XGBoost from the XGBoost Python library, and the MLP-NAS from a combination of TensorFlow, Keras, Ray-tune, and HyperOpt.

We then created a dataset to test these four machine learning models. As mentioned in Section III, in our cargo delivery scenario, three conflict resolution strategies are available to each agent/operator, and for each strategy, the agent/operator can have different preferences (or parametric weights) for the public and private parameters. To simulate an exhaustive scenario, we generated a dataset that consists of 100 different preferences to represent 100 different conflict avoidance strategies. Each of these preferences contains five private and five public parametric weights, each sampled from a uniform distribution between 0 and 1, as described in Section II.

Next, we generated the set of parameters for the evaluation of the reverse parameterization. The private parameters are generated using a normal distribution with random mean and variance, as described in Section II, and 5000 values are sampled from that distribution. To have a private parameter linked to each private preference we randomly sample a vector with 5 values to act as a single private parameter sample. Fig. 3 shows a histogram for each of the five ground-truth private parameters.

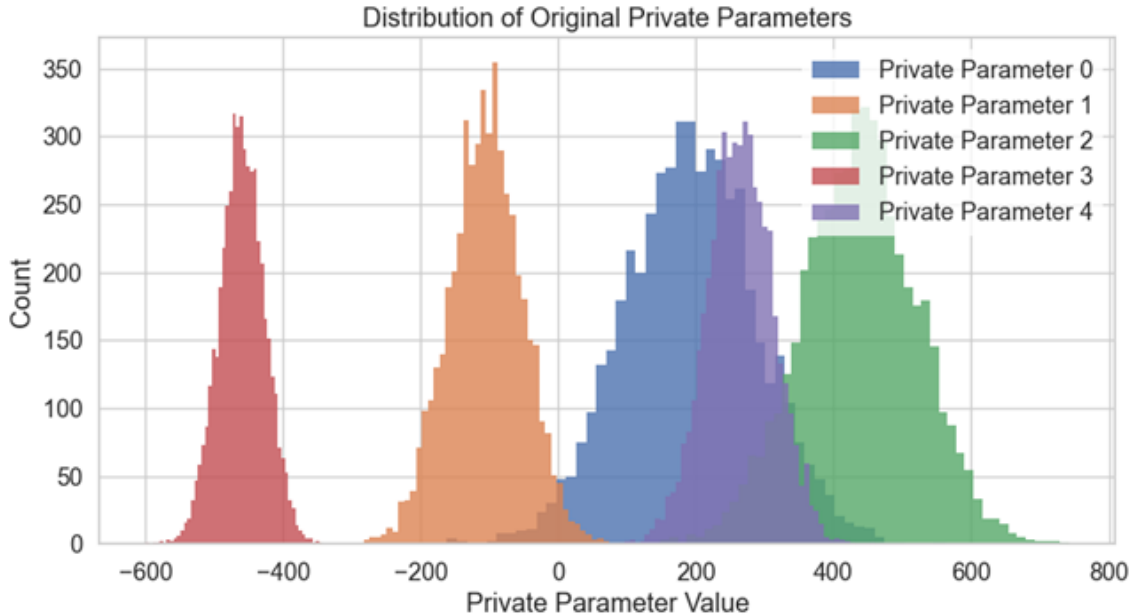


Fig. 3 Histogram of Original Private Parameter Distribution

Next, to generate the public parameters that correspond to each of the private parameters, we considered two cases where the private parameters are either uncorrelated or correlated to the public parameters. For the case where the private parameters are *uncorrelated*, public parameters were generated in the same way as private parameters, sampling from a normal distribution with a new random mean and variance. In the case where private parameters are *correlated*, we created public parameters by applying a linear equation to each sample i in the private parameter set, where m and b , respectively, are a random slope and intercept, and ϵ is noise sampled from a normal distribution, as highlighted in Fig. 4. This creates correlated public parameter P_i , where $P_i = (m * Q_i) + b + \epsilon$,

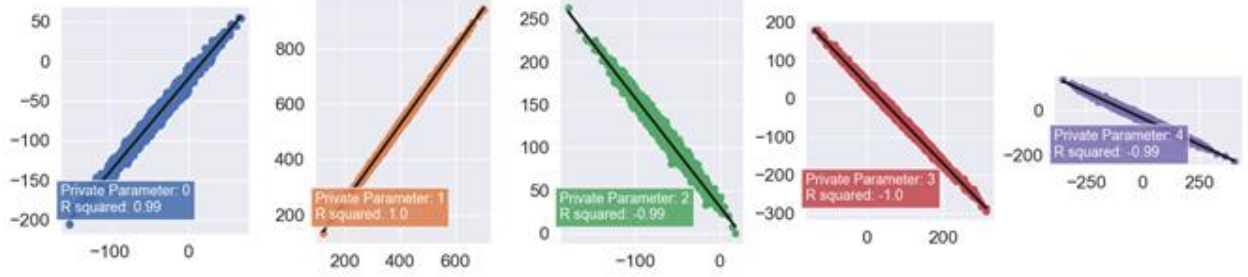


Fig. 4 Private parameter prediction (with correlation between public and private parameters)

Thus, for training the models, we used the public parameters, their private and public costs, and the total cost for each strategy as input features (or known values 'X'), and the private parameters and their preferential weights as regression values (or unknown values 'Y'), which creates a multiple regression problem. For the sake of simplicity, we assumed that the private parameters remain the same for all strategies and their preferential weights change for each strategy. For prediction 1,000 samples out of the 5000 input features were picked and their corresponding 'Y' values were predicted, which represents an 80/20 train/test split, with each 1000 samples being randomized for each strategy. When predicting the private parameters, we considered scenarios where there are certain assumptions of the input dataset were made as follows:

1. No correlation between private and public parameters, but the private/total cost is known
2. No correlation between private and public parameters, but the private/total cost is not known
3. Correlation between private and public parameters, but the private/total cost is known
4. Correlation between private and public parameters, but the private/total cost is not known

V. Summary of Experimental Findings

For each scenario highlighting different assumptions of the input dataset, we then measured the performance of the different model predictions by measuring the following metrics:

1. R value between predicted and actual Private Parameters (Q_i)
2. R value between predicted and actual Private Parameter Preferences (W_i)
3. R value between predicted and actual Private Cost ($Q_i * W_i$)
4. Distribution Overlap between the distribution of predicted and actual

A higher R-value, which can range from -1.0 to 1.0, implies that the model can better predict parameters and preferences for a single negotiation instance, leading to shorter negotiation time for conflict resolution. Similarly, a higher distribution overlap, defined as the integral between the actual normal distribution of private parameters and a normal distribution created from the sample mean and variance of the predicted private parameters, can aggregate reverse parameterization performance results over many negotiation instances. Similarly, a higher distribution overlap value can lead to both shorter negotiation times, and better precision of private parameter predictions for high-density UAS operations.

First, we tested the first scenario, where there is no correlation between private and public parameters, but the private cost is included in our input features, 'X'. We first measured the R value between predicted and actual Private parameters (Q_i). Fig. 5 illustrates the prediction of private parameters for a single, example strategy in the form of scatter plots and line of best fit, and each color represents one of the five private parameters used for this study. To aggregate results over all 100 strategies, we then calculated the R value between predicted and actual private parameters for all parameters in the form of a Monte Carlo simulation. This is shown in Fig. 6 using a combined box and scatter plot containing R values acquired from the Monte Carlo simulation.

In summary, the XGBoost and Decision Tree models could have moderate performance for a small number of strategies, but they tended to overtrain to the dataset heavily, resulting in low performance overall when measuring private parameters. On the other hand, both the MLP-NAS and Linear Model demonstrated better performance than the latter two models, showing a consistent positive trend, with the MLP-NAS model having the highest average R value. Even though some models can predict private parameters with higher performance compared to other models, the performance of private parameter prediction was still highly variable for all models, with low R values, indicating that other scenarios with different assumptions of data may offer better reverse parameterization.

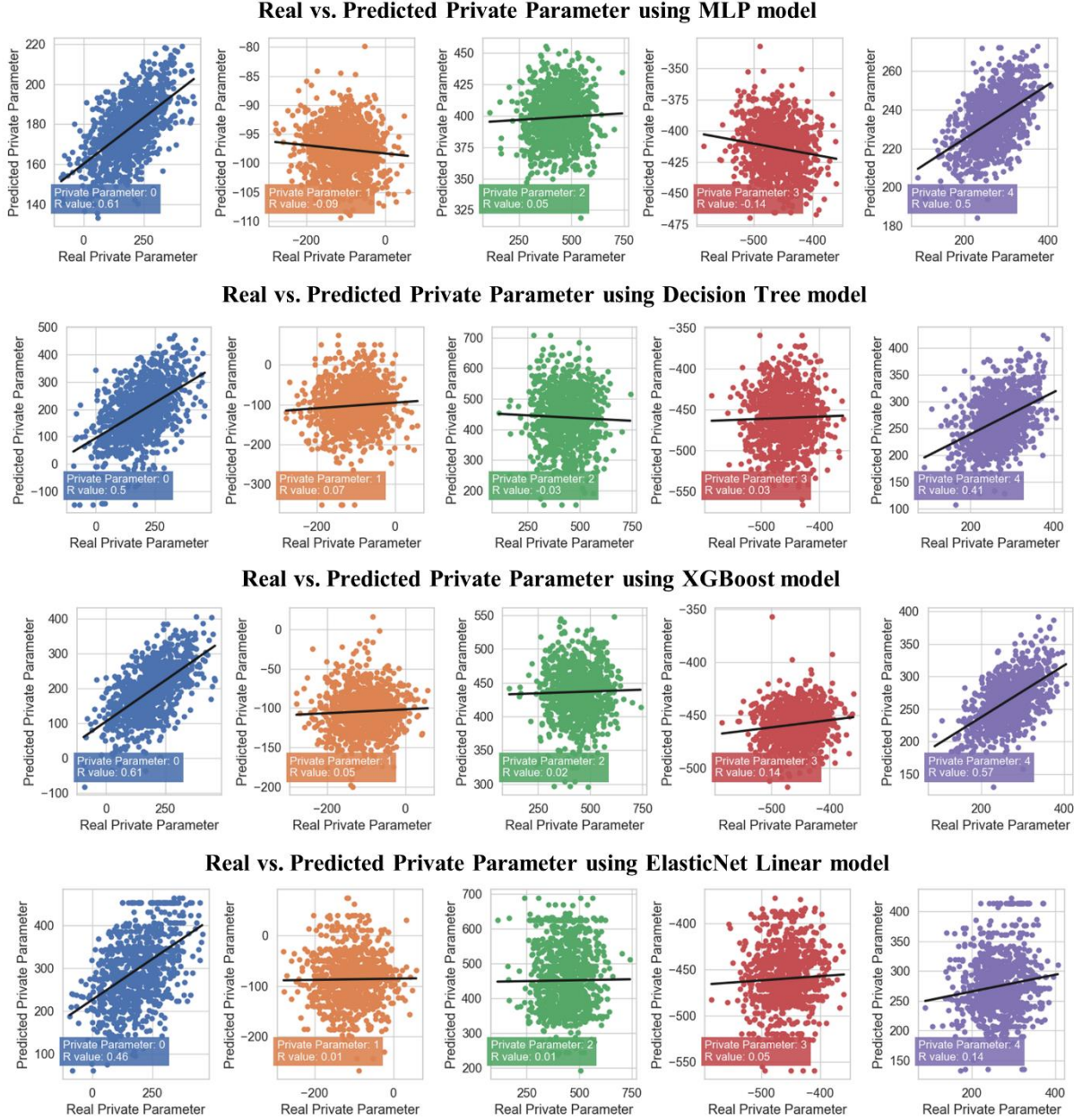


Fig. 5 Private parameter prediction (with correlation between public and private parameters)

Next, by taking the inner product of the predicted parameter values and preferences (weights), we calculated the predicted private costs, U_k^{Pri} , which is given as $[Q_j * W_j]$. Example scatter plots of each model's predicted private costs for a single strategy are shown in Fig. 7. Overall, MLP-NAS and Decision Tree show a consistent and positive trend when predicting U_k^{Pri} , but the costs are scaled incorrectly. On the other hand, XGBoost, and Linear Model show a strong, positive correlation when predicting the U_k^{Pri} , as shown in Fig. 8. This same performance is shown over aggregated results with the Monte Carlo simulation, with all models having high R values, but with MLP-NAS having a large variability in R values when compared to all other models.

We also measured the R values for the private preferences predicted from each of the 100 strategies, as illustrated in Fig. 8, but each model had very high performance, predicting private preferences almost perfectly. This can be attributed to the fact that over each sample in 'X', the private preferences remain the same, and thus each model can easily optimize itself to predict this single value over many samples.

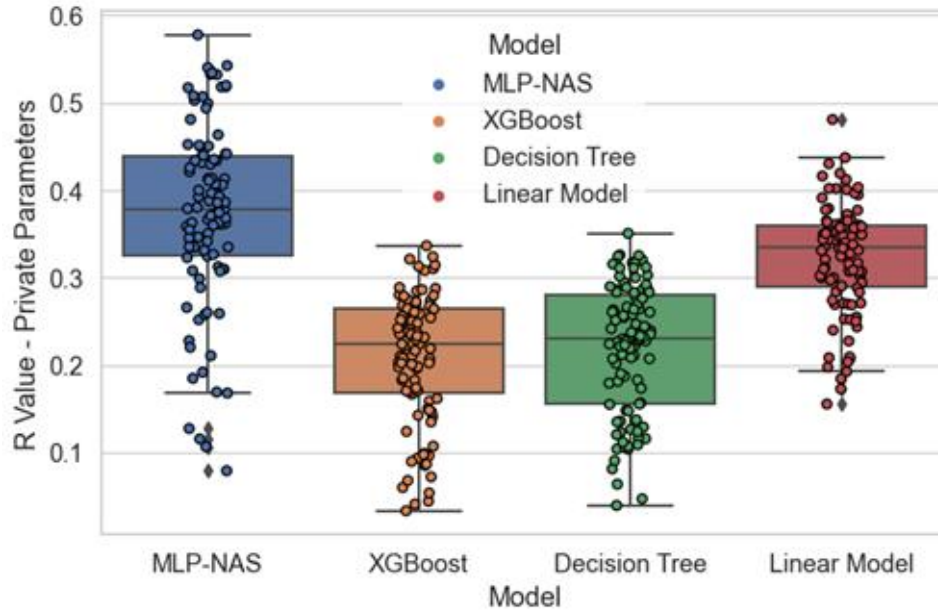


Fig. 6 Box-and-Scatter Plots of Real Private Parameters versus Predicted Private Parameters

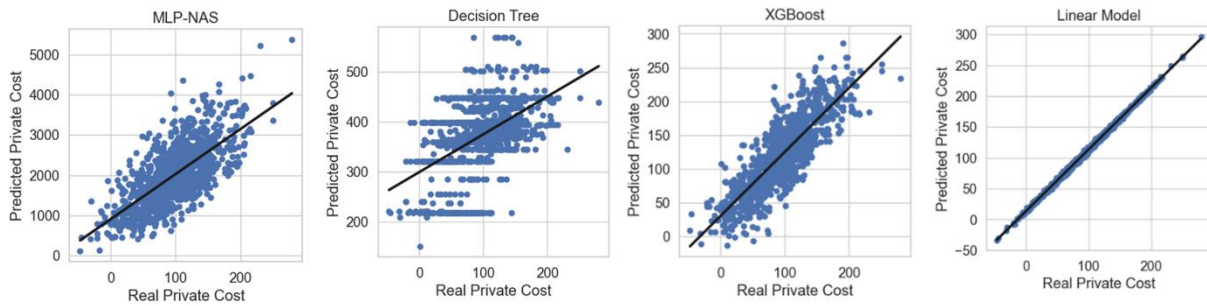


Fig. 7 Scatter Plots of Real Private Costs versus Predicted Private Costs

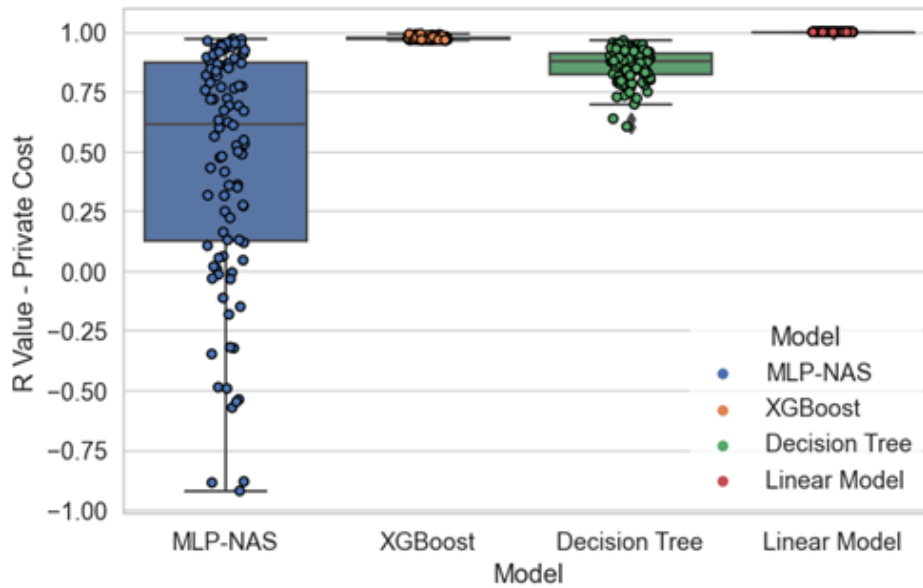
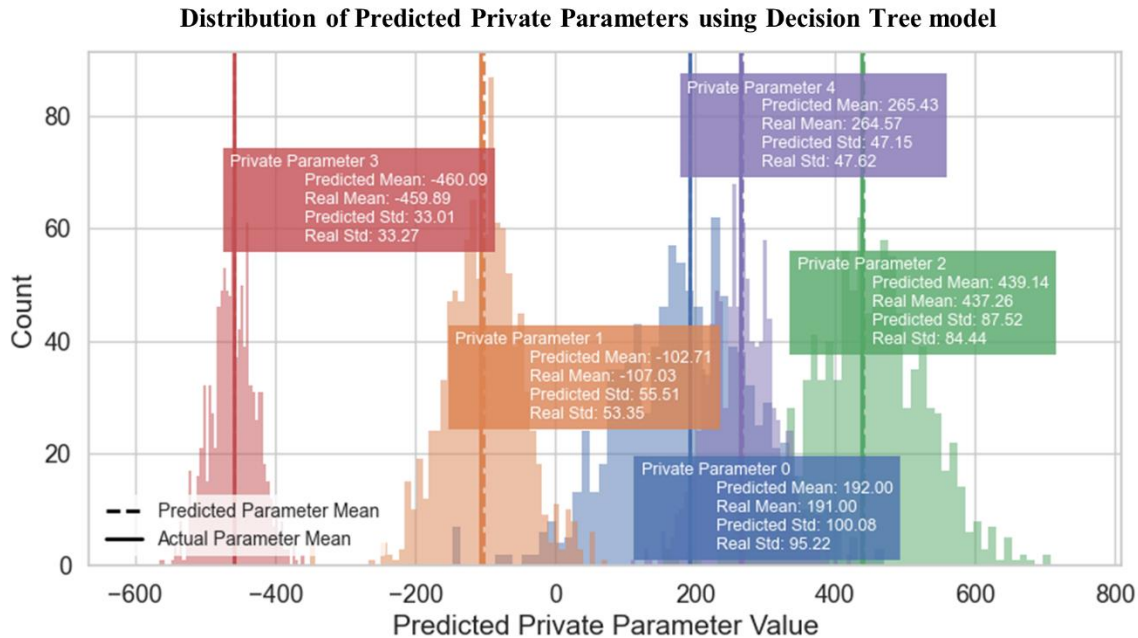
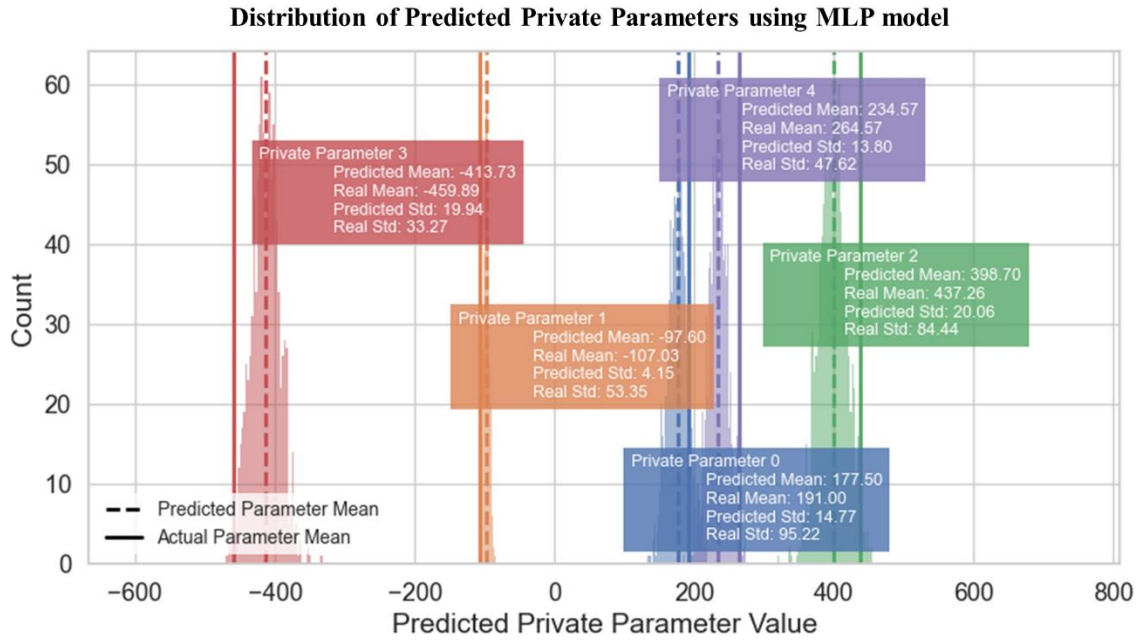


Fig. 8 Box-and-Scatter plot of Private Cost Prediction R values over 100 strategies (without Correlation)

Lastly, we assess how well each model predicted the distributions of private parameters. The plots in Fig. 9 show example distributions of the predicted private parameters, Q_j , using the testing set. We then aggregated these results over 100 strategies using a Monte Carlo simulation, as shown in Fig. 10. Qualitatively, each model performs well when predicting the mean of the Q_j distributions, with varying levels of success when matching the standard deviation of the distributions. Decision trees remain close in standard deviation to the original distribution, while the MLP and XGBoost models tend to overfit to the mean, having tighter distributions than the original and Decision Tree-predicted distributions. The ElasticNet linear regression remains close in mean and standard deviation but has a much less evenly distributed distribution. However, when looking at the Monte Carlo simulation as illustrated in Fig. 10, we see that the tree models, XGBoost and Decision Tree, perform very well at predicting the distributions of private parameters. Notably, although XGBoost-predicted distributions do overfit to the mean, these tighter distributions still can be closer to the distributions generated by other models. MLP-NAS and the Linear Model similarly can perform well, but can on average, have lower distribution overlap values than the former two models.



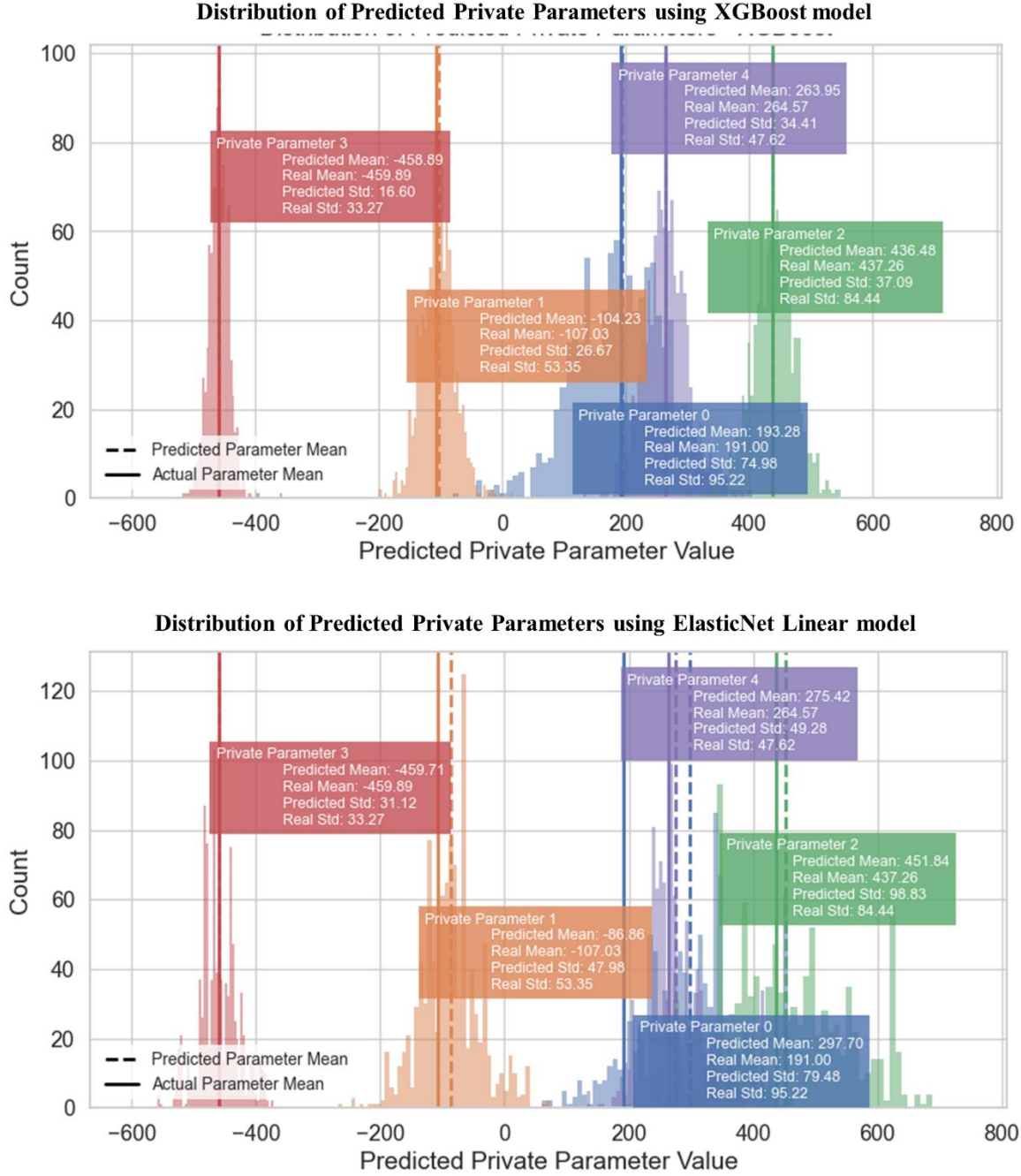


Fig. 9 Histograms of Predicted Private Parameter Distributions for the Models

Overall, all metrics except for the private parameter (Q_j) prediction, R values are strong, with uncorrelated parameters and private cost $[Q_j * W_j]$ included, as shown in Fig. 11. However, model performance shifts drastically when doing an ablation study with the total and private cost. After repeating the same experiments on the scenario where parameters are still uncorrelated, but costs are *removed*, performance reverse overall is much lower. This can be illustrated in Fig. 12, which summarizes the performance of each model when costs are removed. While private preference (W_j) prediction performance is still strong, all other metrics are exceptionally low, indicating that the inclusion of private costs positively influences reverse parameterization performance. This may be because, with private costs removed, there remains no data that could aid in the prediction of private parameters. This is especially apparent with the prediction of the private parameters, as without any cost information, parameters cannot be inferred from a known preference and cost value.

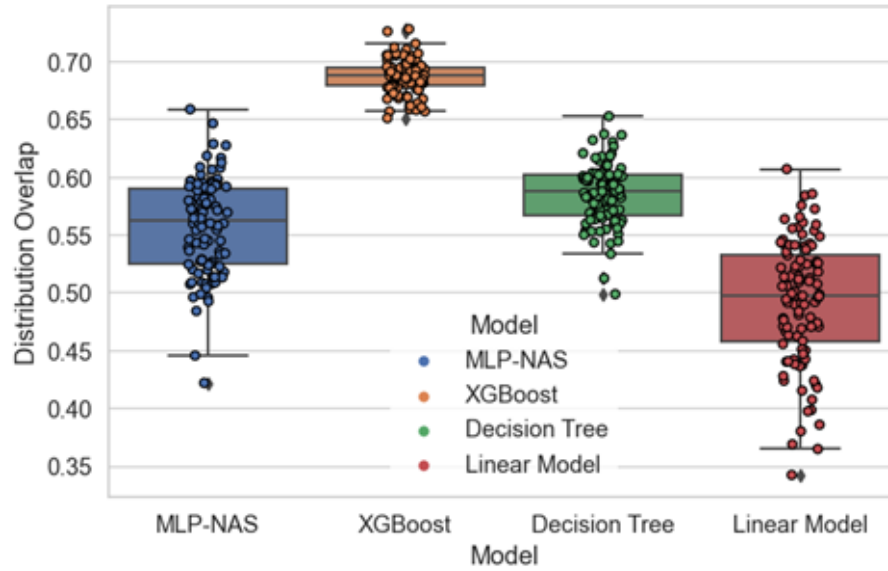


Fig. 10 Box-and-Scatter plot for Distribution Overlap for 100 strategies (without Correlation)

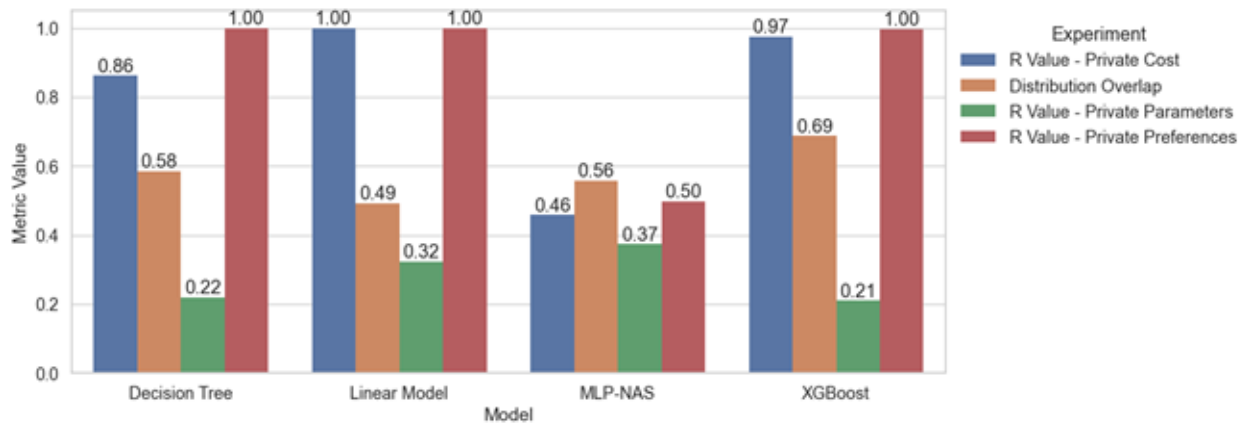


Fig. 11 Bar Plot Summarizing Performance of Reverse Parameterization with Noncorrelated Parameters and Private Costs Included

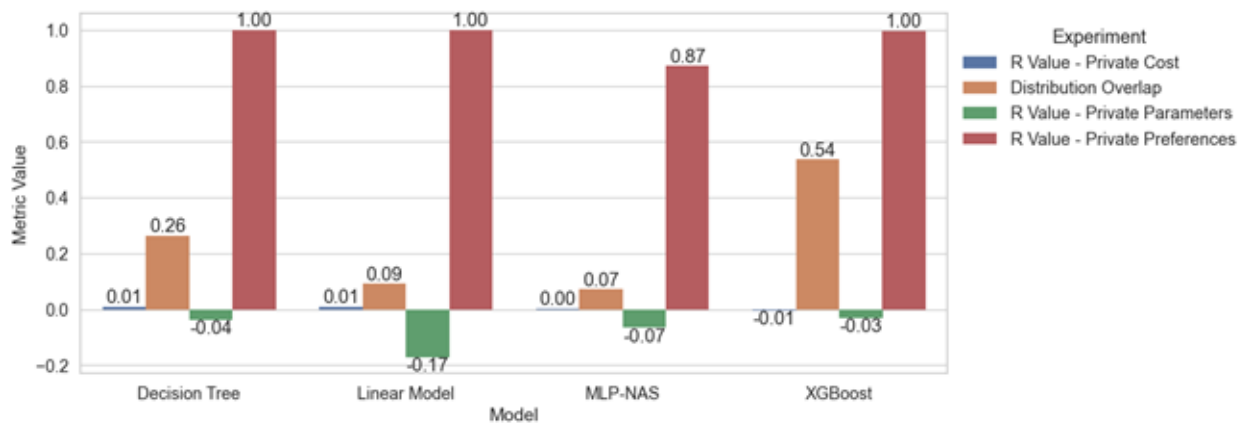


Fig. 12 Bar Plot Summarizing Performance of Reverse Parameterization with Noncorrelated Parameters and Private Costs Removed

On the other hand, this scenario where there is no relationship between private and public parameters and costs may be unrealistic, so we then tested our two remaining scenarios where we assume correlation between private and public variables, as illustrated in Fig. 13 and Fig. 14. With correlated private preferences, all models achieved a higher overall reverse parameterization performance. The performance of the Decision Tree and MLP-NAS models only slightly improved, due to the poor inherent abilities of MLP and Decision Tree models to predict linear relationships. On the other hand, the XGBoost and Linear Models, which excel at predicting linear relationships, had very high reverse parametrization performance over every metric, with R values close to 1, and distribution overlap. From this, we can deduce the correlation of private and public parameters can have an even more positive effect on reverse parameterization than the inclusion of private cost information. With linearly correlated parameters, models that can approximate linear relations can directly infer private parameters, leading to exceptional performance for private cost and parameter prediction. This means that with linearly correlated private and public parameters, even lower conflict negotiation times can be achieved.

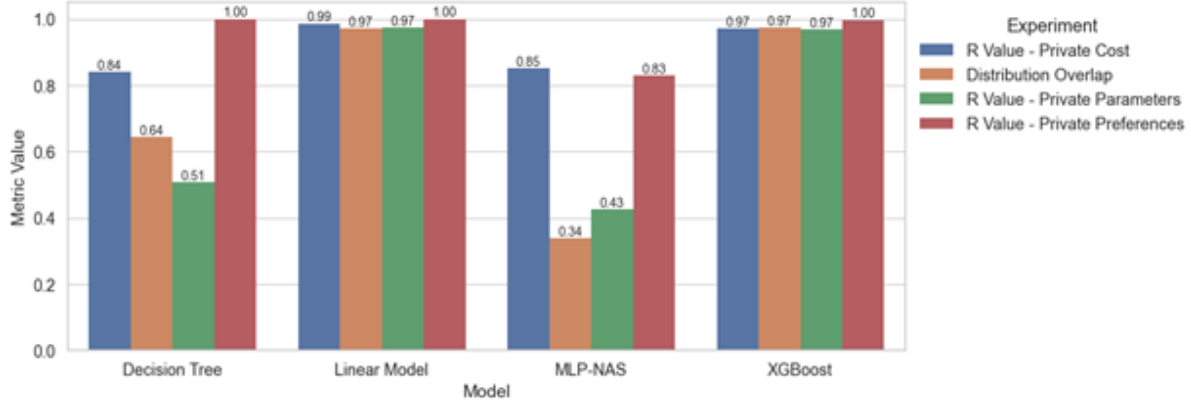


Fig. 13 Bar Plot Summarizing Performance of Reverse Parameterization with Correlated Parameters and Private Costs Removed

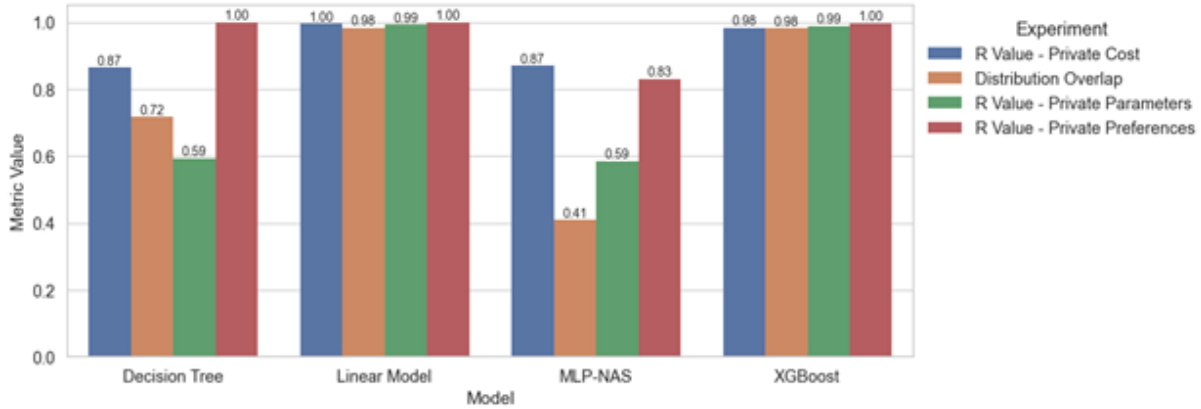


Fig. 14 Bar Plot Summarizing Performance of Reverse Parameterization with Correlated Parameters and Private Costs Included

In summary, we observed that the estimation of the private parameter distribution is significantly aided by the information on private and public parameter correlation. It is also noteworthy that the correlation information has a stronger impact on the model accuracy and convergence speed in comparison to the total private cost (sum of private parameter * preferential weight) information. In a real-world implementation, the correlation information is easy to establish using physical or business models; for example: correlations between (a) temperature and battery performance, (b) headwind speed and engine throttle, (c) crude oil price and profit per flight, (d) number of available UAVs and wait time, (e) length of the route and passenger comfort, and so on. Such correlations help the prediction of private parameter costs, thereby offering a more accurate understanding of the vehicle/operator intent. Lastly, the model choice can have a large effect on model performance, depending on the assumption of data: for all cases, XGBoost had the highest reverse parameterization performance, but a Linear Model could have high performance if private and public parameters are assumed to have a linear relationship.

VI. Conclusion

This paper presented an artificial intelligence (AI) approach for intent prediction in multi-agent operations in shared airspace. The presented approach evaluated several models with unrelated as well as correlated data to reverse-parametrize private parameters. This study is envisioned to enhance distributed decision-making by the shared airspace users and improve the overall convergence rate in multi-agent negotiations for resource sharing and conflict resolution.

In the future, our research will look to expand the Monte Carlo simulations to include different types of noise for correlated private parameters and different kinds of nonlinear correlation. Additionally, this research will work towards fine-tuning the prediction models, and implementing AutoML-based methods for predictions, duplicating the training processes of many modern cloud-based prediction services. The results presented in this paper exhibit strong correlation/conformance between the actual parametric distributions and corresponding AI-estimated reverse parametrized distributions. With more training data, especially from real-world parameters, we expect the machine learning (ML) approach to offer a more accurate and faster outcome, which will be a major part of the future work.

References

- [1] Federal Aviation Administration (FAA) NextGEN, "Concept of Operations v2.0, Unmanned Aircraft Systems (UAS) Traffic Management (UTM)," FAA, Washington DC, 2022.
- [2] A. Khamis, A. Hussein and A. Elmogy, "Multi-robot Task Allocation: A Review of the State-of-the-Art," in *Cooperative Robots and Sensor Networks*, NY, USA, Springer International Publishing, 2015, pp. 31-51.
- [3] H.-L. Choi, L. Brunet and J. P. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912-926, 2009.
- [4] A. Leikna, E. Lavendelis and A. Grabovskis, "Experimental analysis of contract net protocol in multi-robot task allocation," *Applied Computer Systems*, vol. 13, no. 1, pp. 6-14, 2012.
- [5] J. Schwarzrock, I. Zacarias, A. L. Bazzan, R. Q. d. A. Fernandes, L. H. Moreira and E. P. d. Freitas, "Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 10-20, 2018.
- [6] N. Patrinooulou, I. Daramouskas, D. Meimetis and V. Lappas, "A Multi-Agent System Using Decentralized Decision-Making Techniques for Area Surveillance and Intruder Monitoring," *Drones*, vol. 6, no. 11, 2022.
- [7] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, vol. 55, pp. 895-943, 2021.
- [8] A. Wong, T. Bäck, A. V. Kononova and A. Plaat, "Deep multiagent reinforcement learning: challenges and directions," *Artificial Intelligence Review*, 2022.
- [9] Z. Wong and N. Papanikolopoulos, "Spatial Action Maps Augmented with Visit Frequency Maps for Exploration Tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, 2021.
- [10] J. Wu, X. Sun, A. Zeng, S. Song, S. Rusinkiewicz and T. Funkhouser, "Spatial Intention Maps for Multi-Agent Mobile Manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021.
- [11] S. Qi and S.-C. Zhu, "Intent-aware Multi-agent Reinforcement Learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018.
- [12] A. Das, K. Marotta and H. Idris, "Deep Learning-based Negotiation Strategy Selection for Cooperative Conflict Resolution in Urban Air Mobility," in *AIAA SciTech*, 2021.
- [13] A. N. Das and S. D. Hicks, "A 3D Simulation Platform for Decentralized Decision-Making in Advanced Air Mobility," in *AIAA Aviation*, Chicago, IL, 2022.
- [14] A. Das and S. D. Hicks, "Feasibility Study of Distributed Decision - Making on the Edge for Urban Air Mobility," in *AIAA SciTech Forum and Exposition*, National Harbor, MD, 2023.
- [15] W. R. Van Dalsem, S. D. Shetye, A. Das, K. Krishnakumar, S. Lozito, K. Freeman, A. J. Swank, P. Shannon and L. Tomljenovic, "A Data & Reasoning Fabric to Enable Advanced Air Mobility," in *AIAA SciTech Forum*, 2021.
- [16] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, 2005.
- [17] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016.