Enhancing Rover Teleoperation on the Moon With Proprioceptive Sensors and Machine Learning Techniques

Sofía Coloma, *Member, IEEE*, Carol Martinez, Barış Can Yalçın, and Miguel A. Olivares-Mendez

***Abstract*—Geological formations, environmental conditions, and soil mechanics frequently generate undesired effects on rovers' mobility, such as slippage or sinkage. Underestimating these undesired effects may compromise the rovers' operation and lead to a premature end of the mission. Minimizing mobility risks becomes a priority for colonising the Moon and Mars. However, addressing this challenge cannot be treated equally for every celestial body since the control strategies may differ; e.g. the low latency Earth-Moon communication allows constant monitoring and controls, something not feasible on Mars. This letter proposes a Hazard Information System (HIS) that estimates the rover's mobility risks (e.g. slippage) using proprioceptive sensors and Machine Learning (supervised and unsupervised). A Graphical User Interface was created to assist human-teleoperation tasks by presenting mobility risk indicators. The system has been developed and evaluated in the lunar analogue facility (LunaLab) at the University of Luxembourg. A real rover and eight participants were part of the experiments. Results demonstrate the benefits of the HIS in the decision-making processes of the operator's response to overcome hazardous situations.**

***Index Terms*—Machine learning for robot control, space robotics and automation, telerobotics and teleoperation.**

I. Introduction

ROBOTS are the preferred choice for exploring unknown and dangerous space environments. In the last years, their role has increased dramatically since humans are betting on more ambitious space projects, such as the base camps on the Moon or habitats with humans on Mars [1]. These plans require prior preparation and research of the environment. Hence, one of the main objectives is to explore critical regions [2] and to have a deeper understanding of the properties of the place. It demands using rovers, which need to move efficiently and safely on sloping and curved terrains. Nonetheless, it is not a simple task due to the hostile celestial bodies' characteristics, and the undesired effects occurring during the wheel-soil interaction [3], which can cause rover mobility troubles and interfere with the mission objectives. For instance, Mars rovers (e.g., Spirit, Opportunity, and Curiosity), Lunar rovers (e.g., Lunokhod-2, Yutu-2) or Lunar Roving Vehicle (LRV) have experienced different mobility risk phenomena during their missions [4], [5], [6], [7], such as slippage, sinkage, entrapment, among others. Some of them put missions at risk and even caused the early end of the mission.

Previous examples emphasize the seriousness of the problem. Ensuring safe navigation becomes an essential requirement, making risk assessment and prediction an active area of research for rovers [6]. However, tackling the problems on the Moon differs from those on Mars since the robotics requirements are different; e.g. for Mars, a high level of autonomy is needed; whereas the low-latency conditions of the Moon allow engineers to teleoperate robots [8] and to respond faster to unpredictable situations/terrains [9].

This letter proposes a Hazard Information System (HIS), a data-driven approach for detecting and estimating a rover's mobility risks in teleoperation scenarios. The HIS processes the rover's proprioceptive sensors to detect unwanted wheel-soil interaction effects (see Fig. 1). Machine Learning (ML) models (supervised and unsupervised) are trained to estimate slippage risk and stuck status. Additionally, a Graphical User Interface (GUI) was created to present mobility risk indicators to assist and enhance teleoperation.

The structure of the article is as follows. Section II explains the related work. Section III describes the developed HIS and the experimental setup. In Section IV, we present details of the ML models. Section V presents the experiments and results, and finally, Section VI presents the conclusions and the direction of future work.

II. Related Work

The prediction or detection of mobility risks of a rover has been widely studied in the literature with different research directions [9]. ML-based techniques are nowadays on the rise. One of the most well-known strategies is to use exteroceptive sensors, such as cameras or LIDAR, to estimate the types of terrain and their effect on the rover's mobility, including slippage [7], [9]. This strategy is outside the scope of this letter due

Manuscript received 24 February 2022; accepted 26 July 2022. Date of publication 16 August 2022; date of current version 31 August 2022. This letter was recommended for publication by Associate Editor A. Kucukyilmaz and Editor A. Peer upon evaluation of the reviewers' comments. This work was supported by the Luxembourg National Research Fund (FNR) — FiReSpARX Project, ref. 14783405 and the 5G-SpaceLab Project. *(Corresponding author: Sofía Coloma.)*

The authors are with the Space Robotics (SpaceR) Research Group, Interdisciplinary Research Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 1854 Luxembourg, Esch-sur-Alzette, Luxembourg (e-mail: sofia.coloma@uni.lu; carol.martinezluna@uni.lu; bariscan.yalcin@uni.lu; miguel.olivaresmendez@uni.lu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3198794, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3198794

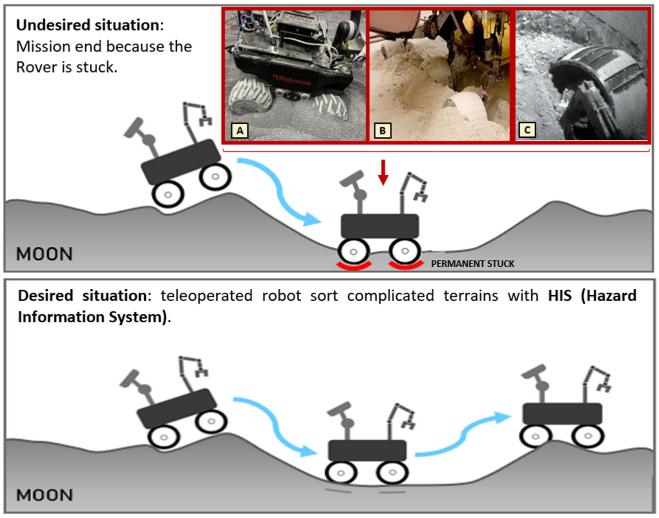

Fig. 1. Mobility scenarios on the Moon. In the top figure, three events of stuck rovers are shown: (A) Rover in LunaLab, (B) NASA recreation of the Spirit sand trap event, (C) Sand trap of Opportunity.

to the high computational requirements, power consumption, and processing time delay, which affect the operator's critical decision-making process. Another strategy is based on proprioceptive sensors, such as the Inertial Measurement Unit (IMU) or encoder [9], to analyze the wheel-soil interaction. These estimations are less dependable on the environment (e.g. lighting conditions), require lower computational calculations, and the sensor data is directly related to the rover state, being more precise than the ones generated with exteroceptive sensors [4], [10]. These reasons motivated the use of proprioceptive sensors for the HIS.

A. *Slippage Level With ML*

In the literature, slippage is usually discretized at three levels (low, medium and high) and trained on different types of soil (sand, gravel, etc.) and conditions (rough terrain, slopes, etc.) [2], [9], [10], [11]. Most of the approaches focus on supervised ML. Only a few focus on unsupervised ML, such as K-means, Self-Organizing Maps (SOM), and Autoencoder [12]. This letter extends the analysis of unsupervised ML algorithms for slippage detection. Apart from the algorithms already tested in the literature, we tested the Density-Based Spatial Clustering of Applications with Noise (DBSCAN), an algorithm robust to noisy data. We also train and validate the model with the lunar analogue facility (LunaLab) and demonstrate the use and feasibility of unsupervised learning for space applications where labelled data is challenging to obtain.

B. *Stuck Status With ML*

Only a few methods have been proposed in the literature to identify if the rover is stuck. In [4], Support Vector Machine (SVM) was used to categorize the stuck level (non-stuck, quasi-stuck, stuck) with supervised ML; and was tested over silica in ISAS/JAXA test field. In this letter, we perform a deeper analysis by conducting an exploratory data analysis to select the best features to determine the stuck status ('stuck,' 'no stuck'). We also compare eight ML algorithms trained and tested using data acquired in the LunaLab. The best algorithm was validated in a teleoperation mission at the LunaLab.

C. *Graphical User Interface*

The literature review revealed a lack of analysis on the impact of mobility risk estimations during teleoperation and a lack of GUI that displays those risks to assist operators. Exiting GUIs display terrain information using wheels data [13], images [6], or rover's sensory data [14], but no information regarding mobility risks is included. Hence, our work creates a customized GUI that assists teleoperation tasks by incorporating mobility risk indicators estimated by the proposed HIS. The system has been validated with users during a teleoperation mission at the LunaLab.

D. *Main Contributions*

In addition to the advances in the state of the art previously mentioned in this section, the main technological contribution of this letter is the Hazard Information System (HIS). A novel software system that processes the rover's proprioceptive information with supervised and unsupervised data-driven approaches to estimate mobility risks and rover information (Slippage Risk, Stuck Detection, Pitch and Roll angles, and Wheel effort alert). The mobility risks are displayed in a GUI to enhance rover teleoperation on the Moon. In addition to this, the present work provides:

- A comparison of 11 ML algorithms to process proprioceptive information (8 supervised and 3 unsupervised), some of which have not been previously tested for mobility risk estimation. Thus, HIS contributes to using Artificial Intelligence (AI) as a powerful tool in space exploration missions.
- An exhaustive evaluation of the proposed HIS using a lunar analogue facility LunaLab (Luxembourg). It includes the assessment of the operator's response and workload during teleoperation and how it is affected by communication delays.

III. The Hazard Information System (HIS)

The HIS is a novel software tool that estimates mobility risks using the rover's proprioceptive sensors (IMU, motor encoder and current sensor). One of the advantages of using proprioceptive sensors is that visibility and environmental characteristics do not affect the estimations (low light or dust clouds). The downside is the lack of future predictions about the terrain that the rover has not stepped on yet. However, in a teleoperation scenario, those problems can be overcome by analyzing the images provided by the cameras, i.e. with the operator's supervision, like when driving cars.

Fig. 2 describes the proposed HIS. The system has two components: the *Data Processing* part and the *Graphical User Interface*. The Data Processing receives information from the proprioceptive sensors of the rover and processes them to

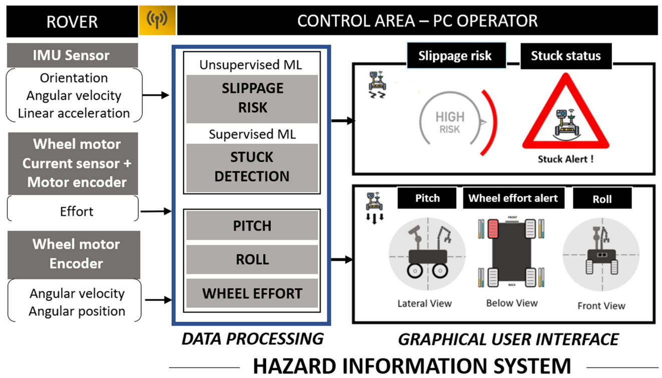

Fig. 2. The Hazard Information System (HIS).

provide useful data for teleoperation purposes. The processed information is displayed to the operator in the GUI, which can be complemented by the rover's camera visualization, as shown in Fig. 3.

A. HIS Graphical User Interface

A customized GUI was created to display the information provided by the HIS to aid the operator's decision-making process during teleoperation (e.g., deciding how to command the rover or the path to explore depending on the terrain and rover status). It also helps avoid risks that are sometimes more difficult to discern with only image-based teleoperation (e.g., due to poor lighting conditions). The combination of HIS with the human-mind and skills improves safety since the operator will be able to react earlier to possible difficulties in the mission, reducing the risks of an early mission end due to permanent immobilization on the lunar soil. The GUI directly displays two mobility risks: the slippage level and the rover stuck status. However, the operator could also deduce whether the rover is sinking by combining the information from the GUI, e.g., knowing pitch and roll angles and stuck status, as described in the video.[1] The following paragraphs provide details of the information displayed in the GUI:

1) Stuck Status: Excessive loss of traction can result in the wheels not being able to generate the necessary force to go forward/backward, causing the rover to get stuck on the soil. The 'stuck' state may be temporal, meaning the rover is stuck for a while and then will regain mobility. However, if the situation is not handled correctly, the rover can end up in a permanently 'stuck' state, making it no longer possible to drive the rover on the lunar surface. A supervised classification model is trained to classify the rover's stuck status ('stuck' or 'not stuck'). The rover is considered 'stuck' when the teleoperation commands do not displace the rover's longitudinal axis. In the GUI, the output of the supervised model is used to generate a visual alert signal when the rover is stuck.

2) Slippage Risk: Slippage is a measure of the loss of traction of a wheeled ground vehicle while driving on certain terrains [11]. The loss of traction can be temporary, but an excessive loss of traction can produce the rover to get stuck. Because of this, it is an important variable to monitor during teleoperation. An unsupervised classification model is used to determine the slippage risk level of the rover. It classifies the proprioceptive data into three risk levels: low, medium and high. A high-risk level indicates a higher chance of the rover getting stuck due to a large loss of traction. In the GUI, the slippage risk levels are displayed in text and with different colors (low, green; medium, orange; high, red), making them visually simple to understand and react on time.

3) Wheel Effort Alert: The wheel effort is the force that moves the wheel along one meter (N/m). This data is obtained from the motor's current sensor and encoder. When the effort value is increased, there is a high probability that the wheel has an abnormal soil interaction. For the SUMMIT-XL rover (see Fig. 3), we have experimentally established a threshold of 10 N/m. Thus, if the wheel effort exceeds this value, it generates an alert in the GUI by turning the affected wheel red.

4) Attitude Angles: The rover's pitch and roll angles are estimated from the rover's linear acceleration provided by the IMU. In the GUI, an image of the rover rotates depending on these angles and can provide clues to the operator about the terrain slope.

B. Scenario and Experimental Setup

HIS's main objective is to reduce rover's mobility risks in a teleoperation scenario by providing assistance to the operator. Possible scenarios include teleoperation not only from Earth, but also from the future MoonBase or Lunar Gateway [1], [8]. Each location has to deal with different communication delays due to distance to the lunar surface, but the MoonBase and Lunar Gateway will allow near real-time teleoperations [8], which is beneficial for telorobotics.

The Lunar Analog facility (LunaLab) from the University of Luxembourg was used to develop and validate the HIS [15]. The LunaLab is an indoor area (11 m $\times$ 7 m) containing 20.000 kg of basalt, used for testing autonomous navigation of lunar robots, multi-robot interaction, manipulation and transportation, among other applications. Different landscapes such as craters, hills, rocky and smooth surfaces can be created, and different visual conditions can also be emulated (conditions of the moon's polar regions, long shadows, etc.). Fig. 3 describes the laboratory implementation setup. It includes two main areas, the test and control areas.

- Test area: It includes the LunaLab and a commercial robot (SUMMIT-XL) adapted as a lunar rover. The robot can operate in autonomous or teleoperation mode, with a payload capacity of up to 65 kg. It is equipped with a LiDAR (RS-LiDAR -M1), three cameras, two computers, the SUMMIT-XL PC and an NVIDIA Jetson Xavier for the cameras.
- Control area: A computer located in this area (Control PC) receives and processes the information transmitted by the rover. The proposed HIS runs on this computer. On the PC screen, the operator can visualize images of the navigation camera and the GUI that displays the mobility

[1]https://youtu.be/O6xLmvAcm6c

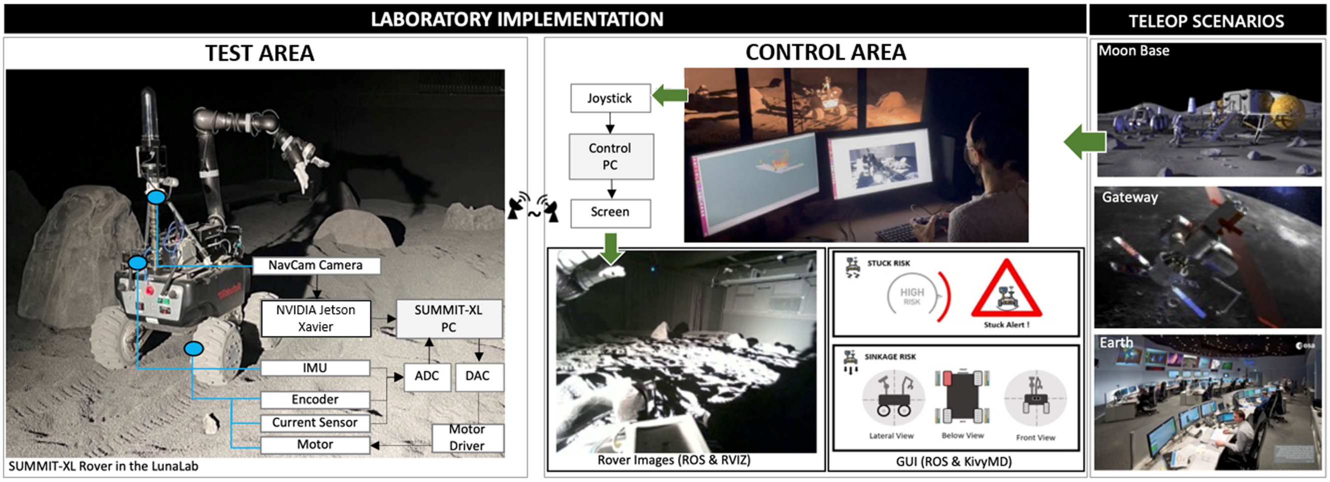

Fig. 3. Laboratory Implementation Setup. For developing and validating the HIS, the test area shown on the left was used. The operator and the HIS system where located on the control area (middle). Images on the right describe possible teleoperation scenarios.

risks estimated by the HIS. This PC also has a joystick used to teleoperate the rover.

The whole system runs in Ubuntu using the Robot Operating System (ROS), where a ROS network is used to communicate with the different computers.

IV. Machine Learning for Estimating Mobility Risks

Estimating mobility risks such as slippage or stuck condition from data is not straightforward. Data collection plays an important role to decide which strategy to use. Identifying the stuck condition experimentally and label the data is easier than identifying slippage risk (difficult to label). Because of this, we implement two different Machine Learning classifiers for rover's mobility risks estimation in HIS. A supervised learning model is used to estimate the stuck condition ('stuck' or 'not stuck'), and an unsupervised learning model is used to estimate the slippage risk level (low, medium and high). The implementation process followed the CRoss Industry Standard Process for Data Mining (CRISP-DM) methodology [16].

A. *Data Collection and Database Generation*

The data were collected experimentally, teleoperating the rover on the LunaLab. We have created three different conditions on the LunaLab surface: 1) nearly flat terrain, 2) rough terrain, and 3) rough terrain with a crater, as depicted in Fig. 4. The most significant experiments from each condition have been selected to build a dataset of 38 k samples. The data include different rover movements, slopes, terrain difficulties, and mobility risk phenomenons (e.g., slippage, temporary entrapment).

The dataset contains 19 input variables: 4x wheels' velocities and 4x efforts, IMU's orientation (4x quaternion), 3x angular velocity, 3x linear acceleration, 1x linear velocity of the rover; and 1 output variable, the stuck state. It was manually labelled during the data collection process by pressing a button when we visually detected that the rover was stuck. The stuck status is used as output variable for training supervised classifiers. On the other hand, labelling slippage risk variable in our experimental setup was difficult to achieve. Because of this, the use of unsupervised classifiers for slippage risk estimation was explored in this letter.

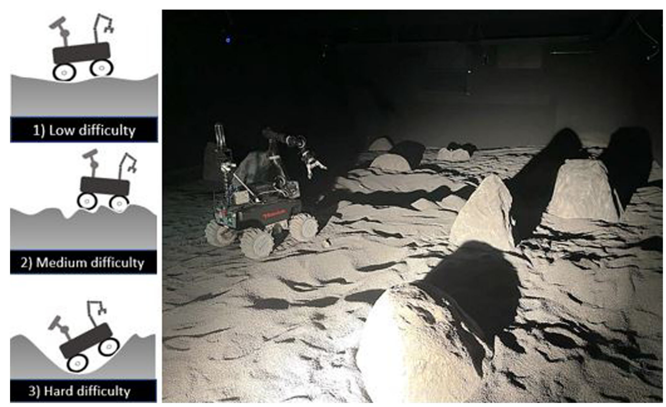

Fig. 4. The rover was teleoperated at the LunaLab facilities over different terrain conditions. The setup emulates the lighting conditions of Lunar South Pole.

B. *Supervised Machine Learning for Stuck Detection*

1) Data Understanding and Preparation: The data were examined and visualised (statistical analysis, histograms, outlier plotting, etc.) to understand and prepare them for modelling. No missing data or relevant outliers were found. The stuck state variable of the dataset was used as the output variable to train the supervised classifier. This output presented imbalanced data, with 66.9% of cases considered 'not stuck' and 33.1% 'stuck'. Feature selection was conducted to reduce the input data (removing meaningless and redundant data). The Principal Component Analysis (PCA) [17] and the Laplacian score were tested but discarded as they did not improve the classification results. After evaluating different methods, the Pearson correlation coefficient [18] was used to analyze the variable dependency,

TABLE I
SUPERVISED CLASSIFIERS FOR STUCK DETECTION

ML Model	MCC	True predictions (%)	False predictions (%)
RF	0.975	99.81	0.19
GB	0.978	99.72	0.28
XGB	0.975	99.69	0.31
LGBM	0.971	99.63	0.37
DT	0.959	99.46	0.54
AdaB	0.900	98.65	1.35
SVM	0.495	92.34	7.66
LR	0.468	92.17	7.83

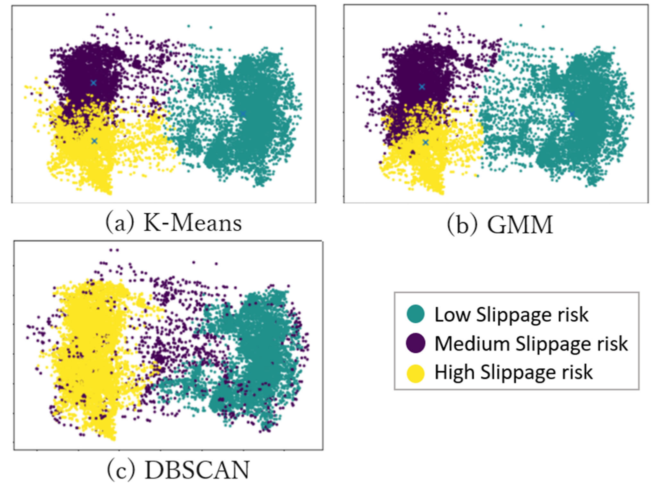

Fig. 5. Clustering results with (a) K-Means, (b) GMM, (c) DBSCAN.

and Random Forest (RF) [19] was used to determine the relevant features (wheels' effort and velocity obtained the highest scores; IMU orientation also ranked as a relevant feature). After the feature analysis and iterative tests in the LunaLab, the selected features to train the supervised models were the IMU orientation (x, y, z, w), the wheels' effort and velocity.

2) Modeling and Evaluation: The Synthetic Minority Oversampling Technique (SMOTE) was applied to balance the output classes [20], and the StandarScaler technique was used to standardise the data. The data were split into 70% for training and 30% for testing. During modeling, we compare the performance of eight well-known ML algorithms for classification [17], [21], [22]: Logistic Regression (LR), Support Vector Machine, Decision Trees (DT), Random Forest, Gradient Boosting (GB), Adaptive Boost (AdapBoost), LightGBM and XGBoost. Ensemble methods have been specially considered for this particular work due to the good results provided in similar works [10].

The mentioned ML algorithms have been trained considering the adjustment of multiple hyperparameters with the Grid Search and Stratified K-Folds Cross-Validation from Scikit-learn (using 30% of training data for validation). The chosen metric to evaluate the algorithms has been the Matthew Correlation Coefficient (MCC), where the best value is 1, and the worst is -1. This metric is more informative and truthful in evaluating binary classifications than accuracy or F1-score, especially when leading with imbalanced datasets [23]. Additionally, the confusion matrix was used to assess the algorithms.

Table I summarizes the results obtained with the test set. The total number of true and false predictions has been specified based on the confusion matrix. It helps to analyze the number of wrong predictions per model. The model with the highest MCC values was Random Forest, followed by models with Gradient Boosting, XGBoost and LGBM. For these models, the false predictions were less than 0.4% (ensemble methods showed good performance for this particular case). Additionally, after testing the algorithms in the LunaLab, the Random Forest model was selected to be integrated into the HIS.

C. Unsupervised Machine Learning for Slippage Risk

1) Data Understanding and Preparation: The data analysis was similar to the one discussed in Section IV-B1, but the data were processed differently in this case. The variance of the characteristics was considered during periods of T = 30 ms. The reason is that identifying slippage levels as trends depends on the time intervals [2]. Data from sensors with different sample times (IMU 1.25 ms, encoder = 0.05 ms, and current sensor = 0.05 ms) have been merged with 30 ms time series to be compatible with the 20 ms ROS topic publishing rate configured in the LunaLab's communication infrastructure. The sampling time of the time series is slower than the ROS topic publishing rate to avoid the aliasing problem.

For feature selection, the Laplacian score was used [24], where the importance of each feature is evaluated by its locality preservation. The feature importance categorization provided by the Laplacian score was assessed in the LunaLab by testing different models with various combinations of features. The results suggest that the most representative features for estimating slippage risk are the wheels' effort, followed by the wheels' velocity, IMU angular velocity (y-axis), and linear acceleration (x-axis and z-axis). Intuitively, the combination of these variables makes sense. When the rover is slipping, more effort is needed, the wheels' velocity increases, and angular velocity and linear acceleration keep stable.

2) Modeling and Evaluation: The selected features of the dataset were used to feed the unsupervised models that generate three clusters, corresponding to the three slippage risk levels (Low, Medium, High). Three popular clustering algorithms were tested: K-Means [17], Gaussian Mixture Model (GMM) [25], and DBSCAN [26]. They interpret the input data and find natural groups or clusters in the feature space. DBSCAN has the particularity that it can take any irregular shape, unlike k-Means or GMM, where their clusters are more or less spherical.

Fig. 5 shows the 2D scatter plots of the three clusters formed by each algorithm. PCA was applied only for visualization purposes to reduce the dimensionality of the input data [17]. The figure shows that the GMM and K-Means clusters are similar and the three groups can be separated. The Silhouette score technique was used as an internal statistical technique to evaluate and validate the goodness of the obtained clusters [27]. The silhouette range belongs to $[-1, 1]$. The upper bound '1' means that clusters are well apart, and the lower bound '-1' means

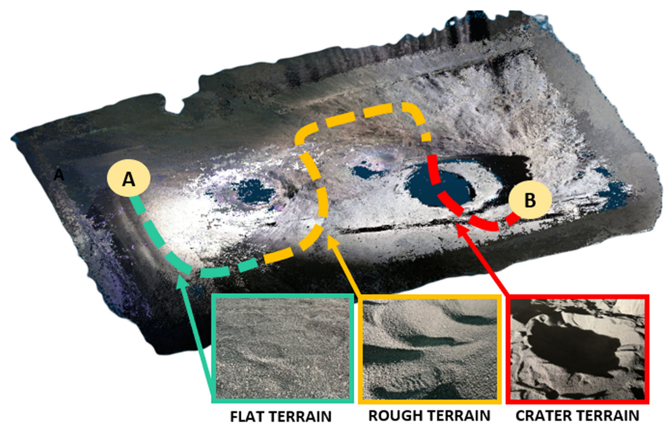

Fig. 6. LunaLab experiments scenario. The dashed line represents the path to follow with the rover (A to B). It has different terrain conditions (flat, rough and rough with crater).

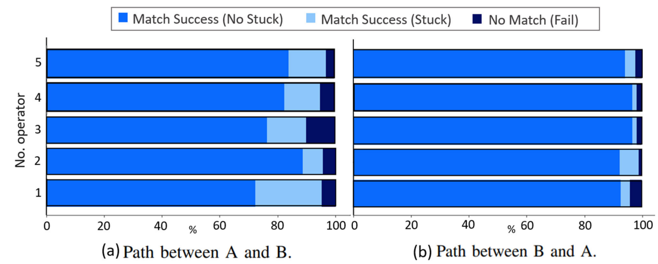

Fig. 7. Comparison between ML model and human perception estimation. The bars represent the match and fail for each user.

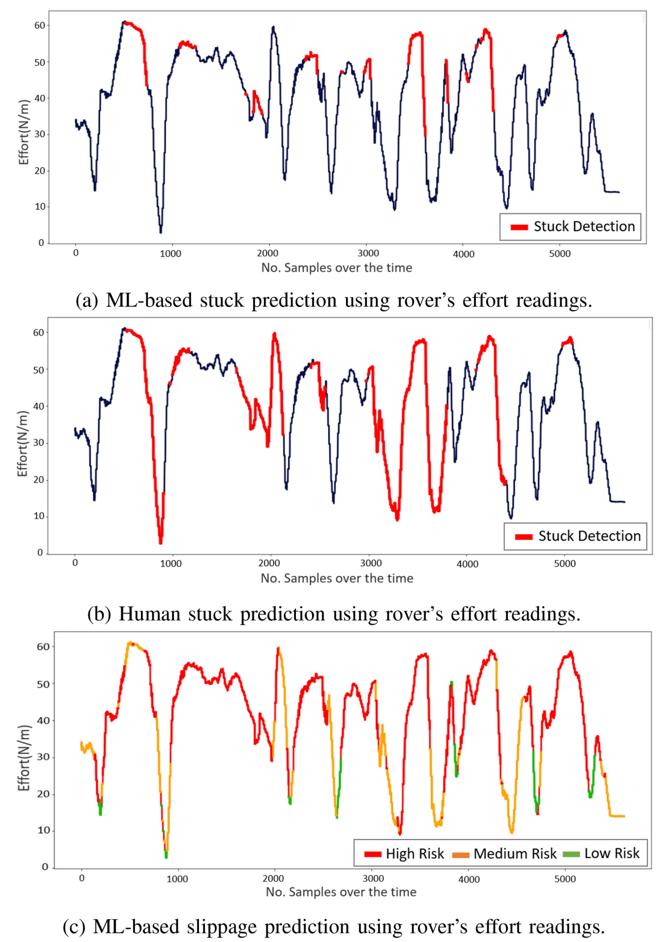

Fig. 8. Representation of mean effort over time with ML detection and human estimation. It is a representation of a critical moment when the rover suffered temporarily stucks during an experiment.

that the distance between clusters is not significant. The results indicate that K-Means with a 0.41 score is slightly better than GMM with 0.40, and DBSCAN has the lowest score with 0.31. Further, we tested the algorithms in the LunaLab. The K-Means model was selected to be integrated into the HIS system because it gave us satisfactory outcomes during real-time tests, and it is considered one of the fastest algorithms, making it attractive for teleoperation purposes.

V. Experiments and Results

The performance of the proposed HIS has been validated with two experiments. The first experiment focuses on analyzing the ML approach for mobility risk estimations. The second examines the advantage of the HIS in a teleoperated mission, including the usability of the GUI. People such as anonymous users have participated in this research. Before the experiments, the participants received training on how to use the system and received details related to the test (e.g., how to interpret the GUI information, what 'stuck' status means in a rover, etc.). Later, they underwent a training session to drive the rover.

In both experiments, data were acquired while users teleoperated the SUMMIT-XL rover at LunaLab. The hardware and software used in the experiments are described in Section III. Fig. 6 shows the scenario created in the LunaLab and the route the rover should follow. The lab was configured with different terrain conditions (flat, rough and crater). The planned route included passing through a crater, one of the most challenging zones for rovers. The lighting conditions emulated the Moon's South Pole during the experiments, as Fig. 4 shows.

A. *Validation of the Risk Indicators*

The experiment consisted of teleoperating the rover around the LunaLab following the path represented in Fig. 6 from A to B and B to A (a path with different terrain slopes, up 20.55°), recording the I) output of the stuck detection ML algorithm and the II) stuck status of the rover determined by the participants. Five persons were asked to contribute to the experiment (4 male, 1 female) between 25 and 34 years ($M = 28.75$, $SD = 3.34$). While one user was teleoperating the rover, another person was directly checking the rover and pressing a button when they visually saw that the rover was temporarily stuck.

For the 'stuck' estimation, the results indicate that the ML model matches an average of 95.92% $SD = 2.52$ with the human estimation, considering all the tests conducted. Fig. 7 details the results found in each experiment per operator. To visually appreciate the differences between human and ML estimation in detail, Fig. 8 represents the evolution of the mean of wheels' effort and the stuck detection over a specific time and samples. Stuck detection is highlighted in red: Fig. 8(a) shows the ML-based stuck prediction and Fig. 8(b) the human estimation. The figures show that the human considers the rover to be stuck for a longer period than the ML-based predictions. This may be due to the reaction capacity of the operator because it only has a visual indicator, unlike the ML model that is using the proprioceptive rover' sensors.

This experiment was also used to validate the unsupervised model. Using the same sample interval, Fig. 8(c) displays how the unsupervised ML model predicts the 'slippage risk level,' which can be visually compared with the stuck prediction of Fig. 8(a) and 8(b). However, considering all the experiments for slippage estimation, the high-level risk has matched an average of 93.75% with the ML stuck detection and 64.52% with human estimation. It shows a high relation with the stuck status. But the remaining percentage also may warn of a high probability of getting stuck, as Fig. 8 shown.

For further validation of the unsupervised model, we have divided path A to B data into three types of terrain, as illustrated in Figs. 6 and 4. The slippage indicator has shown the following risk level estimations in each one: Flat terrain (94.93% low, 3.91% medium, 1.16% high), Rough terrain (3.56% low, 87.95% medium, 8.49% high) and Rough terrain with crater (1.65% low, 18.84% medium, 79.51% high). It shows a rational behaviour of the unsupervised ML model for slippage classification in each terrain.

B. Validation of Hazard Information System

The second experiment evaluated the operator workload by analyzing the impact of the HIS on the teleoperation task. The user study was conducted with N = 8 participants (7 male, 1 female) between 25 and 34 years (M = 30.37, SD = 3.07). Two participants were familiar with driving a rover, and the other six were not. The method to assess the analysis was the NASA Task Load Index (TLX). It is a tool to evaluate the workload perceived by the user (0–100) on a six-dimension scale: mental, physical and temporal time demands; effort frustration; and perceived performance [28].

The experiment consisted of teleoperating the rover from points A and B, as shown in Fig. 6. Each operator performed the path three times under different conditions (1) using the image of the on-board camera (no delay), (2) using the information from the on-board camera and the GUI from the HIS (no delay), and (3) using the camera and the GUI from HIS, with delay. A 3 s delay was simulated in the GUI to visualize the effect of the delay on the operator.

Each operator filled out three forms per situation (camera, camera + HIS, camera + HIS with delay). Fig. 9 shows the results found in the experiment. Each bar represents the average score of the operators per situation in each of the six-dimension scales. The results show that the perception of the mental demand, effort to perform the task and frustration were significantly reduced using the HIS, a 22.5% less. In addition, the perception of how satisfied they were executing the task (performance bar) increased when the HIS was used.

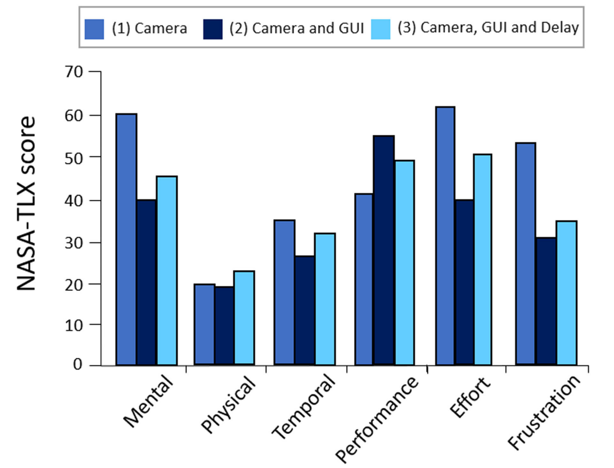

Fig. 9. Average teleoperation workload, (1) with camera (no delay), (2) camera + HIS (no delay) and (3) with camera + HIS (delay).

In general, using Camera + HIS gave the operator confidence to complete the task. Teleoperating the rover only with the camera information and moving inside craters (reduced illuminations) makes it difficult to discern when stuck. In this regard, the rover's information provided by HIS helped operators in decision-making. The results also show that adding the delay increased the mental demand, the effort and the frustration of the operator when conducting the task. The 62.5% of the users reported that it was annoying and harder to identify potential mobility hazards with the 3 s delay.

We also asked participants to provide feedback on the GUI's usefulness for the task: (1) was the GUI functional? (2) which risk indicators were and were not very helpful? (3) what will you improve to the GUI? All participants agreed that the knowledge of the slippage risk was not very useful for the teleoperation task since their attention was mainly fixed on the camera images. They checked the slippage risk information only occasionally. Participants agreed that the preferable variables were stuck status, pitch and roll inclination, and wheel information. They considered these variables helpful in making decisions about the rover's teleoperation, especially in critical parts of the path, e.g. the craters. Based on the results, we hypothesize that the slippage indicator may be more helpful in providing feedback in an autonomous scenario.

- Operator' performance evaluation

Table II summarizes the parameters considered to evaluate the operator's performance: time to move the rover among the three types of terrain and the success rate of finishing the task. The parameters were analyzed under the three experimental conditions. As a result, the flat and rough terrain success rate was 100% under all conditions. However, the crater area was not successful for 25% of the operators. They could not go out of the crater and reach point B using only the information from the camera.

Regarding the time, no relevant conclusions were observed with flat terrain, but some differences were observed in rough and crater terrain. The operator's performance was better combining the camera and HIS; they spent less time conducting the task. Results make sense since we experience that operators tend not

TABLE II
OPERATOR'S PERFORMACE EVALUATION

Experiment Conditions	Flat Terrain	Rough Terrain	Crater	
	Time (sec)	Time (sec)	Time (sec)	Success Rate (%)
1) Camera	M=32.2 SD=4.8	M=142.2 SD=30.5	M=140.3 SD=28.1	75
2) Camera + HIS	M=29.7 SD=5.4	M=115.7 SD=13.1	M=96.2 SD=9.5	100
3) Camera + HIS + Delay	M=28.2 SD=5.7	M=122.5 SD=19.8	M=105.2 SD=8.2	100

to be good at identifying on time when they are temporarily stuck. They tend to continue driving as usual. Instead, HIS notifies it quickly. Therefore, the operator can send the appropriate commands to avoid sinkage.

VI. CONCLUSION AND FUTURE WORK

This letter presented the novel Hazard Information System based on proprioceptive sensors and ML techniques. The HIS estimated the rover's mobility risks and displayed them in a GUI to assist the operator during teleoperation. Eight supervised ML models were analyzed to estimate the rover's stuck state, and three unsupervised ML models were used to estimate slippage risk levels. A Lunar analogue facility was used to verify the ML models' performance and analyze the utility of HIS. The results indicate that HIS helps reduce the operators' workload by providing information about the rover's status and risks, helping them in decision-making. Future work will analyze the effects of the delay in three teleoperation scenarios: the MoonBase, Lunar Gateway and Earth. Additionally, HIS will be integrated with the control of the rover to allow the rover to make in-situ decisions and to improve autonomous lunar navigation.

REFERENCES

[1] C. M. Edwards and T. Cichan, "From the Moon to Mars base camp: An updated architecture that builds on artemis," *ASCEND*, 2021, Art. no. 4137.
[2] M.-R. Bouguelia et al., "Unsupervised classification of slip events for planetary exploration rovers," *J. Terramechanics*, vol. 73, pp. 95–106, 2017.
[3] A. J. R. Lopez-Arreguin et al., "Do lunar rover wheels sink equally on Earth and Moon?," *Results Phys.*, vol. 15, 2019, Art. no. 102617.
[4] T. Omura and G. Ishigami, "Wheel slip classification method for mobile robot in sandy terrain using in-wheel sensor," *J. Robot. Mechatronics*, vol. 29, no. 5, pp. 902–910, 2017.
[5] R. E. Arvidson et al., "Opportunity Mars Rover mission: Overview and selected results from purgatory ripple to traverses to endeavour crater," *J. Geophysical Res. Planets*, vol. 116, no. E7, 2011, Paper IAC-20-A3.2B.3.
[6] K. Skonieczny et al., "Data-driven mobility risk prediction for planetary rovers," *J. Field Robot.*, vol. 36, no. 2, pp. 475–491, 2019.
[7] R. Gonzalez et al., "Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges," *J. Field Robot.*, vol. 35, no. 4, pp. 564–577, 2018.
[8] J. O. Burns et al., "Science on the lunar surface facilitated by low latency telerobotics from a lunar orbital platform-gateway," *Acta Astronautica*, vol. 154, pp. 195–203, 2019.
[9] A. J. R. Lopez-Arreguin and S. Montenegro, "Machine learning in planetary rovers: A survey of learning versus classical estimation methods in terramechanics for in situ exploration," *J. Terramechanics*, vol. 97, pp. 1–17, 2021.
[10] R. Gonzalez et al., "Characterization of machine learning algorithms for slippage estimation in planetary exploration rovers," *J. Terramechanics*, vol. 82, pp. 23–34, 2019.
[11] R. Gonzalez et al., "Slippage and immobilization detection for planetary exploration rovers via machine learning and proprioceptive sensing," *J. Field Robot.*, vol. 35, no. 2, pp. 231–247, 2018.
[12] J. Kruger, A. Rogg, and R. Gonzalez, "Estimating wheel slip of a planetary exploration rover via unsupervised machine learning," in *Proc. IEEE Aerosp. Conf.*, 2019, pp. 1–8.
[13] R. K. Sidharthan et al., "Adaptive neuro-fuzzy inference system based on-the-move terrain classification for autonomous wheeled mobile robots," *Int. J. Comput. Vis. Robot.*, vol. 10, no. 6, pp. 545–560, 2020.
[14] J. C. Gómez et al., "Hardware and software architecture for a Rover robot," *Revista Avances en Sistemas e Informática*, vol. 8, no. 3, pp. 183–189, 2011.
[15] P. Ludivig et al., "Building a piece of the Moon: Construction of two indoor lunar analogue environments," in *Proc. 71st Int. Astronautical Congr.*, 2020.
[16] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in *Proc. 4th Int. Conf. Practical Appl. Knowl. Discov. Data Mining*, 2000, pp. 29–40.
[17] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.:Cambridge Univ. Press, 2014.
[18] J. A. O. Benesty, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.
[19] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Boca Raton, FL, USA:CRC Press, 2019.
[20] N. V. Chawla et al., "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[21] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, 2013, Art. no. 21.
[22] D. Zhang and Y. Gong, "The comparison of LightGBM and XGBoost coupling factor analysis and prediagnosis of acute liver failure," *IEEE Access*, vol. 8, pp. 220990–221003, 2020.
[23] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomic.*, vol. 21, no. 1, pp. 1–13, 2020.
[24] X. He et al., "Laplacian score for feature selection," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 507–514.
[25] C. A. Bouman et al., "Cluster: An unsupervised algorithm for modeling Gaussian mixtures," Apr. 1997. [Online] Available: https://engineering.purdue.edu/~bouman/software/cluster/
[26] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future," in *Proc. IEEE Fifth Int. Conf. Appl. Digit. Inf. Web Technol.*, 2014, pp. 232–238.
[27] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Analytics*, 2020, pp. 747–748.
[28] S. G. Hart, "Nasa-task load index (NASA-TLX); 20 years later," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, 2006, pp. 904–908.