DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Application project

Multi-fidelity machine learning for terramechanics

Witold Merkel, Adam Rydelek

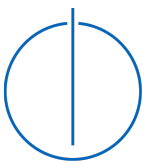
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Application project

Multi-fidelity machine learning for terramechanics

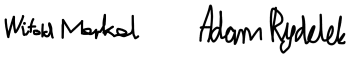
Author:	Witold Merkel, Adam Rydelek
Supervisor:	Dr. Felix Dietrich
Advisor:	Vladyslav Fediukov
Submission Date:	30.09.2022



I confirm that this application project is my own work and I have documented all sources and material used.

Munich, 30.09.2022

Witold Merkel, Adam Rydelek



Acknowledgments

Firstly we would to thank our examiner, Dr. Felix Dietrich, for his support and providing helpful guidance with the project. His knowledge and expertise have been really helpful with the organization of steps in our project.

Secondly, we would like to thank our advisor Vladyslav Fediukov for his help with generating the data, suggesting us how to move forward in times of tie-ups, giving us directions through the entire time of the project, and spending his tme in meetings with us.

Lastly, we would like to thank Tzu-Chien Chang, whose master's thesis gave us some sense of directions in our work.

Abstract

The goal of the project is to explore the possibilities of multi-fidelity machine learning for wheel-soil interactions. The scarcity of high-fidelity data in the problem leads to attempts to enhance models based on lower-fidelity data, which is easier to obtain. With multiple overlapping features present in the Mars rover dataset, dimensionality reduction techniques are implemented to enhance performance and combat memory problems. The correlation between low and medium fidelity data we explore to assess the propriety of multi-fidelity methods along with various Gaussian Processes kernels' fitness for the task. Along with Gaussian Processes, a Neural Network approach we introduce for the multi-fidelity regression task of predicting the force of the Mars rover incorporating both tabular and image data. The models we benchmarke and compare according to their accuracy, the number of required data points, and possible further improvements.

Contents

1 Structure of the report

In chapter 1 we give a problem introduction. In chapter 2 we discuss background, meaning doing a literature overview and analysing work previously done at TUM on this topic. In chapter 3 write about all our work done. Firstly we in detail describe the exploratory data analysis process, after that our experiments with distinction between neural networks and Gaussian process regression. At the end in chapter 4 we write our conclusions and ideas for future work.

2 Problem introduction

The problem of wheel-soil interactions has been present since the first developments of unmanned rovers used in the space research. Mars has been a major point of interest for space exploration since the 1960s [Dra20]. With the technological development, the goal has become closer as the Mars Pathfinder was deployed on December 4, 1996 successfully landing on Mars 7 months later. Since then multiple rovers have explored the Mars' soil. The latest being the Perseverance vehicle which is still traversing the planet since landing on February 18, 2021. A major challenge for successful travel on the planet is the uncertainty of wheel interactions with soft soils such as sand or mud. With the latest rover landing in the Jezero crater, the problem of efficient traversal became crucial [Dra22].

Figure 2.1: Mars Science Laboratory (MSL) engineering model tested on dry, loose sand [Con+12].

With the experience gained from multiple previous missions as well as experiments performed on Earth, the physics model and properties of wheel interactions are being analysed utilising multiple ground contact models in various level of detalization. The scarcity of high fidelity data leads to exploring alternative prediction methods. Machine learning and models combining could provide useful, real-time locomotion predictions [Guo+22].

3 Background and related work

3.1 Literature overview

The following topics are crucial for the project and are described in this overview.

- Wheel locomotion,
- Overview of the terramechanical models such as: DEM, SCM and TeRRA that we will be using throughout the project,
- Gaussian processes,
- Kernels and kernel selection,
- Multi-fidelity approach to resolving problems.

As described by Cheng et al. [Hu+21], the process of rover movement is a complicated procedure that requires not only precision, but also knowledge about the surface that the vehicle is moving on. Gallina et al. [Gal+14] states that we have to take into account the torque and force generated in both perpendicular and parallel directions to the soil in the point of contact with the rover's wheel.

In our application project we are combining a lot of approaches, one of those is multi-fidelity, meaning that we are combining results of different fidelities. Which means simulations of different level of precision and computational cost. In the work of Guo et al. [Guo+22], it is shown that joining multi-fidelity data can be effective in enhancing the performance of the model as shown in Lee et al. [Lee+19a]. We will try this approach with the data from multi-fidelity simulations that are described in the works of Hawker et al. [Haw+18], Tasora et al. [TMN18], and Barthelmes [Bar18].

An approach to utilise neural networks in solving multi-fidelity problems has been described by Guo et al. [Guo+22]. The process of interactions and both forces and trajectories within runs can be described with Gaussian processes [HW21]. Combining the statistical approach of Gaussian modelling with machine learning allows for more complex models creation. This will enhance the search for the best model possible.

Kernel manipulation is an advanced fitting method. It is shown in the work of Abdessalem et al. [Abd+17] that selecting right kernel is a problem itself. We will try

to unveil the underlying trend in data and find the best kernel for this task. Lee et al. [Lee+19b] states that the process of selecting the right kernel can be automated. We will try this approach to see if it is suitable for task at hand.

From further analysis we can see that neural networks can be a valid options to model Gaussian processes both in single-fidelity as in multi-fidelity regression problems [Guo+22]. We will try and use this fact to enhance our performance on the multi-fidelity data obtained from DEM, SCM and TeRRA models [Haw+18], [TMN18]. Similarly to the approach by Guo et al. [Guo+22], we will use multiple architectures of networks, as well as try to fit different kernels for the data to get as much information from it as possible.

3.2 Previous work at TUM

The problem itself has been already explored in the master thesis of Chang [Cha22]. The theoretical basis of rover mechanics and the usage of machine learning to enhance already existing models has already been laid out. The crucial finding from the paper that can enhance results with further work is the way to compose the data to highlight it's importance. This allows to find connections between different levels of fidelity and can lead to finding important features.

The EDA has been conducted and produced meaningful results to build on. The goal of finding a better performing model is easier to benchmark thanks to the introduction of evaluation measures in the thesis [Cha22]. The finding has led to the introduction of time as a measure to ensure real-time capabilities. The metrics of Mean Absolute Error (MAE), Mean Squared Error (MSE), and Certainty of Prediction (CoP) have been used as performance indicators.

In the thesis a basic Multilayered Perceptron (MLP) model has been used. This has led the basis of utilizing deep learning in the problem with a single layer. To deal with all the available input data including surface images Convolutional Neural Networks (CNN) have been utilized. An approach to include such deep networks in the analysis of numeric data proved inefficient. A Gaussian Process Regression model has been also included and experimented with using different kernels. The approach has been used to create a non linear multi-fidelity model that is linked using the Gaussian process. With the combination of such models and both MLP and CNN the author has achieved meaningful results.

The thesis is concluded with a complete benchmark of all approaches, setting a standard to be challenged in this project.

4 Methodology

4.1 Exploratory Data Analysis

4.1.1 Introduction to data

The first part of this project is exploratory data analysis in which we closely examine the data and look for correlations between the variables. Data from both models [TMN18; Bar18] is created in the same way. Simulations of runs of the Mars Rover have been conducted using different terramechanical models [Hu+21]. One simulation consists of 151 rows of data and each row is composed of physical parameters of the rover at given moment and a height map of the surface below. Each record has parameters that describe output forces, output torques, coordinates, velocities, angular velocities and gravity normal, all of those are in each of the x, y and z directions. This means that there is a total of 18 variables. Additionally to that columns from 18th to the 4112 are the image, which is a 64 x 64 picture of a 15 cm x 15 cm surface below the wheels with example shown in figure 4.1.

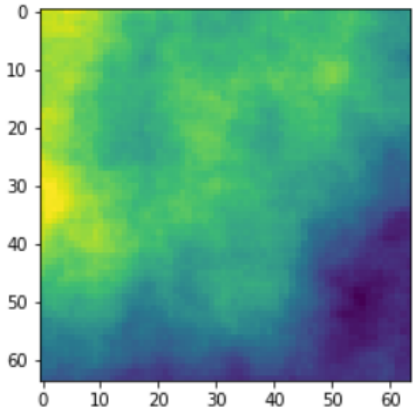


Figure 4.1: Example of soil photo

TeRRa method [Bar18] has been used to generate the low fidelity data. Despite that fact it can be valuable to the final model, that is because it was proven that low computational power is required to work with data coming from this model.

Soil contact model [TMN18] was used to produce the medium fidelity data. This model results in more reliable and more accurate data points, but because of that working with them requires more computational power. Hence in our modelling there are fewer medium fidelity points, than low fidelity.

For this part of work we have decided to use 100 simulations from each fidelity. This comes up to 30200 records used. For all the experiments described below we will be using *to be filled* data points of low fidelity and *to be filled* data points of medium fidelity.

In this task we decided to predict the force along x coordinate. As forces and torques are correlated with coefficient one, we decided to drop all three variables describing torque, to avoid overfitting of the models.

4.1.2 Analysis for TerRa data

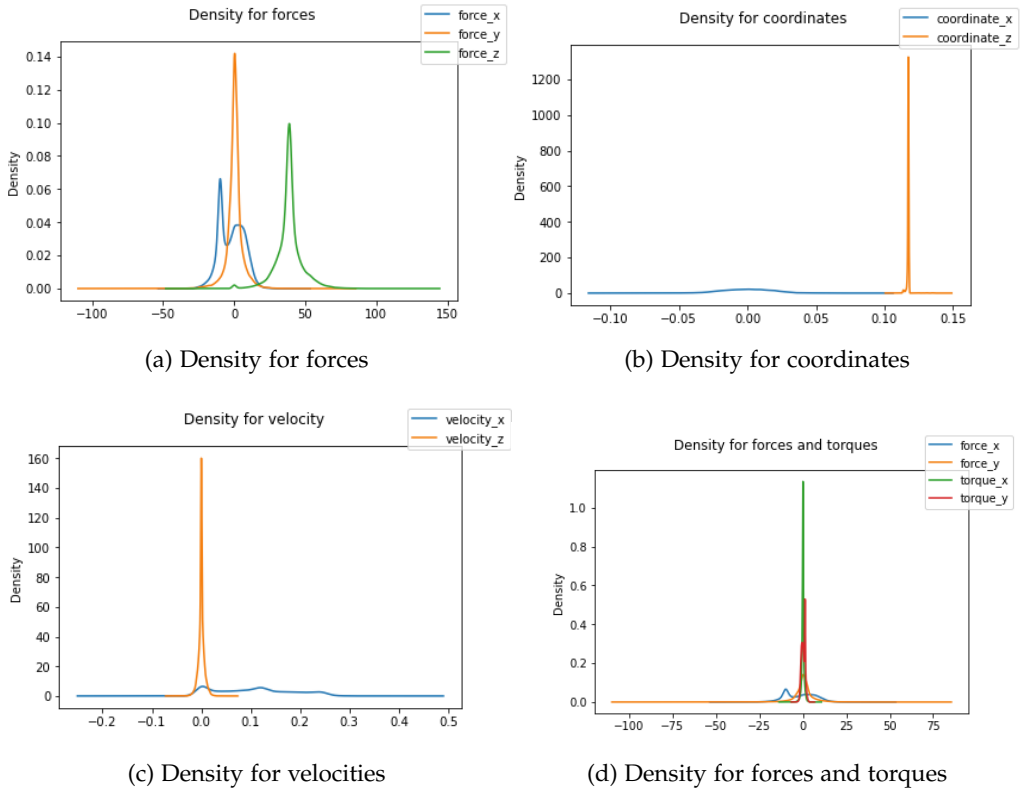
In the given Terra data, the following statistical values have been analysed.

- Mean
- Standard deviation
- Minimum
- First quartile
- Median
- Third quantile
- Maximum

Table 4.1: Positional statistics for low fidelity data

	force_x	force_y	force_z
mean	-1.96	0.35	39.38
std	7.2	5.82	9.31
min	-10.68	-43.91	0.0
max	14.74	14.33	76.68
	torque_x	torque_y	torque_z
mean	0.04	0.24	-0.0
std	0.73	0.9	0.0
min	-5.49	-1.84	-0.0
max	1.79	1.33	0.0
	coordinate_x	coordinate_y	coordinate_z
mean	-0.01	0.0	0.12
std	0.01	0.0	0.0
min	-0.02	-0.0	0.11
max	0.02	-0.0	0.13
	velocity_x	velocity_y	velocity_z
mean	0.09	0.0	-0.0
std	0.08	0.0	0.01
min	-0.01	0.0	-0.02
max	0.25	0.0	0.02
	anqular_velocity_x	anqular_velocity_y	anqular_velocity_z
mean	0.0	0.8	0.0
std	0.0	0.71	0.0
min	0.0	0.0	0.0
max	0.0	2.0	0.0
	gravity_x	gravity_y	gravity_z
mean	0.05	0.0	-0.99
std	0.1	0.0	0.0
min	-0.14	0.0	-1.0
max	0.19	0.0	-0.98

This analysis resulted with table 4.1, it helped detecting variables that had zero or close to zero variance, meaning they would not be of much help in the models. Hence we decided to eliminate: coordinate_y, coordinate_z, velocity_y, velocity_z, angular_velocity_x, angular_velocity_z and gravity_y from further work. Next step was to visualise the distributions of all the chosen variables, it is presented in figure 4.2. We opted for density plots, as they depict the tendencies in the data in a good manner.



(a) Density for forces

(b) Density for coordinates

(c) Density for velocities

(d) Density for forces and torques

Figure 4.2: The result of density analysis

From 4.2 subplot a we can see that there is a lot of change in all three of the forces. Both in velocity and coordinate variables, nearly all the mass for the z coordinate is concentrated and for the x coordinate it is spread out. The last subplot d is placed here to show why we decided to opt out of working with torques, they are to similar to the forces for us to get meaningful results.

For further analysis of the relationships between variables we decided to do a correlation analysis of the remaining dataset. We present the results in figure 4.3.

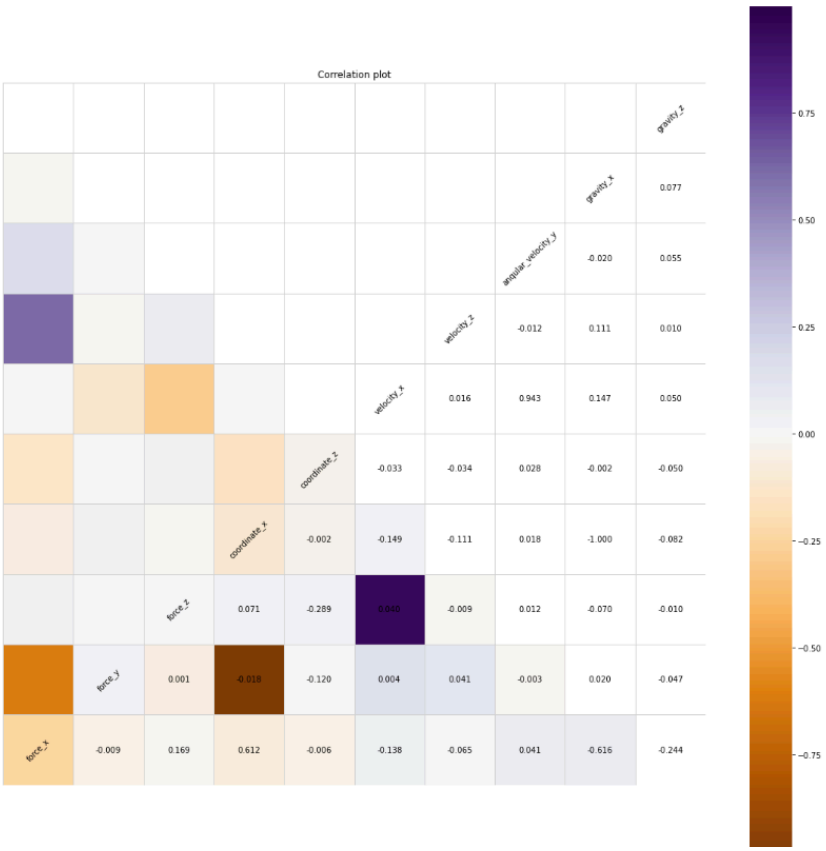


Figure 4.3: Correlation plot low fidelity data

This allowed us to make a decision on the final set of features we will choose from this level of fidelity. Those are: coordinate_x, coordinate_z, velocity_x, velocity_z, angular_velocity_y, gravity_x, gravity_z.

4.1.3 Analysis for SCM data

For the medium fidelity data we decided to take the same steps. First were the positional statistics to determine which variables have no variance.

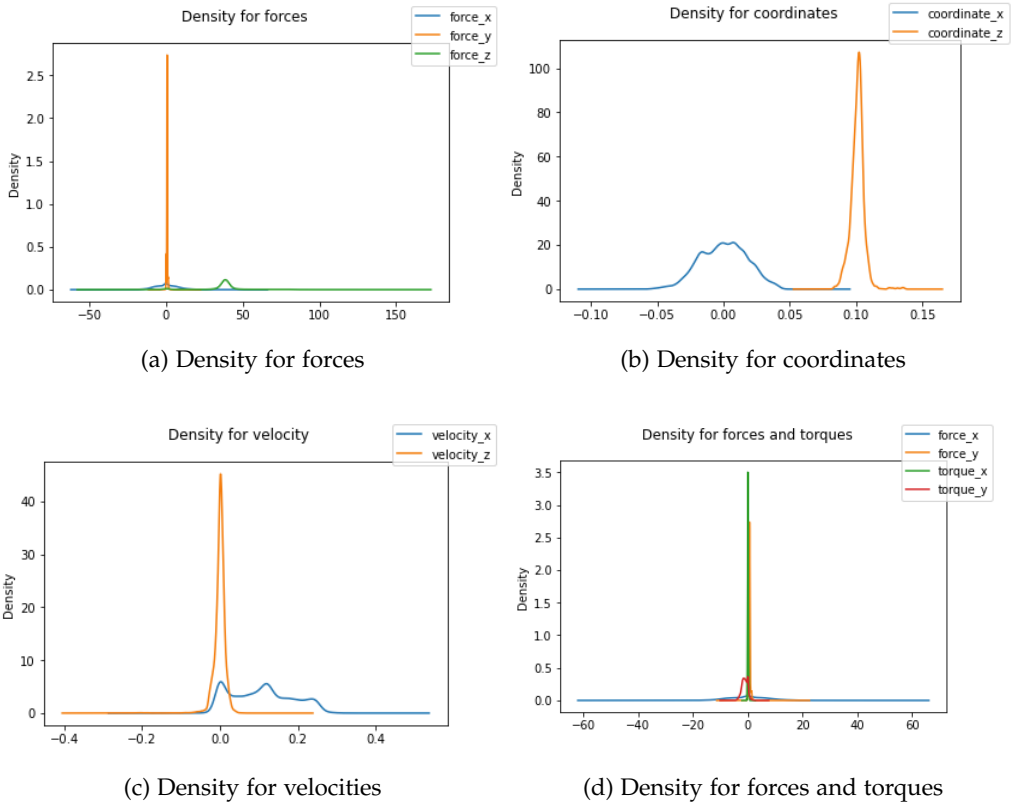
Table 4.2: Positional statistics for medium fidelity data

	force_x	force_y	force_z
mean	-0.75	0.69	38.91
std	5.78	0.39	21.58
min	-11.56	0.0	-0.0
max	16.71	1.83	112.94
	torque_x	torque_y	torque_z
mean	0.06	-0.78	0.02
std	0.15	1.04	0.02
min	-0.58	-3.51	-0.04
max	0.62	0.99	0.07
	coordinate_x	coordinate_y	coordinate_z
mean	-0.0	0.0	0.1
std	0.01	0.0	0.0
min	-0.02	0.0	0.09
max	0.03	0.0	0.13
	velocity_x	velocity_y	velocity_z
mean	0.09	-0.0	-0.0
std	0.08	0.0	0.02
min	-0.01	-0.01	-0.19
max	0.25	0.0	0.04
	anqular_velocity_x	anqular_velocity_y	anqular_velocity_z
mean	-0.0	0.8	-0.0
std	0.0	0.71	0.0
min	-0.0	0.0	-0.0
max	0.0	2.0	0.0
	gravity_x	gravity_y	gravity_z
mean	0.02	-0.02	-0.99
std	0.14	0.0	0.01
min	-0.28	-0.02	-1.0
max	0.17	-0.02	-0.96

From table 4.2 we can deduce that variables: coordinate_y, anqular_velocity_x, an-

qular_velocity_z and gravity_y have no variance or variance close to zero, thus we will move forward without considering them.

The next step is to look at the density plots. They are presented in figure 4.4.



(a) Density for forces

(b) Density for coordinates

(c) Density for velocities

(d) Density for forces and torques

Figure 4.4: The result of density analysis

In this case the behaviour is different. For force_y nearly all the mass is around zero, meaning that it is often equal or close to zero, but the other forces are more spread out. Also in the case of velocities and coordinates variables are more spread out in case of the z coordinate, but in the other one the values seem to be more focused around certain interval of values. Once again subplot d shows enormous correlation between forces and torques.

The last part of this step was to look at the correlations between the variables to once again choose those that are well suited for the problem, we keep in mind the variables already chosen in previous analysis. We present the results in figure 4.5.

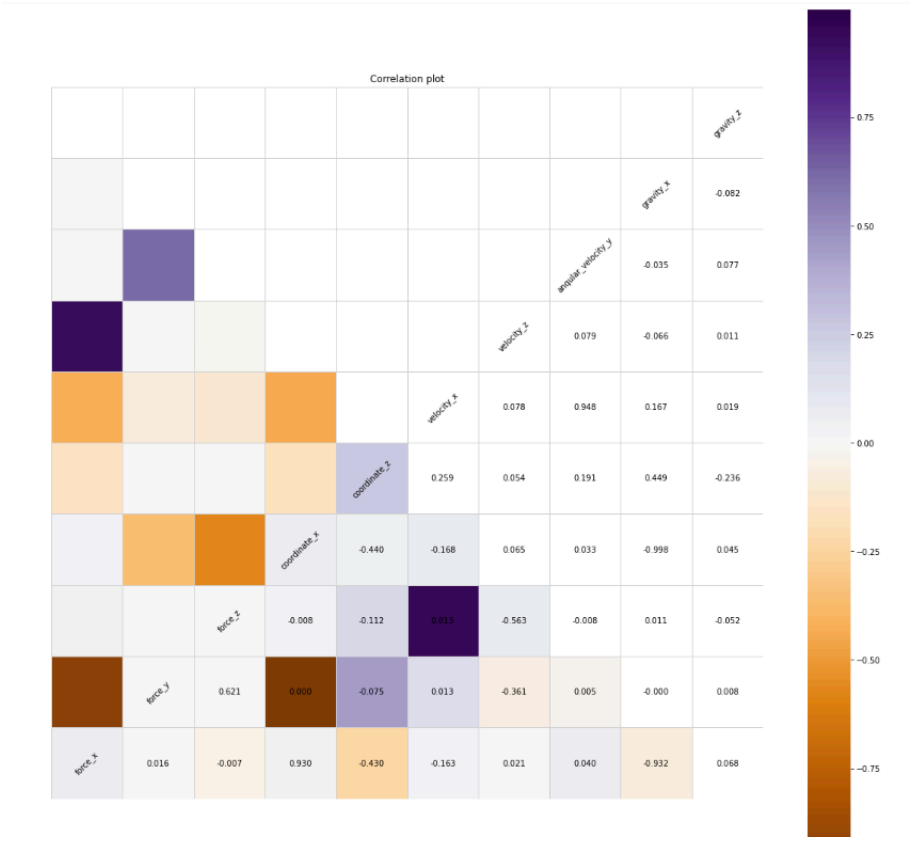


Figure 4.5: Correlation plot medium fidelity data

This allowed us to make a decision on the final set of features we will choose from this level of fidelity. Those are: coordinate_x, coordinate_z, velocity_x, velocity_z, anqular_velocity_y, gravity_x, gravity_z.

4.1.4 Multi-level exploratory analysis

In this step we decided to look closely at the correlation between variables across the fidelity levels. For all the chosen variables we calculated the correlation coefficient between datasets. The results are presented in figure 4.6.

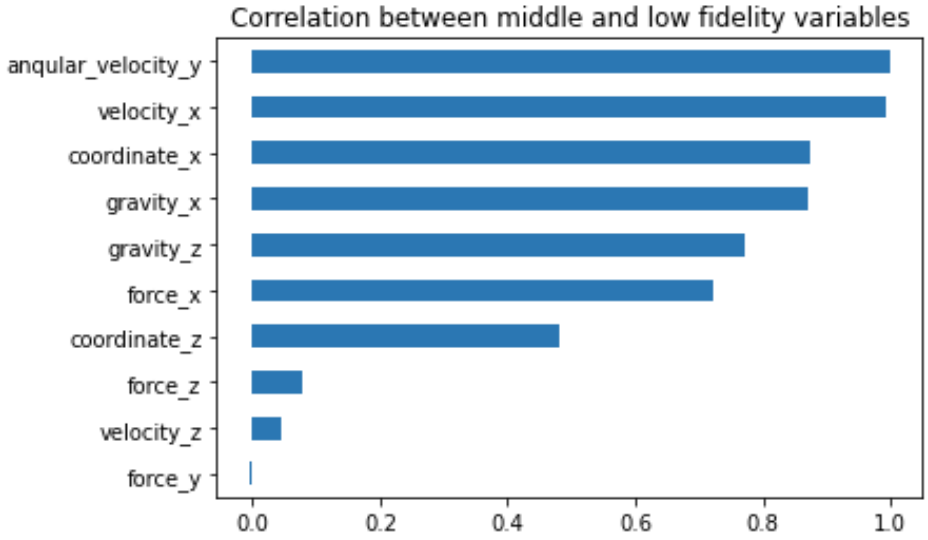


Figure 4.6: Correlation plot between the levels of fidelity

From this figure 4.6 we can deduce that there in fact is a correlation between some of the variables between datasets of different levels of fidelities.

4.1.5 Correlation of force in each simulation

In order to test the correlation of the predicted variable in both fidelity levels, we selected 250 random runs and visualised the correlation in a density plot between the force x variable in SCM and Terra datasets.

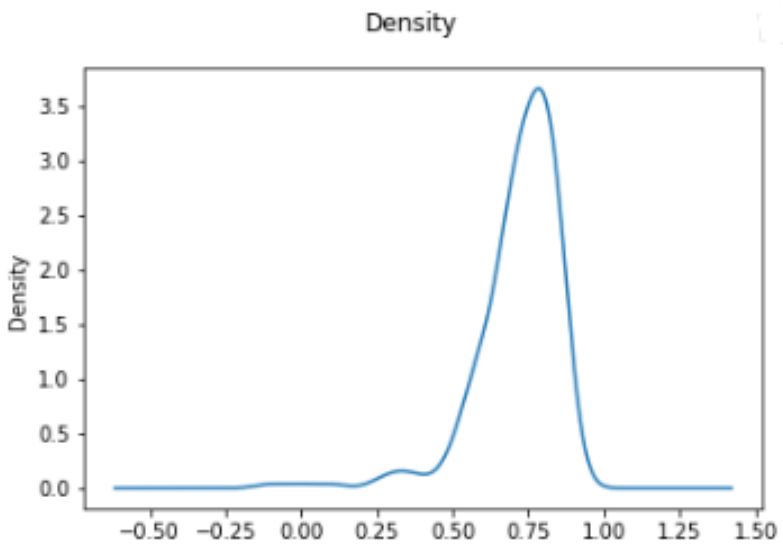


Figure 4.7: Density of correlations between $force_x$ of both fidelity levels

The figure depicts correlation for between a single run of 151 rows. From this we can see that most of the runs are highly correlated. There are however long tails, mostly on the negative side hence removing the outliers can provide more meaningful results. With the percentile outlier removal method, we filtered out the results to only retain percentiles between the 25th and the 75th.

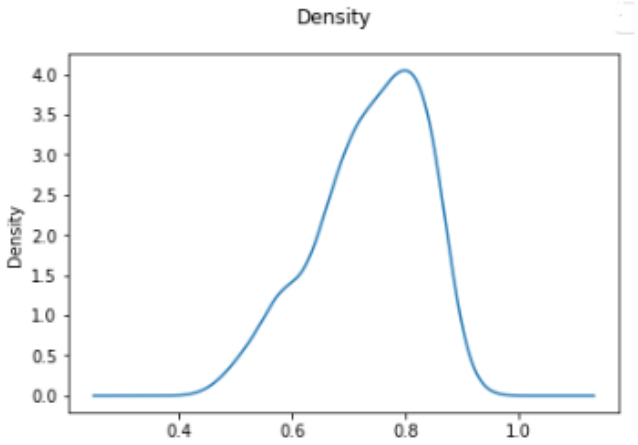


Figure 4.8: Density of correlations between $force_x$ of both fidelity levels without outliers

Not we can clearly see that with the removal of less than 4 percent of runs, the distribution is a slightly skewed normal distribution with a mean around 0.75. This means that the experiments with multi-fidelity models will be meaningful in most cases and should provide decent results.

The problem with models that should operate in real-time on incoming data is that filtering outliers might prove difficult, hence it is important to look at them and see why the correlation can differ so much from the mean in such cases. With exploring and comparing the distribution of forces on both datasets for the highest and lowest correlation outliers might provide answers.

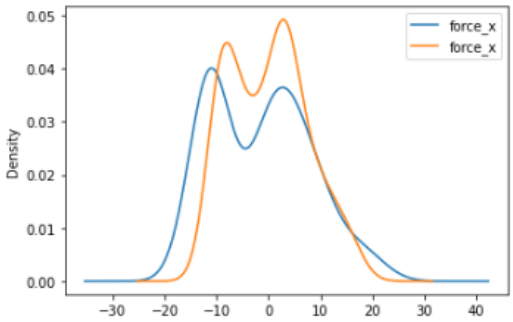


Figure 4.9: Density of $force_x$ of both fidelity levels for the highest correlation outlier (blue = scm, orange = terra)

For the highest correlation outlier we can see as intended that the force distributions have similar shape.

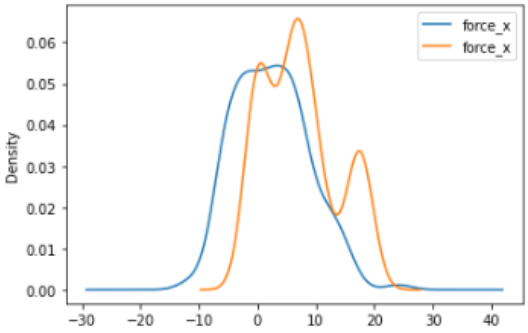


Figure 4.10: Density of $force_x$ of both fidelity levels for the lowest correlation outlier

For the lowest correlation sample, we can see that both the shapes of the distribution are incorrect and the values appear much lower for the medium fidelity case. The key to understanding the differences between those two outliers might be in the shape of the force x from the SCM model. In most highly correlated cases force x in medium fidelity data has a double apex distribution, while in the lowest correlated ones it has a single apex.

4.1.6 PCA

After consideration we decided to try and reduce the amount of data that we give to the model without reducing the amount of information. To do that we looked into several dimensionality reduction techniques and choose to use principal component analysis (PCA) 4.1.6.

Principal components of a specific data are a sequentially defined unit vectors that are calculated step by step with the usage of the previous one. The i-th vector is the one parallel to the line that best fits the points and also is orthogonal to all the previous vectors. In this case the task of best fitting line is defined as the one that minimizes the mean squared distance form the points to the line. By minimizing this value it also maximizes the variance. Thanks to the fact that all the components are orthogonal we get that the new data is linearly uncorrelated.

In previous step we have selected seven variables to be the input of our models. Those variables are: coordinate_x, coordinate_z, velocity_x, velocity_z, angular_velocity_y, gravity_x, gravity_z.

To maximize the potential reduction of dimensionality we decided to run the algorithm on both the low fidelity level data and medium fidelity level data. This will allow to potentially decrease the size of input data in both parts of the model, also it will help in searching for the best suited neural network, as any reduction in the number of variables will mean a great reduction in the calculations needed to train a network.

4.1.7 PCA on low fidelity data

The first input is the low fidelity level data. After running the algorithm on the training set without the target variables we get the following results, presented in figure 4.11.

From the closer examination we can get that the percentage of explained variance in the data is the following:

- 89% - if we take first four principal components,
- 98,8% - if we take first five principal components,
- 99,6% - if we take first six principal components.

4.1.8 PCA on medium fidelity data

The next input is the medium fidelity level data. After running the algorithm on the training set without the target variables we get the following results, presented in figure 4.12.

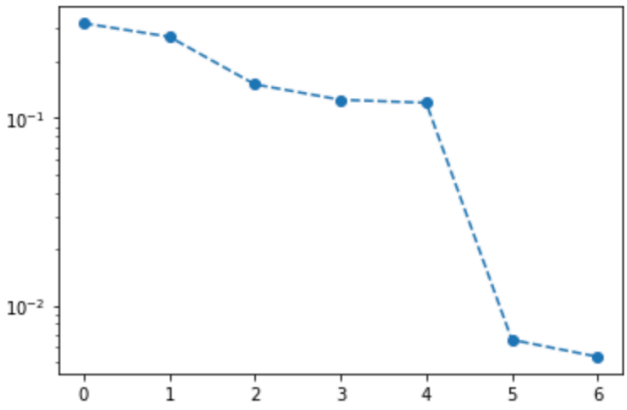


Figure 4.11: Semilogy plot of variance explained by principal components in low fidelity data

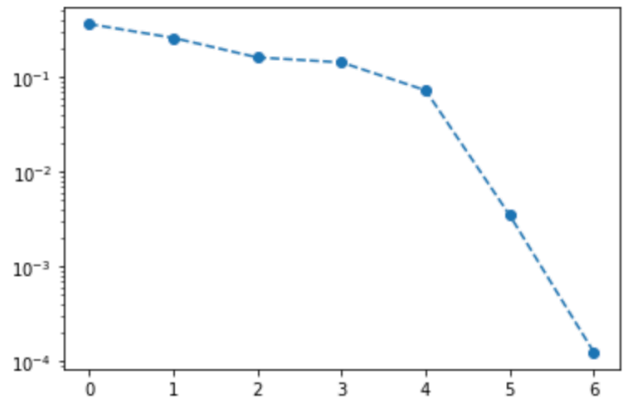


Figure 4.12: Semilogy plot of variance explained by principal components in medium fidelity data

From the closer examination we can get that the percentage of explained variance in the data is the following:

- 94% - if we take first four principal components,
- 99,8% - if we take first five principal components,
- 99,99% - if we take first six principal components.

4.2 Image data analysis

After examination of numeric features present in the dataset, we analysed the images. For each timestamp in a run, there is a photo of the soil below the wheel captured by the rover. It provides information about the terrain and the distance to the surface. The analysis was conducted to assess a possible information gain for the prediction task from including photos in the model architecture.

The images are saved as 64 by 64 pixel matrices and hence it was possible to find the most distinct photos from the mean in terms of distance. The mean is set on pixel-level and the two outliers presented in Figure 4.13 are analysed further to asses their properties.

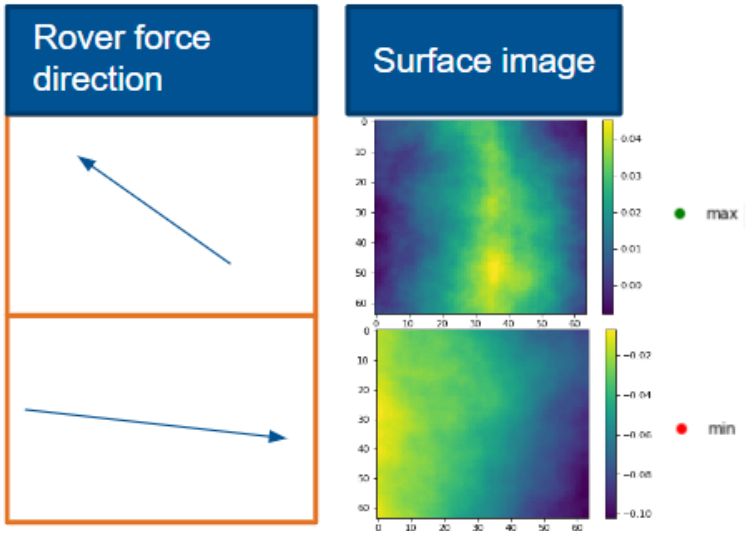


Figure 4.13: Two images present in a single run depicting the most distant datapoints from the mean image, labelled min and max accordingly. Along them the connected rover force direction at that timestamp is presented.

Based on the rover movement fitting the image in Figure 4.13, it can be observed that they proceed in opposing directions basing strictly on pixel values. This led to further investigation based on that values. For a single run the force x values were plotted to see where the min and max images stack up when compared to all datapoints. This can be observed in Figure 4.14 as they are on the opposing sides of the mean line. While that covers only a single run and two outliers, it provides a support for a claim that crucial information is present in the image data.

For further correlation testing, data over multiple runs has been collected. The mean

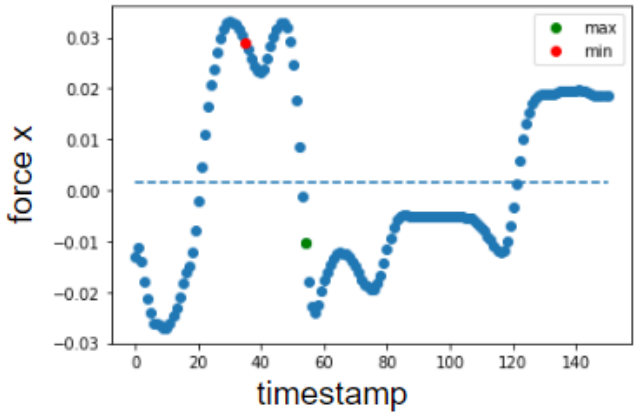


Figure 4.14: A graph presenting the force values in x direction at all timestamps in a single run. Images labelled as min and max in Figure 4.13 are presented on the graph along with a dotted line presenting the mean value of force.

and minimum values have been calculated for the matrices of pixel values along with the force x. The values have then been normalised to allow for better comparisons on a similar value scale. The results seen in Figure 4.15 show a inverse correlation between the force and pixel metrics, which is stronger in the first 8 seconds of the runs. Based on that result and previous plots, we found images to have potential in adding information to the model and decided to experiment on them, which is explained in the latter part of the report.

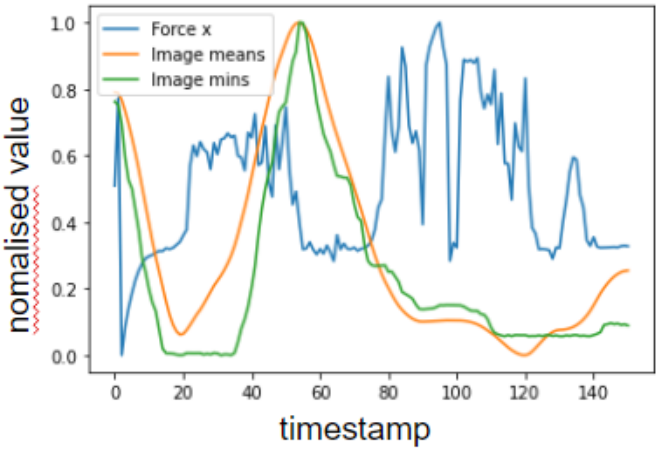


Figure 4.15: A graph presenting the force x, pixel mean, and pixel minimum values aggregated over multiple runs for a timestamp.

4.3 Convolutional Neural Networks

The next step was to try and find a right architecture of neural network for solving the problem of our prediction. In this section of our report, unless stated otherwise we are working on the variables that we have chosen before. The work with neural networks consists of several parts, that at the end come together to create one final neural network model.

4.3.1 Low fidelity neural network

The first thing that was done, was creating a low fidelity model with neural networks. It predicts the force based on the input from the low fidelity data. First we tried a standard architecture with parameters presented in table 4.3:

Parameter	Values
Dropout	0.2
Learning rate	0.005 with Adam optimiaztion
Number of epochs	80
Batch size	32
Initialization	from Normal distribution
Hidden layers	256 nodes, 512 nodes, 256 nodes

Table 4.3: Hyper-parameters of the first network

This setup resulted in the Mean Squared Error (MSE) of 201.4 when tested on the medium fidelity dataset and the training and validation errors presented in figure 4.16.

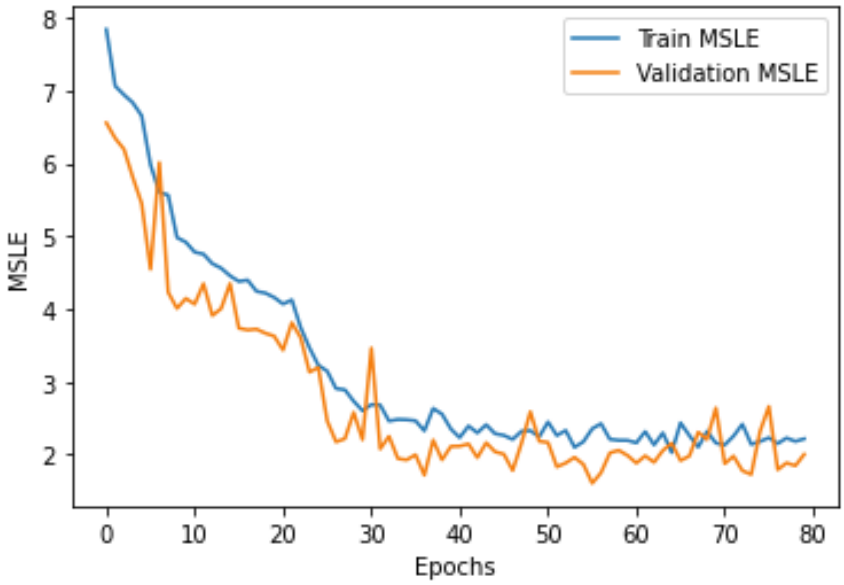


Figure 4.16: Training and validation errors of a basic neural network

We found this result to be promising and decided to do a grid search for the best parameters.

4.3.2 Grid search for parameters

The idea of grid search is creating a grid of parameters and then training and testing a neural network for every possible set created from those parameters. It can be seen that this gets problematic as the number of test runs grows exponentially with the number of parameters. Hence we need to pick important parameters for our search.

As a starting architecture we took the same as in the previous section. The parameters of the grid are presented in table 4.4:

All the networks were thought on 80 epochs and batch size equal to 32. The best sets of parameters are presented in table 4.5.

4.3.3 Neural Networks with PCA

As the next step we decided to try the neural network approach on the data from PCA transformation. We assumed that 80% of explained variance will be a good value

Parameter	Values
Dropout	0.2 or 0.5
Learning rate	0.001, 0.002, 0.005, 0.01, 0.02 or 0.1
Initialization	normal distribution N(0, 0.05) or Xavier

Table 4.4: Grid for hyper-parameters search

Parameter	Values
Dropout	0.2
Learning rate	0.002
Initialization	normal distribution N(0, 0.05)

Table 4.5: Best parameters found in grid search

to cut of the principal components. The problem of different PCA vector spaces in low and medium fidelity levels however led to a diminishing return after applying the transformation. We decided to not pursue the idea further and rather focus on enhancing the model itself as our selected feature-space is already low dimensional.

4.3.4 Image Regressor

For Neural Networks we decided to also include image data along with numerical features to really test whether the argument made in 4.2 was true. For the first tests, a very simple Convolutional Neural Network was created with pooling. This proved to not capture the data too well and hence we opted to utilise the autokeras package with the function of ImageRegressor. This method searches for the best model and parameters. This way we found a final architecture of a modified ResNet-50 [He+15]. It is a residual framework based method that is proven to work on a number of use cases.

This model allowed us to make predictions on the force output based on images. For training of the model we utilised the images for 1500 low fidelity datapoints. This model achieved a MSE of 40.2 when tested on the medium fidelity test set. This shows that when compared to a low fidelity model based strictly on features, the image model outperforms it. It might be due to less of a difference in images between fidelity levels when compared to numeric features.

4.3.5 Medium and low fidelity neural networks

With all the low fidelity models created and tuned, a multi-fidelity neural network was created. As depicted in the figure 4.17, the architecture is based on the Feed-Forward Neural Network described in 4.3.1 as well as the image regressor described in 4.3.4. The flow of the model starts with the image regressor being trained on the TerRA low fidelity image data. The prediction based on this model is than added to the feature set of the low fidelity set. On this updated features, a low fidelity feed-forward network is than trained. This concludes the low fidelity part of the model. Now adding onto that the medium fidelity level, the SCM train data is evaluated on the SCM model, which produces an output to be added to the SCM train feature space. This allows for the final low fidelity model to be evaluated on the medium fidelity data. The output of this model is stacked with the input features of SCM train, which creates a new set for the medium fidelity network to be trained on. This network is of the same architecture as the one described in 4.3.1 but has tuned parameters to fit the lower number of datapoints. This mostly came down to decreasing the learning rate to 0.001. This final model is than tested on a SCM test set to benchmark the results.

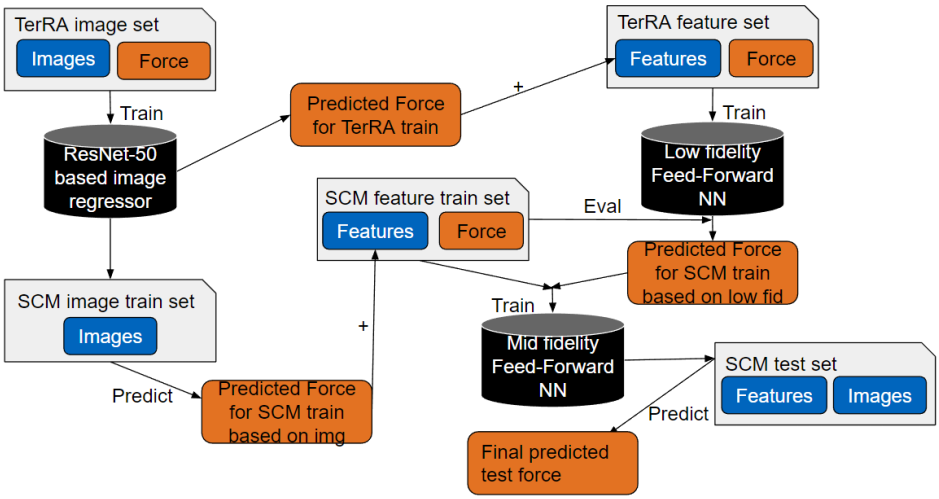


Figure 4.17: The graph presents the architecture of a final multi-fidelity neural network model used in experiments.

The loss of the final models' training process is depicted in the figure 4.18 and shows that the increase of epochs from 80 for the low fidelity model to 150 for the final medium fidelity model was a correct decision as the loss keeps declining after

the 80 epoch mark.

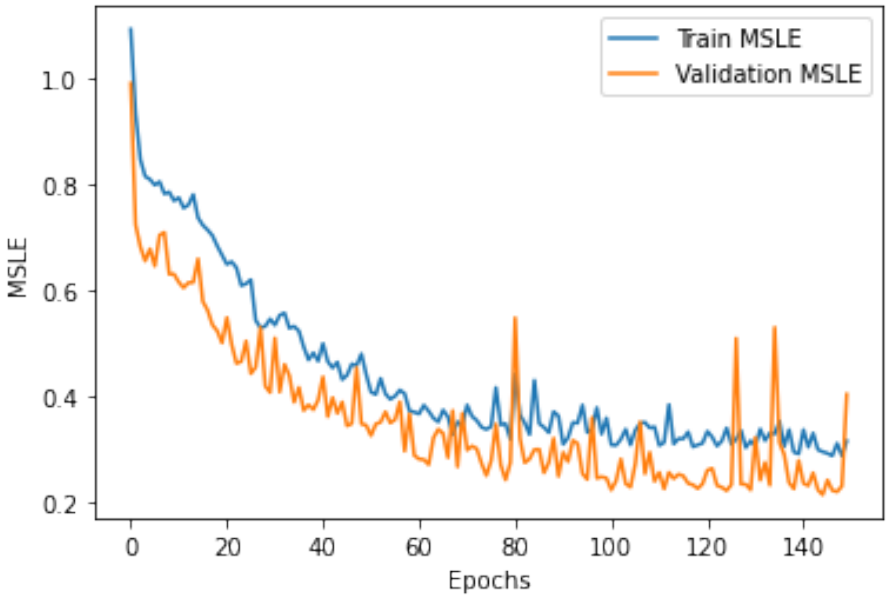


Figure 4.18: The graph presents the train and validation loss change during the training process.

This final model was trained on 1500 low fidelity, 150 medium fidelity, and 50 test points from the SCM dataset, but different from the data used in training. The results in the table 4.6 show that the final multi-fidelity model using images is much better than without images as it shows better performance on a lower number of data. Generally in our experiments the neural networks benefited from a higher number of datapoints yet the final benchmark was made on 1500 low fidelity points to ensure real-world usability.

	Features	Features + Images
Data size	10k low, 1.5k medium fidelity	1.5k low, 150 medium fidelity
MAE	6.86	2.57
MSE	63.61	11.5

Table 4.6: The table shows the comparison of the multi-fidelity Neural Network results using numerical features only (left column) and a mixture of both numeric and photo data (right column).

A medium-fidelity only model was also created to test the impact of using multi-

fidelity data. It is the same in architecture as the final multi-fidelity model without the low fidelity part. It resulted in a MAE of 3.49, which is a much worse result than the final 2.57 achieved using multi-fidelity.

4.4 Gaussian process regression

The next approach we took while modelling data was Gaussian process regression. In our case the relation between the input and output can not be mapped with a simple linear function, hence a simple regression model will not be sufficient. This is the reason why during our modelling phase we are also searching for the optimal kernel.

Kernel is a positive-definite function that maps one space to another. It allows to change spaces, dimensions and relations between data, with does properties we can model non-linear relationships with linear models.

In our case we are searching for a suitable kernel that will give the best results. We are also using combinations of several kernels at once.

4.4.1 Single-fidelity Gaussian process regression

The whole idea of the project was to see how multi-fidelity affects the results of the model. To see that there is a need for a base single-fidelity model. We decided to create a medium-fidelity Gaussian process regression to achieve this goal.

In this case we used and RBF kernel. It is given by: $K(x, y) = exp(-\gamma \|x - y\|^2)$. This results in the following errors presented in table 4.7:

MAE	MSE
5.87	54.33

Table 4.7: Single-fidelity Gaussian process results

We can see those results do not even come close to those that were achieved by fully trained neural networks. Next we can see what multi-fidelity gives us in terms of results.

4.4.2 Multi-fidelity Gaussian process regression

The next step was to use multi-fidelity and try better our results. The whole process is based on two predictions that go with the following flow presented in the figure 4.19.

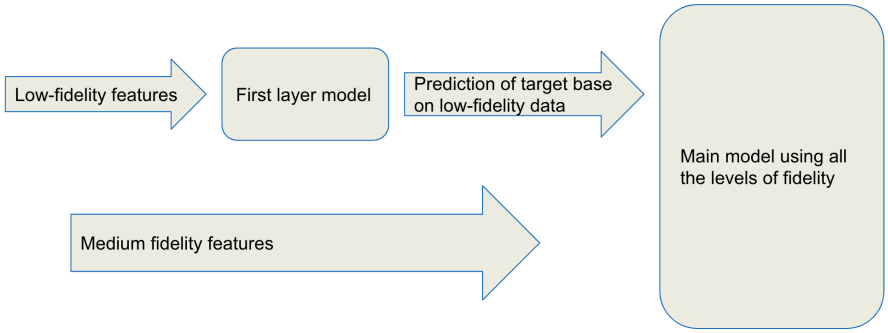


Figure 4.19: Workflow of the system

Firstly we wanted to see how does the results look in a situation comparable to the one from only medium-fidelity model. This means that we will use in both prediction the RBF kernel.

This results in the following errors presented in table 4.8:

MAE	MSE
5.70	53.17

Table 4.8: Multi-fidelity Gaussian process results

We can see there is a slight improvement when compared to 4.7.

Next we will analyse how does the kernel selection impacts the results. From our testing phase it become clear that the first kernel has to be RBF.

To find the best suitable kernel we modelled the relationship between force along the x axis between the data from low-fidelity model with the data from medium-fidelity model. This can not be done explicitly, so we decided to use a smoothing operation. This operation is based on normal distribution and convolutions. Once we found the best parameters we got the following graph presented in figure 4.20.

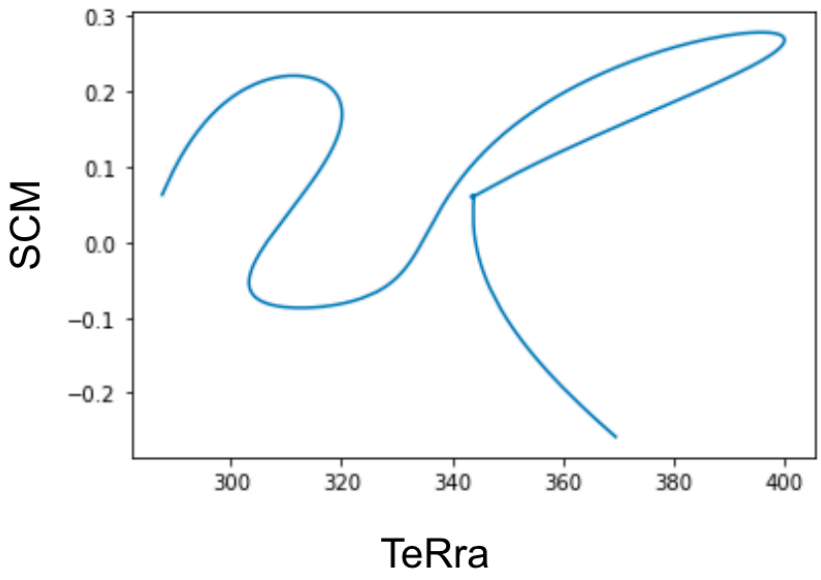
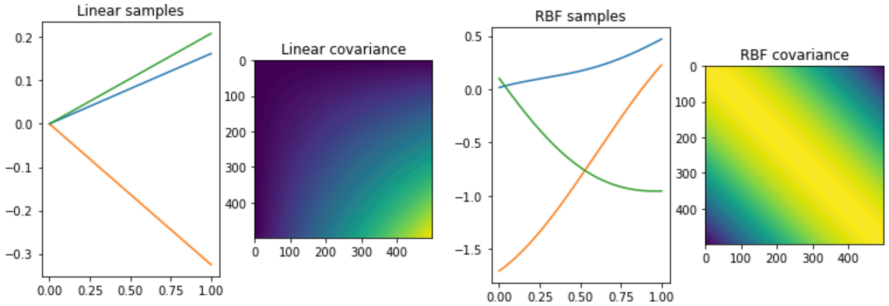


Figure 4.20: Relationship between targets in different datasets after smoothing

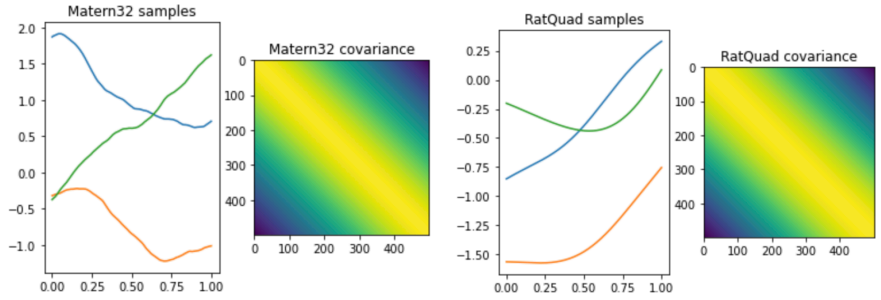
After that we listed all suitable kernels from the GPY package. We choose the two that we found closest and also two common ones to broaden the search area. Those are listed below with equations, function examples and covariance visualisation.

The kernels that we were looking at in this situation were:

1. Linear kernel, with equation: $K(x, y) = x^T y + c$
2. Radial basis function kernel, with equation: $K(x, y) = exp(-\gamma \|x - y\|^2)$
3. Matern 3/2 kernel, with equation: $K(r) = \sigma^2(1 + \sqrt{3}rexp(-\sqrt{3}))$, where $r = \sqrt{\sum_{n=1}^{inputdim} \frac{(x_i - y_i)^2}{l_i^2}}$
4. Rational Quadratic kernel, with equation: $K(r) = \sigma^2(1 + \frac{r^2}{2})^{-\alpha}$



(a) Linear kernel function and covariance (b) RBF kernel function and covariance



(c) Matern32 kernel function and covariance (d) RatQuad kernel function and covariance

Figure 4.21: The result of density analysis

We tried looking for the best possible combination of kernels using the four listed above and operations of summation and multiplication.

We tested two varieties of data. Used 500 low-fidelity points and 1500 low-fidelity points, to see if the amount of low-fidelity data has a big impact on the performance of the model.

The best results were achieved for the following combination: Linear+RBF+RatQuad +Matern32+Linear*RBF*RatQuad*Matern32. The results are presented in the table 4.9.

	MSE	MAE
500 points	51.35	5.27
1500 points	28.26	3.84

Table 4.9: The best Gaussian process

We can see there is a big improvement, while using a bigger number of low-fidelity data points. This proves that even low quality data can be very helpful when it comes to multi-fidelity.

An observation that the when compared to the medium-fidelity a significant improvement of around 35% can be made. Demonstrating the fact that even if data is of low quality it can be really helpful if used correctly.

In table 4.10 there are presented the results of different kernels and how they impact the performance of Gaussian process regression model. From the values in the table 4.10 we can deduce that adding another level of fidelity into the model can in fact improve the performance, even if as in this case only a little. Another thing that we can deduce is the fact that finding the best kernel is crucial for the optimisation task. One more interesting observation is that for different cases we can observe a decrease in MAE, but increase in MSE and in other we can observe increase in MSE, but decrease in MAE. It is important to declare the most important goal function and follow it, as the situation stated above can be misleading. From table 4.9 we can deduce that the amount of data in the training set has a high impact on the performance, but it is important to remember that the whole purpose of multi-fidelity machine learning is to use a small amount of computationally costly data to achieve the best result. We have also not been able to test all the possible combinations of kernels in the package, so there can be a better solution to this problem that the one we have found.

Kernel	MAE	MSE
RBF only on medium fidelity	5.87	54.33
RBF on multi-fidelity	5.70	53.17
Best found kernel	3.84	28.26

Table 4.10: Overview of Gaussian process performance

5 Conclusions

In our project we have utilised a lot of techniques such as: Exploratory data analysis, principal components analysis, neural networks, Gaussian process regressions and more to try and improve the performance of our final model. In this paragraph we will describe all the positives that come from them in our opinion.

In section 4.1 we described the exploratory data analysis of our low and medium-fidelity datasets. This step is a must if one thinks seriously about the project. It will uncover a lot of important characteristic about the data. Point to important variables, show which are irrelevant for our future model. In our case it helped us eliminate the variables that were constant, which allowed us to cut down on computations in a great manner. Apart from that we also visualised densities to get familiar with the data, in cases where some familiar functions show up it can be helpful. Also it may indicate if there is a need to use transformations on data to change the distribution to some more familiar or easier for the model to predict. Another thing that is of value in this type of project is seeing how all the variables correlate with the target, it indicates which to keep and which to discard.

Approach described in section 4.1.6 is very important for the project. This was trying using principal components analysis to try and even more reduce the dimensionality of the data. This step may be useful in cases when we have a lot of dimensions, but we suspect that some of them do not carry a lot of information with them. In our case it actually helped reduce the number of dimensions, if we decided to keep 80% of the variance then in both datasets we would need only four variables instead of seven. Another interesting fact is that if we actually could discard one or two dimensions from each dataset with near zero loss of information.

From the literature overview it become clear that the right kernel can make a lot of difference. In section 4.4.2 we describe multiple approaches and results on finding the correct kernel. During our test phase we discovered a lot of facts among others that combinations of kernels matter and that the more complicated the kernel or combination it does not necessary mean it will yield better results. We suspect that if the computation power did not restrict us we would be able to find even better solution with the usage of summation and multiplication combination techniques.

We have also demonstrated that even if we keep the number of medium-fidelity points the same we can change the performance with manipulation of the number of low-fidelity points. It shows that the idea of multi-fidelity is valid and can help with computation problems and hardship of obtaining high quality data.

Furthermore, we compared the two common approaches to multi-fidelity modelling and found the differences and advantages of both methods. Neural Networks during the presented experiments proved to be the best performing on the benchmarked set. When we decrease the number of low or medium fidelity data to a third - 500 TerRA and 50 SCM datapoints, we can see the trend reversing. In that case, Gaussian Process Regression outperforms the Neural Network as it reaches a MAE of 5.27 in comparison to 9.16 achieved by the NN. On the other hand, when the number of low fidelity datapoints gets bigger than 3000, the GPR gets very expensive to compute, while the NN remains computable within a reasonable time-frame.

The impact of images was also examined, demonstrating that they provide a lot of additional information for the task of force prediction. The models based only on them managed to outperform some of the numeric feature based on their own. This leads to a conclusion that they should be included in the predictions and their weight on the final predictions should be high.

5.1 Future work

There are many steps that can be done further in this work. Firstly and in our opinion the most important step is to incorporate high-fidelity data into the model. It will mean that there will be a need for more hyper-parameters and also more kernels. It will result in a need for a new model.

Once all three levels of data are taken into account, all the steps have to be repeated. We start with the exploratory data analysis for high-fidelity data, then we do the neural network steps and Gaussian process regression.

Good improvement would be doing a bigger search of hyper-parameters and kernels, but for that there would be a need of machines with better computation capabilities than our personal computers.

As we have demonstrated the images can make a lot of difference in the quality of prediction. We suspect time spent working with the photos, maybe doing more exploratory steps or trying to use manipulations to produce more images.

List of Figures

List of Tables

Bibliography

[Abd+17] A. B. Abdessalem, N. Dervilis, D. J. Wagg, and K. Worden. "Automatic Kernel Selection for Gaussian Processes Regression with Approximate Bayesian Computation and Sequential Monte Carlo." In: *Frontiers in Built Environment* 3 (2017). ISSN: 2297-3362. DOI: 10.3389/fbuil.2017.00052.

[Bar18] S. Barthelmes. "TerRA: Terramechanics for Real-time Application." In: *5th Joint International Conference on Multibody System Dynamics*. June 2018.

[Cha22] T.-C. Chang. "Data-driven multi-fidelity modeling for terramechanics." In: (Apr. 2022), p. 72.

[Con+12] M. T. Contreras, C.-Y. Peng, D. Wang, and J.-S. Chen. "Determining Wheel-Soil Interaction Loads using a Meshfree Finite Element Approach Assisting Future Missions with Rover Wheel Design." In: 2012.

[Dra20] N. Drake. "Why we explore Marsand what decades of missions have revealed." In: (2020).

[Dra22] N. Drake. "What We Learned from the Perseverance Rovers First Year on Mars." In: (Feb. 2022), p. 17.

[Gal+14] A. Gallina, R. Krenn, M. Scharringhausen, T. Uhl, and B. Schäfer. "Parameter Identification of a Planetary Rover WheelSoil Contact Model via a Bayesian Approach." In: *Journal of Field Robotics* 31 (Jan. 2014). DOI: 10.1002/rob.21480.

[Guo+22] M. Guo, A. Manzoni, M. Amendt, P. Conti, and J. S. Hesthaven. "Multi-fidelity regression using artificial neural networks: Efficient approximation of parameter-dependent output quantities." In: *Computer Methods in Applied Mechanics and Engineering* 389 (2022), p. 114378. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.114378.

[Haw+18] L. Hawker, P. Bates, J. Neal, and J. Rougier. "Perspectives on Digital Elevation Model (DEM) Simulation for Flood Modeling in the Absence of a High-Accuracy Open Access Global DEM." In: *Frontiers in Earth Science* 6 (2018). ISSN: 2296-6463. DOI: 10.3389/feart.2018.00233.

[He+15] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385.

[Hu+21] C. Hu, J. Gao, X. Song, M. Zhang, and X. Tan. "Analytical modeling and DEM analysis of soilwheel interaction under cornering and skidding conditions in off-road vehicles." In: *AIP Advances* 11.8 (2021), p. 085122. DOI: 10.1063/5.0057046. eprint: https://doi.org/10.1063/5.0057046.

[HW21] E. Hüllermeier and W. Waegeman. "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods." In: *Machine Learning* 110.3 (Mar. 2021), pp. 457–506. ISSN: 1573-0565. DOI: 10.1007/s10994-021-05946-3.

[Lee+19a] S. Lee, F. Dietrich, G. E. Karniadakis, and I. G. Kevrekidis. "Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion." In: *Interface Focus* 9.3 (Apr. 2019), p. 20180083. DOI: 10.1098/rsfs.2018.0083.

[Lee+19b] S. Lee, F. Dietrich, G. E. Karniadakis, and I. G. Kevrekidis. "Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion." In: *Interface Focus* 9.3 (2019), p. 20180083. DOI: 10.1098/rsfs.2018.0083. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsfs.2018.0083.

[TMN18] A. Tasora, D. Mangoni, and D. Negrut. "An Overview of the Chrono Soil Contact Model (SCM) Implementation." In: (Aug. 2018).